

ORGANIZAÇÃO:



ISBN 972-98464-6-4
IMPRESSÃO OUTUBRO 2005

de computação gráfica
encontro português

13^o

12 » 14 Out. 05
Universidade de
Trás-os-Montes
e Alto Douro

Prefácio

Este volume contém as contribuições dos Autores para o 13º Encontro Português de Computação Gráfica, realizado em Vila Real entre 12 e 14 de Outubro de 2005. Tal como em edições anteriores, o Encontro visa essencialmente a reunião de investigadores, docentes e profissionais da Computação Gráfica e áreas afins, permitindo a divulgação de trabalhos realizados ou em curso, bem como a troca de experiências e a discussão de temas relacionadas com a Computação Gráfica.

Todos os artigos foram avaliados e revistos por Membros da Comissão Científica e por revisores convidados. Neste processo foram seleccionados 14 artigos, 15 comunicações curtas e 3 *posters*, nas várias áreas de estudo: modelação geométrica, ambientes virtuais e realidade aumentada, animação, visão por computador, rendering, interacção com o utilizador, visualização científica e de informação, processamento de imagem, multimédia, computação móvel e outras.

O volume está estruturado em três secções: os artigos, as comunicações curtas, e os *posters*. As comunicações curtas permitem a apresentação escrita e oral de trabalhos de menor dimensão ou em fase de desenvolvimento. Os *posters*, de dimensão inferior, permitem um registo escrito de ideias embrionárias.

O programa inclui ainda três apresentações convidadas de inegável valor: Bryan Wyvill da Universidade de Calgary, Canadá, versando sistemas de modelação implícita, Carsten Rother da Microsoft Research, Cambridge, Reino Unido, em visão por computador e, finalmente, Wayne Wooten da Pixar Animation Studios, EUA, em animação por computador e produção de filmes.

Para terminar, gostaria de agradecer aos Autores pelas suas contribuições, aos Membros da Comissão Científica pelo trabalho desenvolvido, à Comissão Organizadora pela realização da conferência em Vila Real, e ainda, ao Grupo Português de Computação Gráfica pelo apoio institucional. Agradeço ainda ao meu colega Miguel Goulão pelo auxílio prestado na gestão do sistema de submissões e revisões dos artigos.

Adriano Lopes

Comissão Científica

Abel Gomes	UBI
Adérito Marcos	CCG, UM
Adriano Lopes	FCT/UNL
Ana Maria Mendonça	FE/UP
Ana Paula Cláudio	FC/UL
António Augusto Sousa	FE/UP
António Costa	ISEP/IPP
Armando J. Pinho	IEETA, UA
Beatriz Carmo	FC/UL
Beatriz Sousa Santos	DET/UA
Fernando Nunes Ferreira	FE/UP
Hélder Araújo	ISR, UC
João Cunha	LNEC
João Luís Monteiro	DEI/EE/UM
João Paulo Costeira	ISR, IST/UTL
João Pereira	IST/UTL
Joaquim Jorge	IST/UTL
Joaquim Madeira	DET/UA
Jorge Salvador Marques	ISR, IST/UTL
José Carlos Teixeira	FCT/UC
José Dionísio	IST/UTL
Luís Magalhães	UTAD
Luís Marcelino	FCT/UNL
Luís Paulo Santos	UM
Manuel João Fonseca	IST/UTL
Mário Rui Gomes	IST/UTL
José Miguel Dias	ISCTE
Miguel Leitão	ISEP/IPP
Miguel Velhote Correia	FE/UP
Nuno Correia	FCT/UNL
Nuno Jardim Gomes	UMA
Óscar Mealha	DeCA/UA
Pedro Faria Lopes	ISCTE
Próspero dos Santos	FCT/UNL
Rafael Bidarra	UT Delft, Holanda
Rogério Caldas Pinto	IST/UTL
Teresa Chambel	FC/UL
Teresa Romão	UE

Índice

Artigos

PolyMeCo - Uma Ferramenta de Análise e Comparação de Malhas Poligonais Samuel Silva, Joaquim Madeira e Beatriz Sousa Santos	1
Superfícies de Coons: Estudo comparativo de alguns algoritmos de geração e análise da sua qualidade Eunice Oliveira e José Carlos Teixeira	9
Algoritmo de Subdivisão de Superfícies Deformáveis para Execução em GPU Fernando Birra e Manuel Próspero Santos	19
Round Tables: algoritmo para rasterização run-length de formas convexas Helder Magalhães	29
Calculating the stereo pairs of a mirror-based augmented reality system Filipe Marreiros e Adérito Marcos	41
Zarabatana, uma interface tangível para ambientes de entretenimento em realidade aumentada Miguel Dias e António Pocinho	49
TARCAST: Uma Taxonomia para Sistemas de Realidade Aumentada José Pereira e João Pereira	57
Avaliando a Veracidade das Histórias sobre Documentos Daniel Gonçalves e Joaquim Jorge	69
Plataforma de Suporte à Geração de Cenas Animadas com Agentes Inteligentes M. Beatriz Carmo, Ana Paula Cláudio, João D. Cunha, Helder Coelho, Miguel Silvestre e Maria Pinto-Albuquerque	79
Modelação Expedita de Ambientes Virtuais Urbanos para Utilização em Dispositivos Móveis Maximino Bessa, António Coelho, José Bulas Cruz, Fernando Ferreira e António A. de Sousa	85
Visualização de Dados Arqueológicos: Um Exemplo de Utilização do Visualization Toolkit Paulo Bernardes e Joaquim Madeira	95
Visualização de edifícios a partir de dados LIDAR Nuno Rodrigues, Adriano Lopes, Manuel Próspero Santos e Gil Gonçalves	105
Qualidade Medida vs. Qualidade Percebida em Malhas Triangulares dos Pulmões Samuel Silva, Joaquim Madeira, Carlos Ferreira e Beatriz Sousa Santos	111
Caracterização do Movimento Tridimensional de Corpos Articulados Pedro Couto, Luís Magalhães, Ronaldo Gabriel e Pedro Melo Pinto	119

Comunicações curtas

Jogo Educacional para Telemóveis

Ana Prata, Susana Lobão e Mário Rui Gomes 129

InStory Client – a Browser for Spatial Narratives and Gaming Activities

Tiago Martins, Nuno Correia, Filipe Barrenho e Teresa Romão 133

Construção de Ambientes Virtuais – Um método de trabalho

Fernando Silva e Alexandrino Gonçalves 139

Um Ambiente de Baixo Custo para Aplicações de Realidade Virtual e Aumentada

Paulo Moreira, Mário Cruz, Paulo Dias, Paulo Martins, Rui Gomes, Joaquim Madeira,
Manuel Bernardo Cunha e Beatriz Sousa Santos 145

Using Essential Reality's P5 glove as a low cost tracking device for Virtual Reality applications

Bruno Oliveira e Pedro Ângelo 151

Sistema de orientação para o interior de edifícios, uma solução visual de localização

Miguel Dias, João Marcos e António Lopes 155

A Graph-Matching Approach to Sketch-Based Modeling

Filipe Dias e Joaquim Jorge 161

Direct Manipulation of Feature Models using Handles

Daniel Lourenço, Pedro Oliveira e Rafael Bidarra 167

Malhas Poligonais Multiresolução

Frutuoso Silva 173

Improved Line/Edge Detection and Visual Reconstruction

João Rodrigues e J.M.H. du Buf 179

Web LOG File Information Analysis and Visualization Inside Organizations

Florin Zamfir, Óscar Mealha, José Nunes e Beatriz Sousa Santos 185

LEMe Wall: Desenvolvendo um Sistema de Multi-Projecção

Bruno Araújo, Tiago Guerreiro e Joaquim Jorge 191

Visual Syntax Analysis for Calligraphic Interfaces

Filipe Pereira, Joaquim Jorge e Manuel Fonseca 197

Recuperação de Desenhos de Moldes Combinando Imagens e Esboços

Alfredo Ferreira, Manuel J. Fonseca e Joaquim Jorge 203

Simplificação de Plantas de Arquitectura para Recuperação Usando Esboços

António Simão, Ricardo Ramião, Alfredo Ferreira, Manuel J. Fonseca e Joaquim Jorge 209

Posters

Publicação Digital em Computação Gráfica: Uma proposta de redesenho da Revista Virtual

Nuno Gonçalves, Rodrigo Machado, Frederico Figueiredo e Joaquim Jorge 215

SeaView: a Seafloor and Habitat Visualizer

Daniel Almeida, Samuel Nunes e Hans du Buf 219

Simulação Visual de Trajectórias Aéreas

Cesário Nogueira, Raul Costa, Carlos Amaro, António Coelho, Maximino Bessa, Manuel Cabral,
António Valente, Raul Morais e Salviano Soares 223

ARTIGOS

PolyMeCo – Uma Ferramenta de Análise e Comparação de Malhas Poligonais

Samuel Silva
IEETA – Univ. de Aveiro
Aveiro
sss@ieeta.pt

Joaquim Madeira
DET / IEETA – Univ. de Aveiro
Aveiro
jmadeira@det.ua.pt

Beatriz Sousa Santos
DET / IEETA – Univ. de Aveiro
Aveiro
bss@det.ua.pt

Sumário

A representação de modelos usando malhas poligonais é utilizada em diversos domínios. Por vezes, de forma a adequar uma determinada malha a uma aplicação particular, torna-se necessário processá-la (p.ex., para atenuar ruído), o que conduz a uma alteração das suas características (p.ex., número e/ou posição dos vértices). Qualquer operação de processamento introduz, assim, diferenças em relação à malha original que, de acordo com a aplicação a que o modelo se destina, deverão ser avaliadas.

Neste artigo é apresentada uma ferramenta, para análise e comparação das características de malhas poligonais, que procura ser inovadora na forma como permite visualizar e comparar os valores obtidos para diferentes números de mérito. Assim, é possível trabalhar em simultâneo com vários modelos e diversos números de mérito, e usar diferentes tipos de representações (p.ex., modelos coloridos ou histogramas) para visualizar propriedades intrínsecas de uma malha, ou a distribuição de uma medida de desvio de uma malha processada relativamente à original.

Palavras-chave

Malhas poligonais, Análise e Comparação, Visualização

1. INTRODUÇÃO

A representação de modelos usando malhas poligonais é utilizada em diferentes domínios de aplicação, desde a Visualização Médica até aos jogos por computador, passando pela Arqueologia e sistemas de Desenho Assistido por Computador. Nas diversas áreas, as malhas utilizadas devem, tanto quanto possível, adequar-se à aplicação a que se destinam. Nesse sentido, muitas vezes é necessário processá-las de modo a dar-lhes características mais apropriadas, por exemplo, para possibilitar um determinado grau de interactividade ou animar uma dada cena com uma qualidade aceitável. No entanto, ao modificar as características das malhas originais, diminuindo o seu número de faces (simplificação) ou modificando a posição dos seus vértices (por exemplo, filtragem de ruído), é introduzida uma diferença (erro ou desvio). Esta diferença deverá sempre ser avaliada no sentido de se perceber o seu impacto no âmbito da aplicação a que os modelos se destinam.

Existem diversas ferramentas que permitem a comparação das características de malhas poligonais relativamente a uma malha de referência [Cignoni98, Zhou01, Aspert02, Roy04], fornecendo quer dados numéricos (erro mínimo, médio, máximo e variância; distância de Hausdorff), quer informação acerca da distribuição do erro por representação de um modelo

colorido de acordo com o erro calculado em cada vértice. Em [Zhou01] foi também prestada atenção à forma como a informação relativa ao erro pode ser apresentada e são mostradas algumas alternativas de visualização.

A ferramenta apresentada neste artigo, PolyMeCo (*“Polygonal Mesh Comparison Tool”*), tem as seguintes características inovadoras:

- Ambiente integrado em que é possível trabalhar, em simultâneo, com diversos modelos e vários números de mérito (propriedades intrínsecas ou medidas de diferença).
- Comparação dos resultados do cálculo de diferentes números de mérito usando a) modelos 3D coloridos, sendo possível a utilização de uma escala de cor comum, b) representações estatísticas, como histogramas e diagramas de extremos e quartis (*“boxplots”*).
- Ajustamento da escala de cor de modo a esta ser usada para representar uma gama específica de valores, rejeitando, por exemplo, *“outliers”* que estejam presentes nos dados.
- Sonda que permite obter o valor do erro (ou outra propriedade) associado aos vértices e faces de uma malha (figura 3c).

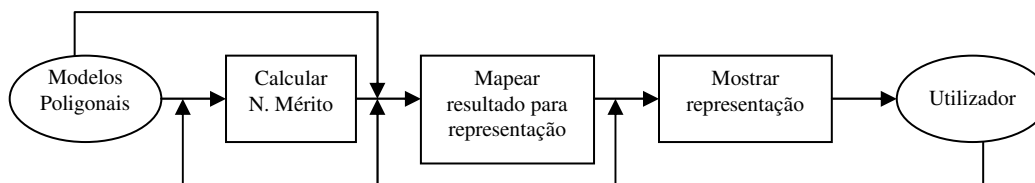


Figura 1 – Diagrama da sequência de operações envolvidas no cálculo e visualização de números de mérito associados a malhas poligonais

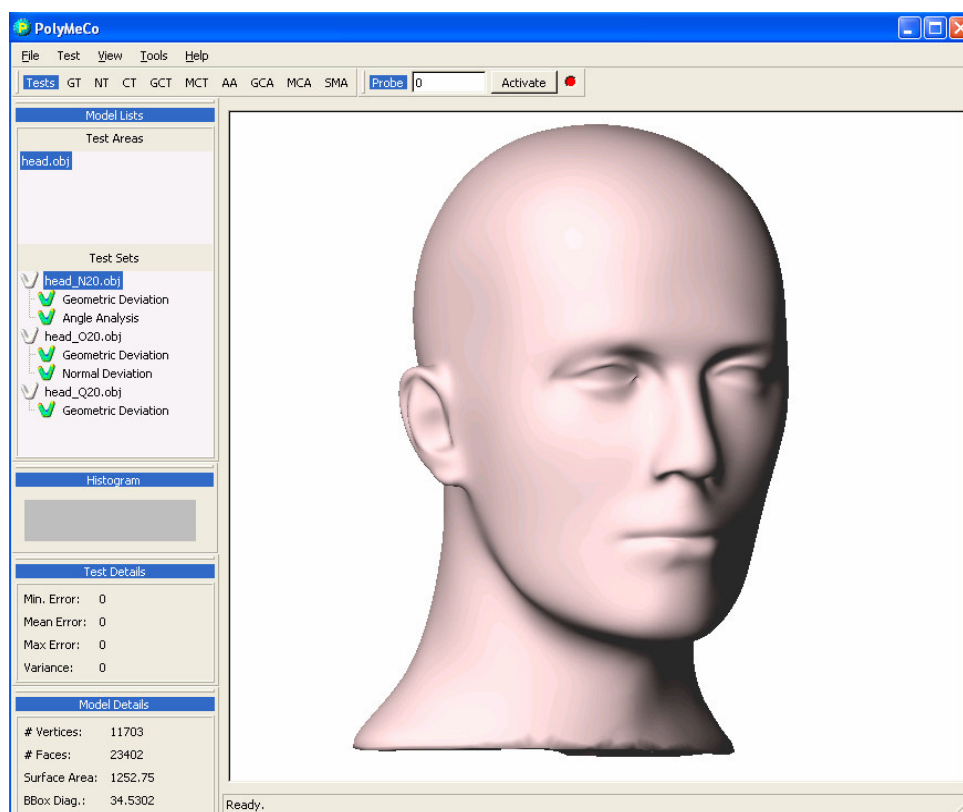


Figura 2 – A interface do PolyMeCo exibindo um modelo poligonal

Nas secções seguintes é feita uma apresentação das características e funcionalidades principais do PolyMeCo. Começa-se por dar uma ideia geral da arquitectura seguida, sendo depois feita uma descrição do tipo de números de mérito que esta ferramenta permite calcular, assim como das representações e modos de visualização disponíveis, no sentido de facilitar a compreensão e análise dos resultados obtidos. Finalmente, são apresentadas conclusões, sobre o trabalho já desenvolvido, e alguns objectivos para trabalho futuro.

2. POLYMECO

O PolyMeCo é uma ferramenta desenvolvida usando C++ e a biblioteca Fox Toolkit [Fox05], para plataformas Windows e Linux, que permite a análise e comparação das características de modelos representados usando malhas poligonais; esta representação é feita recorrendo às bibliotecas OpenMesh [Botsch02] e OpenGL. O processo de análise e comparação pode ser descrito pelo diagrama apresentado na figura 1. Assim, partindo de um

modelo poligonal, algumas das suas propriedades são descritas por números de mérito. Os valores obtidos pelo cálculo destes números de mérito são mapeados numa representação adequada. Depois deste mapeamento, essa representação é mostrada ao utilizador. Este pode alterar parâmetros ao longo de todo o processo de modo a, por exemplo, escolher o número de mérito desejado ou a representação a utilizar.

Com base nestas ideias, o PolyMeCo tem três módulos principais:

- **Números de mérito** – permite o cálculo de diversos números de mérito, i.e., diversas maneiras de medir e/ou comparar propriedades de malhas.
- **Representações** – proporciona formas de representar os resultados obtidos com os diferentes números de mérito.
- **Apresentação/Interacção** – permite visualizar e interagir com as diversas representações disponíveis.

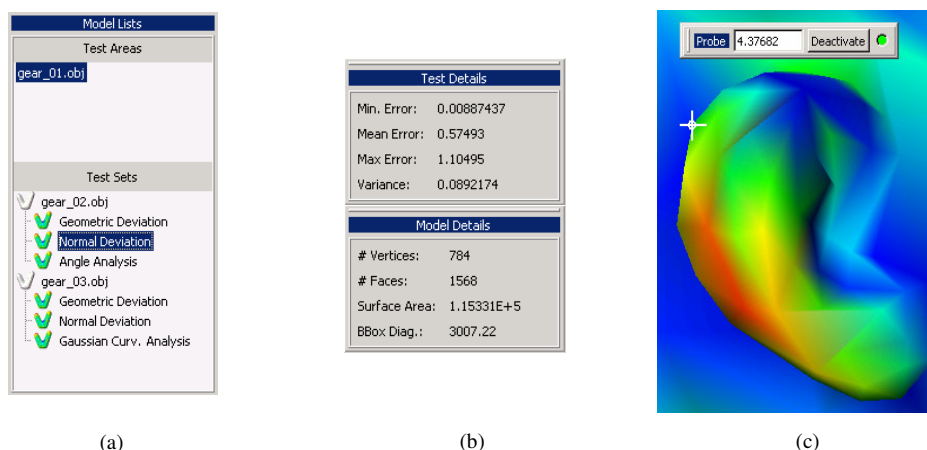


Figura 3 – Detalhes da interface.

Com o PolyMeCo é possível trabalhar com diversos modelos e resultados simultaneamente. Tome-se o seguinte exemplo: pretende-se comparar diversas versões simplificadas de um modelo com o seu original. Primeiro, efectua-se o carregamento do modelo original. De seguida, efectua-se o carregamento de todos os modelos simplificados, associando-os com o modelo original, i.e., quando for efectuada uma operação de comparação, estes serão comparados com o original. Finalmente, podem efectuar-se as comparações usando um número de mérito apropriado e os resultados aparecem associados a cada um dos modelos simplificados. Os modelos presentes, assim como todas as comparações efectuadas, ficam organizados numa estrutura em árvore que o utilizador pode utilizar para escolher o que pretende visualizar (figura 3a). Quando se pretender calcular o mesmo número de mérito para todos os modelos associados a um modelo original, o PolyMeCo permite efectuar essa operação rapidamente com apenas um comando. A figura 2 mostra a interface principal do PolyMeCo.

3. NÚMEROS DE MÉRITO

O PolyMeCo proporciona dois tipos de números de mérito: medidas de propriedades intrínsecas de uma malha poligonal, e medidas de diferenças (erro ou desvio) entre duas malhas. Estas últimas são obtidas habitualmente pela comparação de propriedades intrínsecas.

3.1 Propriedades Intrínsecas

Na versão actual do PolyMeCo, encontram-se disponíveis as seguintes propriedades intrínsecas:

Qualidade dos Triângulos – A qualidade dos triângulos é analisada medindo o menor ângulo (entre 0 e 60 graus) de cada triângulo que compõe a malha. Esta medida permite obter uma ideia geral sobre a forma de cada triângulo: se o menor ângulo for próximo de 60 graus então o triângulo está próximo de ser equilátero. Quando o menor ângulo é pequeno não é possível, usando esta métrica, saber se o triângulo a que este pertence é “achatado” (“flat”) ou “alongado” (“needle”). Nestas

situações uma análise mais elaborada pode ser efectuada usando os números de mérito descritos e analisados em [Pébay03].

Curvatura – É permitido o cálculo das curvaturas média e Gaussiana. A curvatura pode ser uma boa métrica para análise de malhas, já que se relaciona directamente com a forma da superfície que descreve. Alguns métodos têm sido propostos que permitem o cálculo da curvatura para malhas poligonais [Zhou01, Meyer02, Rusinkiewicz04]. A curvatura é aproximada calculando os seus valores para cada vértice. Os métodos descritos em [Zhou01] e [Meyer02] foram implementados, tendo o segundo sido escolhido, principalmente por proporcionar valores de curvatura mais próximos dos valores teóricos.

3.2 Medidas de Diferença

As medidas de diferença são obtidas comparando propriedades de duas malhas. Uma das malhas é considerada a referência, sendo a outra uma sua versão processada.

Na versão actual do PolyMeCo, encontram-se disponíveis as seguintes medidas de diferença:

Desvio geométrico – Permite a comparação de duas malhas através da medida das diferenças geométricas locais entre as suas superfícies. Quando se comparam malhas com diferente número de vértices (por exemplo, devido a uma simplificação) é necessário extrair pontos da superfície da malha processada de modo a permitir pontos comuns de comparação. Este tipo de “amostragem” da superfície da malha processada está disponível na ferramenta MeshDev [Roy04]. Nesse sentido, a versão deste teste apresentada no PolyMeCo é fortemente baseada na do MeshDev.

Desvio de normais – Permite medir qual é a variação das normais às faces entre dois modelos. A implementação desta medida de diferença, pelas razões já referidas anteriormente, é fortemente baseada na do MeshDev.

Desvio de curvatura – Permite a comparação das curvaturas média e Gaussiana entre duas malhas.



Figura 4 – Modelos coloridos de acordo com os resultados obtidos com diversos números de mérito: da esquerda para a direita, desvio geométrico, curvatura Gaussiana e qualidade dos triângulos.

4. REPRESENTAÇÕES

Existem diversas formas de representar os dados obtidos usando os diferentes números de mérito, de modo a poderem ser analisados. O PolyMeCo permite três tipos de representações:

- Valores numéricos
- Modelo tridimensional
- Representações estatísticas

4.1 Valores numéricos

Uma forma de apresentar a informação relacionada com as malhas ou os números de mérito calculados é simplesmente apresentando alguns valores numéricos que os caracterizem.

Os seguintes valores numéricos estão disponíveis:

Características da malha: número de vértices, número de faces, área da superfície e diagonal da “*bounding box*” (figura 3b).

Características dos Números de Mérito: valores mínimo, médio, máximo e variância (figura 3b).

4.2 Modelo Tridimensional

A análise e comparação das propriedades de uma malha podem ser efectuadas recorrendo a modelos tridimensionais: quer simplesmente pela sua representação geométrica, quer pela sobreposição da distribuição dos valores obtidos com um determinado número de mérito ao longo da superfície do modelo. Todos os modelos são apresentados, por omissão, usando uma projecção paralela ortogonal.

Visualização do modelo tridimensional – Uma possível forma de analisar algumas características de um modelo é através da inspecção da sua superfície. Os modelos podem ser visualizados usando sombreamento constante ou de Gouraud. É ainda possível visualizá-los em modo “*wireframe*” para, por exemplo, inspecionar a distribuição dos vértices.

Modelo colorido – Perceber, para qualquer número de mérito, como os seus valores se distribuem ao longo da malha é bastante importante, uma vez que permite julgar quanto à sua importância local. Isto pode ser conseguido colorindo um modelo de acordo com os valores calculados para um determinado número de mérito. Na

versão actual, o PolyMeCo permite colorir os modelos usando uma escala de cor do tipo espectro da luz solar (do azul ao vermelho). Usando esta escala de cor, o valor 0 é representado usando a cor azul, sendo o valor máximo obtido para cada número de mérito representado usando vermelho. Apesar desta escala de cor não ser percebida como linear [Bergman95], é a escala usada, uma vez que se pretende fornecer, antes de mais, uma informação qualitativa com os modelos coloridos.

Se a distribuição mapeada pertencer a uma propriedade intrínseca, é sobreposta ao modelo para o qual foi calculada. Se pertencer a uma medida de diferença, é sobreposta ao modelo de referência usado. A figura 4 mostra alguns exemplos de modelos coloridos de acordo com os resultados obtidos com diversos números de mérito.

4.3 Representações estatísticas

Apesar de, através dos modelos coloridos, ser possível ter alguma ideia da distribuição dos valores de um número de mérito ao longo da malha, mostrando onde esses valores ocorrem, pode ser difícil obter uma ideia global, devido à impossibilidade de, por exemplo, visualizar toda a superfície do modelo em simultâneo. As representações estatísticas disponibilizadas podem ajudar a perceber e comparar melhor os resultados obtidos.

Histogramas – Ao representar os valores obtidos através de um número de mérito usando um histograma, permite-se ao utilizador obter informação complementar àquela dada pelo modelo colorido, uma vez que o histograma resume a forma como os valores de um número de mérito estão distribuídos.

Um problema que ocorre ao representar os valores obtidos através de um número de mérito (ou outro qualquer tipo de dados) usando uma escala de cor, é o facto da presença de “*outliers*” aumentar a gama da escala, não permitindo a distinção entre valores próximos. Tendo isto em conta, foi adicionada uma característica especial à representação do histograma que permite variar a gama de valores representada e a escala de cor utilizada. É possível, usando dois “*sliders*”, definir os limiares de saturação inferior e superior. Os valores abaixo do limiar inferior de saturação são representados a azul e os valores acima do limiar superior de saturação são representados a vermelho, o que deixa toda a escala

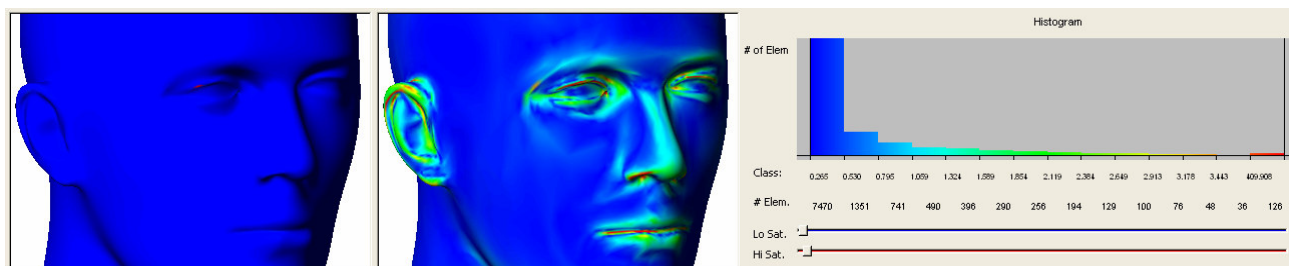


Figura 5 – À esquerda, modelo colorido de acordo com os resultados do cálculo da curvatura Gaussiana: a presença de valores de curvatura elevados não permite uma visualização da distribuição. Ao centro, modelo colorido após a redefinição da gama de valores representada pela escala de cor, o que permite visualizar melhor a incidência dos valores da curvatura. À direita, o histograma apresentando os *sliders* usados para modificar a gama de valores representados.

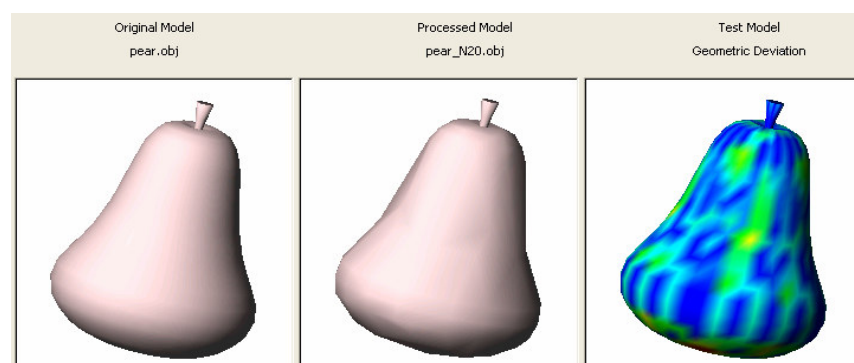


Figura 6 – Visualização em simultâneo do modelo original, do modelo processado e do modelo colorido de acordo com os resultados obtidos por uma medida de desvio geométrico

de cor (assim como todas as barras do histograma excepto a primeira e a última, que conterão os valores saturados) para representar a gama de valores entre os dois limiares.

Esta funcionalidade permite visualizar de forma mais clara a distribuição dos valores de um número de mérito quer no histograma, quer no modelo colorido, que é re-colorido de acordo com as variações da escala de cor. É ainda possível visualizar, por exemplo, onde é que os valores do número de mérito são maiores ou menores do que um determinado valor. A figura 5 apresenta (à esquerda) um modelo colorido de acordo com os valores de curvatura Gaussiana calculados. Note-se que não é possível perceber a distribuição dos valores, já que a presença de valores de curvatura muito elevados teve como consequência que todos os outros fossem representados com uma mesma cor. Através de uma mudança na gama de valores representados pela escala de cor, actuando sobre os “*sliders*”, tornou-se possível perceber um pouco melhor como a curvatura está distribuída (modelo ao centro).

Diagramas de extremos e quartis – Os diagramas de extremos e quartis são bastante úteis quando se pretende comparar conjuntos de dados. Através deste tipo de diagrama, diferenças de localização, gama e assimetrias são claramente visíveis. Um diagrama de extremos e quartis permite também averiguar da simetria do conjunto

de dados e verificar a presença de “*outliers*”. A detecção de “*outliers*” pode também dar algumas pistas sobre a forma como os “*sliders*” associados ao histograma podem ser ajustados, de modo a melhor visualizar os dados.

5. MODOS DE VISUALIZAÇÃO

No PolyMeCo tenta-se proporcionar ao utilizador diversas formas de alternar entre representações, ou de visualizar várias delas em simultâneo, de forma a permitir uma mais fácil análise e comparação dos números de mérito calculados.

Do ponto de vista da interacção, é possível manipular todos os modelos apresentados mudando a sua posição, orientação e tamanho (*pan*, *rotate*, *zoom*). Para além disso, quando vários modelos são apresentados em conjunto, é possível manipulá-los de uma forma sincronizada, i.e., manipular um dos modelos resulta na aplicação automática da mesma transformação a todos os outros modelos visíveis.

Quando um modelo colorido é visualizado, é possível utilizar uma sonda que permite ao utilizador obter o valor do número de mérito associado a um ponto particular da malha. Uma vez que os números de mérito só são calculados para locais específicos da malha (geralmente vértices, excepto para a qualidade dos triângulos),

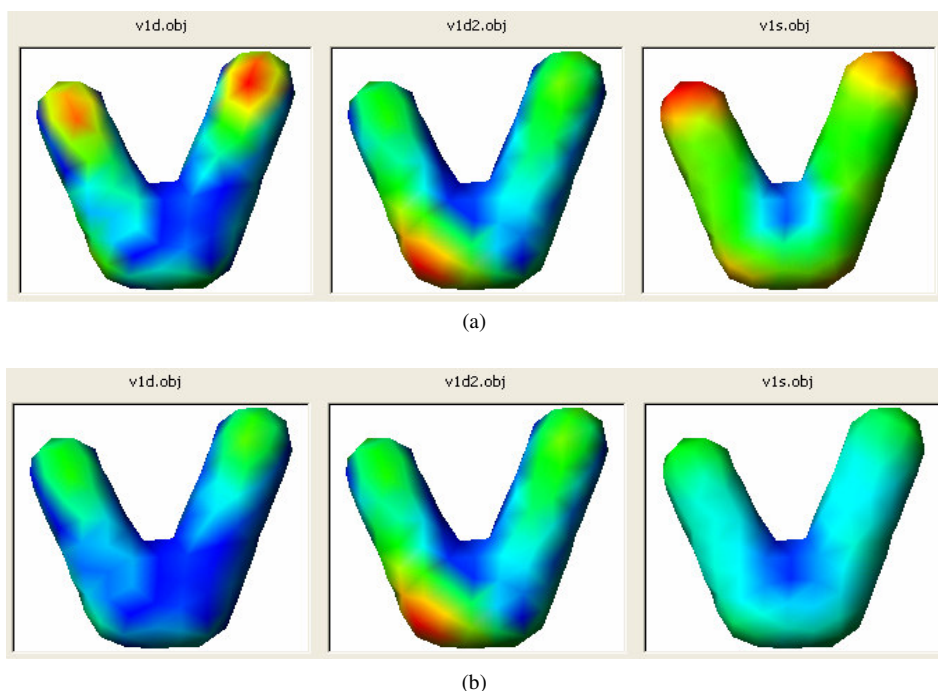


Figura 7 – Os resultados de um número de mérito calculado para diversos modelos são comparados lado-a-lado: a) usando escalas de cor individuais; b) usando uma escala de cor comum a todos os modelos

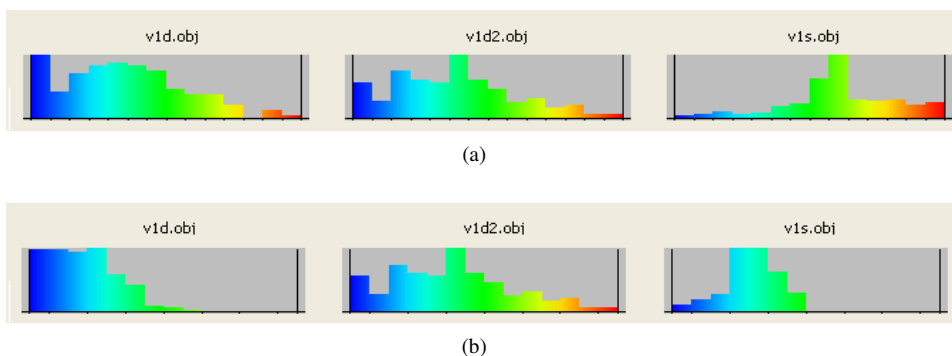


Figura 8 – Comparação de resultados usando histogramas: a) usando gamas de valores individuais; b) usando uma mesma gama de valores para todos os histogramas.

apenas os valores associados aos vértices ou faces serão fornecidos.

5.1 Modelo Colorido

Neste modo de visualização, é mostrado um modelo colorido de acordo com os valores obtidos com um dado número de mérito. Se o utilizador escolher também visualizar o modelo de referência ou o modelo para o qual o número de mérito foi calculado, estes aparecerão com a mesma posição, orientação e tamanho que o modelo colorido.

5.2 Modelo Original vs. Processado vs. Colorido

Este modo de visualização mostra o modelo original, o modelo processado e o modelo colorido de acordo com

os valores obtidos usando um dado número de mérito. Isto permite fazer uma comparação visual entre os modelos original e processado, nomeadamente nas zonas onde a incidência do erro (observada no modelo colorido) é maior. Apesar de o modelo colorido ser semelhante ao modelo original (quando se está a visualizar resultados de uma medida de diferença), a presença de cor pode mascarar algumas das suas características, razão pela qual o modelo original é também apresentado. A figura 6 mostra este modo de visualização.

5.3 Visualização de Resultados Lado-a-Lado

Neste modo de visualização é possível visualizar, lado-a-lado, os resultados de todos os números de mérito calculados para um mesmo modelo. Por exemplo, é possível visualizar, em simultâneo, os valores do desvio

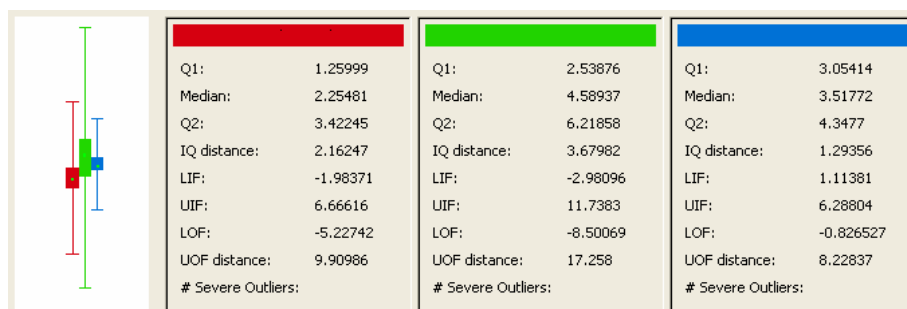


Figura 9 – Comparação de resultados usando diagramas de extremos e quartis.

geométrico e da curvatura Gaussiana e verificar se, por exemplo, o desvio geométrico é maior para as zonas onde a curvatura é também maior. Sendo assim, este modo de visualização pode ajudar o utilizador a encontrar semelhanças/relações no comportamento dos números de mérito e a ter uma ideia global das suas distribuições.

5.4 Comparação de Resultados Lado-a-Lado

Quando diversas versões processadas de um modelo são testadas, pode ser útil visualizar lado-a-lado os resultados obtidos com um mesmo número de mérito (por exemplo, os desvios geométricos para todos os modelos simplificados a partir de um mesmo modelo) de modo a comparar, por exemplo, as diferentes distribuições que resultam de cada tipo de processamento. No entanto, é necessário algum cuidado para comparar apenas dados que possam ser comparados. Primeiro, os resultados devem ser provenientes de um mesmo número de mérito, tendo o mesmo modelo como referência (no caso das medidas de diferença). Segundo, uma vez que todos os modelos coloridos usam a mesma escala de cor, representando 0 com azul e usando o vermelho para o valor máximo obtido em cada modelo, uma comparação directa de modelos coloridos apenas facultaria informação sobre o grau de incidência dos valores do número de mérito ao longo dos modelos, mas não pode ser usada para decidir qual o modelo apresentando valores mais elevados desse número de mérito.

Apesar de os valores numéricos poderem ser usados nesse sentido, a comparação das distribuições dos valores dos números de mérito não é fácil de executar. Sendo assim, é necessário usar uma escala de cor comum a todos os modelos, usando o azul para representar 0 mas passando a cor vermelha a representar o máximo valor obtido entre todos os modelos comparados, usando esse número de mérito. A figura 7 mostra uma comparação entre os resultados obtidos para três modelos usando um mesmo número de mérito. Note-se que na figura 7a) algumas das regiões dos três modelos parecem ter o mesmo valor associado. Na figura 7b), usando uma escala de cor comum a todos os modelos, verifica-se que o modelo do centro é o que apresenta maiores valores do número de mérito usado.

A comparação é também possível usando histogramas, representados lado-a-lado e, mais uma vez, deve ter-se em atenção que a comparação directa só pode ser efectuada quando estes são desenhados para uma mesma gama de valores. Este tipo de comparação pode ser útil, por exemplo, quando se pretende comparar a qualidade dos triângulos entre duas malhas, uma vez que permite verificar facilmente em que sentido é que a distribuição da qualidade se deslocou. Na figura 8 são apresentados os histogramas correspondentes aos modelos apresentados na figura 7. Note-se como a utilização de uma mesma gama de valores para todos os modelos (figura 8b) facilita a análise dos dados.

Finalmente, a comparação pode ser efectuada usando diagramas de extremos e quartis, como se verifica na figura 9.

6. CONCLUSÕES E TRABALHO FUTURO

Neste artigo foi apresentada uma ferramenta que permite a análise e comparação das características de malhas poligonais. Esta ferramenta proporciona um ambiente onde é possível trabalhar com diversos modelos e resultados ao mesmo tempo, e onde estão disponíveis diversas representações para os diferentes números de mérito que é possível calcular. É também possível escolher de entre diversos modos de visualização que permitem, por exemplo, visualizar diversas representações em simultâneo ou comparar resultados. Quando se comparam resultados provenientes de um mesmo número de mérito usando modelos coloridos (histogramas), a opção de usar uma escala de cor (gama de valores) comum é bastante importante.

O PolyMeCo pode ainda ser bastante melhorado. Nesse sentido seria importante acrescentar outros números de mérito (por exemplo, ver [Pébay03]) e novas representações e modos de visualização. Para além disso, torna-se também importante acrescentar outras escalas de cor de modo a possibilitar uma melhor representação de resultados com sinal.

Seria também interessante adicionar a hipótese de armazenar o conteúdo da área de trabalho, nomeadamente os resultados e representações geradas. Esta opção pode tornar-se útil, por exemplo, em situações em que se trabalha com modelos mais complexos e os resultados

demoram algum tempo a ser obtidos, ou quando se pretende guardar resultados para uma futura análise.

7. AGRADECIMENTOS

Os autores agradecem a Michel Röy por disponibilizar o código fonte do MeshDev.

O primeiro autor agradece à Unidade de Investigação 127/94 IEETA, da Universidade de Aveiro, a bolsa que tem permitido a realização do seu trabalho.

8. REFERÊNCIAS

- [Aspert02] N. Aspert, D. Santa-Cruz, e T. Ebrahimi, “Mesh: Measuring Errors between Surfaces Using the Hausdorff Distance”, *Proc. IEEE Int. Conf. Multimedia and Expo 2002*, Lausanne, Suíça, pp. 705-708, 2002.
- [Bergman95] L. Bergman, B. E. Rogowitz, e L. A. Treimish, “A rule-based tool for assisting color map selection”, *Proc. IEEE Visualization '95*, pp. 118-125, 1995.
- [Botsch02] M. Botsch, S. Steinberg, S. Bischoff, e L. Kobbelt, “Openmesh - a Generic and Efficient Polygon Mesh Data Structure”, *Proc. 1st OpenSG Symp.*, Darmstadt, Germany, Jan. 2002.
- [Cignoni98] P. Cignoni, C. Rocchini, e R. Scopigno, “Metro: Measuring Error on Simplified Surfaces”, *Computer Graphics Forum*, vol. 17, pp. 167-174, 1998.
- [Fox05] J. van der Zijp, “FOX TOOLKIT”, <http://www.fox-toolkit.org> (Abril 2005)
- [Meyer02] M. Meyer, M. Desbrun, P. Schröder, e A. H. Barr, “Discrete Differential-Geometry Operators for Triangulated 2-Manifolds”, in *Visualization and Mathematics III*, pp. 35-58, Springer-Verlag, 2003.
- [Pébay03] P. P. Pébay e T. J. Baker, “Analysis of Triangle Quality Measures”, *Mathematics of Computation*, vol. 72, pp. 1817-1839, 2003.
- [Roy04] M. Roy, S. Foufou, e F. Truchetet, “Mesh Comparison Using Attribute Deviation Metric”, *Int. Journal of Image and Graphics*, vol. 4(1), pp. 1-14, 2004.
- [Rusinkiewicz04] S. Rusinkiewicz, “Estimating Curvatures and Their Derivatives on Triangle Meshes”, *Proc. 2nd Int. Symp. on 3D Data Processing, Visualization, and Transmission*, pp. 486-493, 2004.
- [Zhou01] L. Zhou e A. Pang, “Metrics and Visualization Tools for Surface Mesh Comparison”, *Proc. SPIE 2001*, vol. 4302, *Visual Data Exploration and Analysis VIII*, pp. 99-110, 2001

Superfícies de Coons: Estudo comparativo de alguns algoritmos de geração e análise da sua qualidade

Eunice Oliveira
Dep. de Matemática, ESTG-IPLEIRIA
Morro do Lena – Alto do Vieiro
Apartado 4163
2411-901 Leiria
eunice@estg.ipleiria.pt

José Carlos Teixeira
Instituto de Telecomunicações – Coimbra
Dep. de Matemática, FCTUC
Apartado 3008
3001-454 Coimbra
teixeira@mat.uc.pt

Resumo

A modelação geométrica de superfícies de forma livre é uma área da maior importância, nomeadamente para a visualização científica e as aplicações industriais. A escolha do melhor método de geração de uma superfície nem sempre é uma tarefa fácil uma vez que, para além da adequação aos dados disponíveis e aos resultados pretendidos, deve ter-se em conta a flexibilidade da representação para futuras reutilizações e edição.

É frequente, o utilizador de um sistema de modelação assumir que a representação gerada corresponde a uma “boa solução”. No entanto, esta pode ser uma assunção precipitada, uma vez que os mesmos dados poderão gerar superfícies diferentes, em função do algoritmo utilizado.

Neste trabalho são apresentadas as superfícies de Coons, as suas representações na forma B-Spline e NURBS e os problemas associados a esta última formulação. Referem-se alguns algoritmos existentes, nomeadamente métodos com ajustamento dos pesos dos pontos de controlo na forma racional, e comparam-se as diferentes superfícies obtidas recorrendo a um teste estatístico. Deste modo, para além de se evidenciar o problema, apresenta-se uma metodologia de análise comparativa das superfícies geradas.

Palavras chave

Superfícies de Coons, NURBS, B-Splines, Teste de Kolmogorov-Smirnov.

1. INTRODUÇÃO

A crescente utilização de modelos digitais 3D para representar objectos reais ou imaginários e/ou fenómenos físicos e naturais impulsionou o desenvolvimento de sistemas de modelação cada vez mais versáteis e abrangentes, capazes de representar diversos tipos de curvas e de superfícies.

Para representar essa diversidade de curvas e superfícies muitos sistemas de modelação geométrica representam essas entidades na forma NURBS.

De entre os diversos métodos para gerar superfícies que se podem encontrar nesses sistemas de modelação geométrica estão as superfícies de Coons bilineares e bicúbicas. No entanto, estas superfícies podem apresentar uma forma irregular e oscilações quando geradas a partir de curvas racionais. Este problema pode ser ultrapassado realizando uma alteração ao algoritmo de Coons [Lin94].

A aplicação de diferentes algoritmos ao mesmo conjunto de curvas fronteiras gera, regra geral, diferentes superfícies. A comparação das superfícies resultantes não é por vezes possível ou suficiente apenas através de uma análise visual.

Por esse motivo, implementámos diversos algoritmos e centrámos a nossa atenção no estudo estatístico dos erros obtidos em cada superfície. Esta análise estatística permite avaliar qual das superfícies de Coons gerada se aproxima mais da superfície original e comparar, por exemplo, a qualidade das superfícies obtidas.

2. SUPERFÍCIES DE COONS BILINEARES

Uma superfície de Coons bilinear é uma superfície que interpola quatro curvas fronteiras como as representadas na Figura 1.

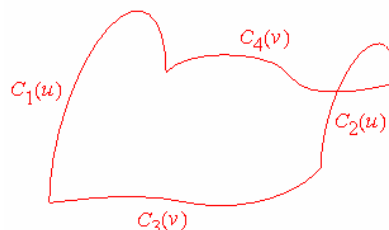


Figura 1 – Curvas fronteiras a interpolar.

Se considerarmos que as curvas em causa são $C_1(u)$, $C_2(u)$, $C_3(v)$ e $C_4(v)$ então estas curvas têm que verificar as seguintes condições:

$$\begin{aligned}
C_1(0) &= C_3(0) \\
C_1(1) &= C_4(0) \\
C_2(0) &= C_3(1) \\
C_2(1) &= C_4(1)
\end{aligned} \quad (1)$$

e além disso, $C_1(u)$ e $C_2(u)$, tal como $C_3(v)$ e $C_4(v)$, têm que ser curvas compatíveis¹.

A superfície de Coons bilinear pode ser obtida a partir de duas superfícies regradas e de uma superfície tensorial que interpola linearmente os vértices das curvas. Ou seja,

$$S(u, v) = R_1(u, v) + R_2(u, v) - T(u, v)$$

onde R_1 e R_2 são as superfícies regradas e T é a superfície que interpola linearmente os vértices das curvas.

2.1 Forma B-Spline de uma superfície de Coons bilinear

No caso das curvas a interpolar serem curvas B-Splines não racionais definidas por

$$C_k(u) = \sum_{i=0}^n N_{i,p}(u) P_i^k, \text{ com } k=1,2 \text{ e } u \in [0,1]$$

e

$$C_l(v) = \sum_{j=0}^m N_{j,q}(v) P_j^l, \text{ com } l=3,4 \text{ e } v \in [0,1]$$

então a superfície de Coons bilinear é da forma

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p;j,q}(u, v) P_{i,j}$$

com

$P_{i,j} = P_{i,j}^1 + P_{i,j}^2 - P_{i,j}^3$ para $i=0, \dots, n$ e $j=0, \dots, m$ onde $P_{i,j}^1$, $P_{i,j}^2$ e $P_{i,j}^3$ são, respectivamente, os pontos de controlo das superfícies R_1 , R_2 e T . Obviamente, a soma algébrica dos pontos de controlo das diferentes superfícies só é possível se estas forem compatíveis, isto é, se tiverem o mesmo grau, o mesmo número de pontos de controlo e se estiverem definidas sobre o mesmo vector de nós em cada direcção paramétrica. Para que estas condições sejam verificadas pode ser necessário recorrer a operações como a inserção de nós e/ou a elevação de grau.

A Figura 2 apresenta as superfícies R_1 , R_2 e T obtidas a partir das curvas fronteiras representadas na Figura 1. A superfície de Coons bilinear que interpola essas mesmas curvas fronteiras é apresentada na Figura 3.

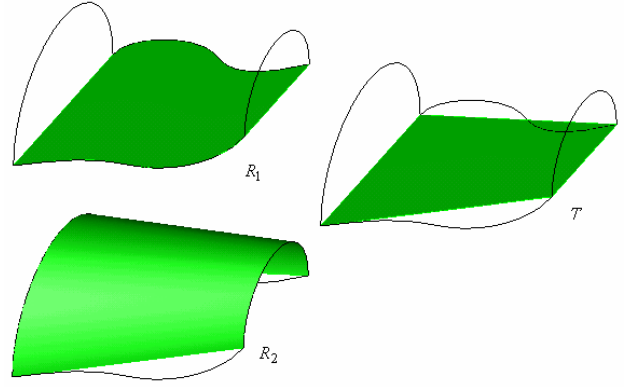


Figura 2 – Superfícies regradas e superfície tensorial.

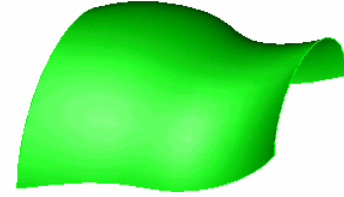


Figura 3 – Superfície de Coons bilinear.

2.2 Forma NURBS de uma superfície de Coons bilinear

No caso das curvas a interpolar serem NURBS, será possível pensar numa extensão do algoritmo ao espaço 4D, tal como Piegl e Tiller sugerem em [Piegl97].

Assim, se as curvas a interpolar forem definidas no espaço homogêneo por

$$C_k^w(u) = \sum_{i=0}^n N_{i,p}(u) (P_i^k)^w, \text{ com } k=1,2 \text{ e } u \in [0,1]$$

e

$$C_l^w(v) = \sum_{j=0}^m N_{j,q}(v) (P_j^l)^w, \text{ com } l=3,4 \text{ e } v \in [0,1]$$

e as condições (1) também se verificarem no espaço homogêneo, a superfície de Coons bilinear é dada por

$$\begin{aligned}
S^w(u, v) &= R_1^w(u, v) + R_2^w(u, v) - T^w(u, v) \\
&= \sum_{i=0}^n \sum_{j=0}^m N_{i,p;j,q}(u, v) P_{i,j}^w
\end{aligned}$$

com $P_{i,j}^w = (P_{i,j}^1)^w + (P_{i,j}^2)^w - (P_{i,j}^3)^w$ para $i=0, \dots, n$ e $j=0, \dots, m$. $(P_{i,j}^1)^w$, $(P_{i,j}^2)^w$ e $(P_{i,j}^3)^w$ são os pontos de controlo de R_1^w , R_2^w e T^w , respectivamente.

Tal como pretendido, a superfície resultante da extensão do algoritmo ao espaço 4D interpola as curvas fronteiras. No entanto, a superfície racional desejada, resultante da projecção daquela superfície no espaço tridimensional, pode apresentar irregularidades ou oscilações não esperadas e/ou não desejadas (Figura 4 e Figura 5).

¹ Duas, ou mais, curvas dizem-se compatíveis se têm o mesmo grau, se estão definidas sobre o mesmo vector de nós e, consequentemente, se têm o mesmo número de pontos de controlo.

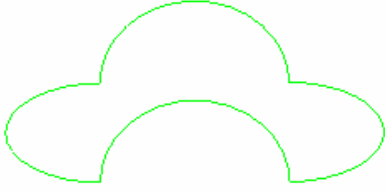


Figura 4 – Curvas fronteiras.

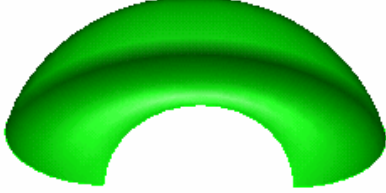


Figura 5 – Superfície racional de Coons bilinear.

Esta situação pode verificar-se porque a superfície resultante pode ter pontos de controlo com pesos negativos (devido à subtração realizada) ou porque a superfície tensorial T^w não reflecte convenientemente a distribuição de pesos das superfícies regradas.

De modo a contornar este problema, Lin e Hewitt [Lin94] propõem uma solução que consiste em alterar o algoritmo de Coons de modo a garantir que os pesos dos pontos de controlo sejam sempre positivos. A solução proposta baseia-se na ideia das superfícies de revolução e de varrimento para processar o valor dos pesos. Para os pontos de controlo fronteiros é aplicada a extensão ao espaço 4D, ou seja, $P_{i,j}^w = (P_{i,j}^1)^w + (P_{i,j}^2)^w - (P_{i,j}^3)^w$, para

$$\begin{cases} i=0,1 \\ j=0,\dots,m \end{cases} \quad \text{e} \quad \begin{cases} j=0,1 \\ i=0,\dots,n \end{cases}, \text{ garantindo assim a interpolação}$$

das curvas fronteiras. Para os restantes pontos de controlo (pontos de controlo interiores) é feita a seguinte adaptação $P_{i,j}^w = (w_{i,j}^1 w_{i,j}^2 (P_{i,j}^1 + P_{i,j}^2 - P_{i,j}^3), w_{i,j}^1 w_{i,j}^2)$, onde $w_{i,j}^1$ e $w_{i,j}^2$ são os pesos associados aos pontos de controlo $P_{i,j}^1$ e $P_{i,j}^2$, respectivamente. Deste modo, os pesos dos pontos de controlo da superfície resultante são sempre positivos.

A superfície apresentada na Figura 6 resulta da aplicação do algoritmo de Coons bilinear modificado às curvas representadas na Figura 4.

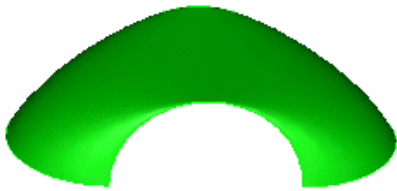


Figura 6 – Superfície racional de Coons bilinear com ajustamento.

3. SUPERFÍCIES DE COONS BICÚBICAS

As superfícies de Coons bicúbicas permitem construir superfícies que, para além de interpolarem quatro curvas fronteiras $C_k(u)$ e $C_l(v)$, com $k=1,2$ e $l=3,4$, interpolam também as derivadas parciais ao longo das curvas fronteiras $D_k(u)$ e $D_l(v)$, com $k=1,2$ e $l=3,4$ (Figura 7).

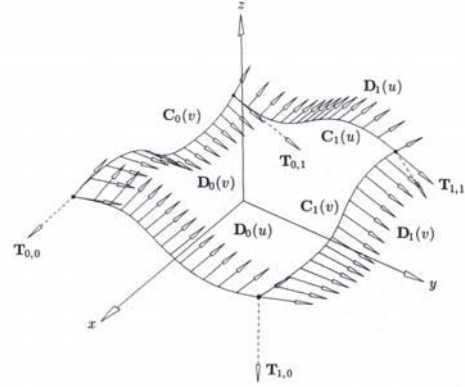


Figura 7 – Curvas fronteiras, vectores tangentes ao longo das curvas e derivadas mistas [Piegl97].

Deste modo, a superfície S , que se pretende obter, é limitada pelas curvas fronteiras $C_k(u)$ e $C_l(v)$, com $k=1,2$ e $l=3,4$, e as derivadas parciais ao longo destas fronteiras verificam as seguintes condições

$$\begin{aligned} D_1(u) &= S_v(u,0)^2 & D_2(u) &= S_v(u,1) \\ D_3(v) &= S_u(0,v) & D_4(v) &= S_u(1,v). \end{aligned} \quad (4)$$

Nas superfícies em causa, as condições de compatibilidade são mais exigentes. Além das condições referidas em (1), as curvas $C_k(u)$ e as derivadas $D_k(u)$, assim como, $C_l(v)$ e $D_l(v)$, têm que ser compatíveis, e é necessário também a compatibilidade das derivadas parciais mistas, ou seja,

$$\begin{aligned} T_{0,0} &= \frac{\partial S_v(0,0)}{\partial u} = \frac{\partial S_u(0,0)}{\partial v}, \\ T_{1,0} &= \frac{\partial S_v(1,0)}{\partial u} = \frac{\partial S_u(1,0)}{\partial v}, \\ T_{0,1} &= \frac{\partial S_v(0,1)}{\partial u} = \frac{\partial S_u(0,1)}{\partial v} \quad \text{e} \\ T_{1,1} &= \frac{\partial S_v(1,1)}{\partial u} = \frac{\partial S_u(1,1)}{\partial v}. \end{aligned} \quad (5)$$

O método de construção das superfícies de Coons bicúbicas é idêntico ao utilizado para construir as superfícies de Coons bilineares, dado que a superfície final é obtida a partir de três superfícies S_1 , S_2 e S_3 . No entanto, neste caso, S_1 , S_2 e S_3 são obtidas recorrendo à interpolação

² $S_u(u,v)$ e $S_v(u,v)$ representam as derivadas parciais de S em ordem a u e a v , respectivamente.

cúbica em substituição da interpolação linear utilizada nas superfícies de Coons bilineares.

A geração destas superfícies pode conduzir a um conjunto de problemas. Um deles tem a ver com a inexistência de informação sobre as derivadas parciais de primeira ordem ao longo das curvas fronteiras, o outro problema está relacionado com a ausência ou não compatibilidade das derivadas mistas.

Foram já propostas diversas soluções para ultrapassar estes obstáculos, algumas dessas soluções podem ser encontradas em [Rogers90], [Hoschek93] e [Farin90].

Neste trabalho optou-se por considerar o valor das derivadas mistas nos vértices das curvas a interpolar nulo e as derivadas parciais ao longo das curvas fronteiras foram calculadas por interpolação linear das derivadas parciais nesses mesmos vértices.

3.1 Forma B-Spline de uma superfície de Coons bicúbica

Considere-se que $C_k(u)$ e $C_l(v)$, com $k=1,2$ e $l=3,4$, são as curvas fronteiras a interpolar, definidas pelas equações (2) e considere-se que as derivadas parciais ao longo destas curvas são definidas por

$$D_k(u) = \sum_{i=0}^n N_{i,p}(u) Q_i^k, \text{ com } k=1,2 \text{ e } u \in [0,1]$$

e

$$D_l(v) = \sum_{j=0}^m N_{j,q}(v) Q_j^l, \text{ com } l=3,4 \text{ e } v \in [0,1].$$

Para obter as superfícies S_1 , S_2 e S_3 , Coons utilizou a interpolação cúbica de Hermite. Dado que neste trabalho se pretende determinar uma representação B-Spline para a superfície de Coons bicúbica, cada uma destas superfícies auxiliares deverá ser representada também nessa forma. Uma vez que uma superfície interpoladora de Hermite de terceiro grau pode ser representada como uma superfície de Bézier do mesmo grau [Lin94], pode recorrer-se a superfícies de Bézier para determinar uma forma B-Spline para S_1 , S_2 e S_3 .

Deste modo, S_1 é uma superfície de Bézier cúbica na direcção v , isto é, nessa direcção, o grau da superfície é três e o vector de nós é $\{0,0,0,1,1,1,1\}$. Na direcção u , a superfície S_1 é uma superfície que interpola as curvas $C_k(u)$, $k=1,2$, o grau e o vector de nós na direcção u são o grau e vector de nós das curvas $C_k(u)$. Assim, os pontos de controlo $B_{i,j}^1$ da superfície S_1 podem ser calculados do seguinte modo [Piegl97]:

$$B_{i,0}^1 = P_i^1$$

$$B_{i,1}^1 = P_i^1 + \frac{1}{3} Q_i^1$$

$$B_{i,2}^1 = P_i^2 - \frac{1}{3} Q_i^2$$

$$B_{i,3}^1 = P_i^2, \text{ com } i=0,\dots,n.$$

A superfície S_2 que interpola as curvas $C_l(v)$, $l=3,4$, é também uma superfície de Bézier cúbica, mas na direcção u . Logo, os pontos de controlo de S_2 são determinados de modo idêntico ao referido para S_1 .

A superfície S_3 é uma superfície de Bézier bicúbica definida por dezasseis pontos de controlo (Figura 8). Rogers e Adams [Rogers90] apresentam uma dedução para a relação entre os dezasseis pontos de controlo de uma superfície de Bézier bicúbica e os parâmetros que definem uma superfície de Coons bicúbica. Por exemplo, os pontos de controlo adjacentes ao vértice $S(0,0)$ podem ser calculados da seguinte forma:

$$B_{0,0}^3 = S(0,0) = C_1(0) = C_3(0)$$

$$B_{1,0}^3 = B_{0,0}^3 + \frac{1}{3} S_u(0,0)$$

$$B_{0,1}^3 = B_{0,0}^3 + \frac{1}{3} S_v(0,0)$$

$$B_{1,1}^3 = B_{0,0}^3 + \frac{1}{9} T_{0,0} + \frac{1}{3} S_u(0,0) + \frac{1}{3} S_v(0,0)$$

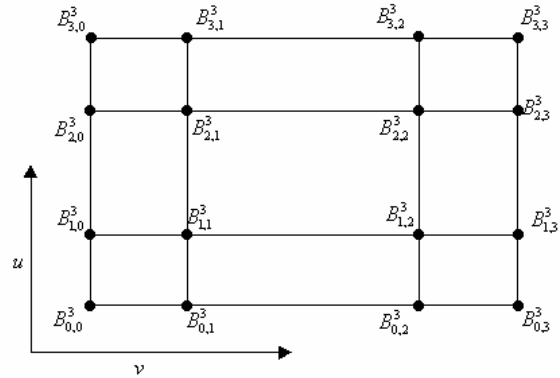


Figura 8 – Dezasseis pontos de controlo que definem uma superfície de Bézier bicúbica.

Após a determinação das superfícies S_1 , S_2 e S_3 , procede-se, se necessário, à elevação dos graus e ao refinamento de nós, de modo que estas superfícies sejam compatíveis. Nestas condições, a soma booleana entre as três superfícies é possível e os pontos de controlo $P_{i,j}$ da superfície de Coons bicúbica podem ser calculados a partir dos pontos de controlo das superfícies S_1 , S_2 e S_3 . Se $B_{i,j}^1$, $B_{i,j}^2$ e $B_{i,j}^3$ denotarem os pontos de controlo das superfícies compatíveis S_1 , S_2 e S_3 então $P_{i,j} = B_{i,j}^1 + B_{i,j}^2 - B_{i,j}^3$ para $i=0,\dots,n$ e $j=0,\dots,m$.

A Figura 9 e a Figura 10 representam, respectivamente, as superfícies de Bézier cúbicas nas direcções u e v e a superfície de Bézier bicúbica obtidas a partir das curvas fronteiras representadas na Figura 1. A superfície resul-

tante da soma booleana dessas três superfícies é apresentada na Figura 11.

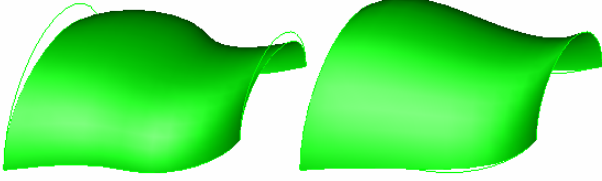


Figura 9 – Superfícies de Bézier cúbicas.

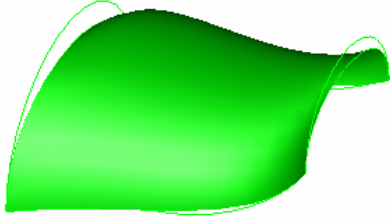


Figura 10 – Superfície de Bézier bicúbica.

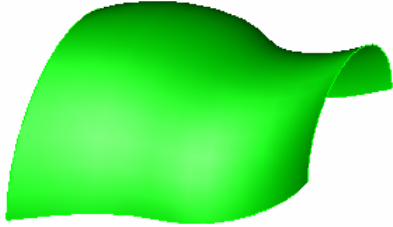


Figura 11 – Superfície de Coons bicúbica.

3.2 Forma NURBS de uma superfície de Coons bicúbica

Caso as curvas a interpolar sejam racionais, pode pensar-se numa extensão ao espaço 4D do algoritmo que gera as superfícies de Coons bicúbicas.

Considere-se que as curvas fronteiras são definidas no espaço homogêneo pelas equações (3) e as derivadas parciais ao longo dessas mesmas curvas fronteiras são representadas por

$$D_k^w(u) = \sum_{i=0}^n N_{i,p}(u) \left(Q_i^k \right)^w, \text{ com } k=1,2 \text{ e } u \in [0,1]$$

e

$$D_l^w(v) = \sum_{j=0}^m N_{j,q}(v) \left(Q_j^l \right)^w, \text{ com } l=3,4 \text{ e } v \in [0,1].$$

A extensão ao espaço homogêneo exige que as condições de compatibilidade, assim como as condições definidas em (1), (4) e (5) se verifiquem no espaço homogêneo.

Nestas condições as superfícies S_1 , S_2 e S_3 também são definidas no espaço homogêneo. Após realizar, se necessário, o processo de compatibilização destas superfícies, a superfície resultante é da forma

$$S^w(u, v) = S_1^w(u, v) + S_2^w(u, v) - S_3^w(u, v)$$

$$= \sum_{i=0}^n \sum_{j=0}^m N_{i,p;j,q}(u, v) P_{i,j}^w$$

onde $P_{i,j}^w = \left(P_{i,j}^1 \right)^w + \left(P_{i,j}^2 \right)^w - \left(P_{i,j}^3 \right)^w$ para $i = 0, \dots, n$ e $j = 0, \dots, m$, sendo $\left(P_{i,j}^1 \right)^w$, $\left(P_{i,j}^2 \right)^w$ e $\left(P_{i,j}^3 \right)^w$ os pontos de controlo S_1^w , S_2^w e S_3^w , respectivamente.

Tal como acontece para as superfícies racionais de Coons bilineares, a extensão do algoritmo ao espaço homogêneo nem sempre apresenta resultados satisfatórios, uma vez que os problemas mencionados para as superfícies racionais de Coons bilineares também se podem verificar para as superfícies racionais de Coons bicúbicas. Assim, se a superfície S_3^w não reflectir convenientemente a distri-

buição de pesos das superfícies S_1^w e S_2^w , a forma da superfície NURBS resultante pode ser inesperada. Além disso, não podemos garantir que todos os pontos de controlo da superfície resultante tenham pesos positivos, devido à subtracção realizada.

A Figura 12 mostra a superfície racional de Coons bicúbica obtida a partir das curvas fronteiras representadas na Figura 4, utilizando a extensão do algoritmo ao espaço homogêneo.



Figura 12 – Superfície racional de Coons bicúbica.

De modo a ultrapassar estes problemas, Lin e Hewitt [Lin94] propõem também para as superfícies racionais de Coons bicúbicas uma adaptação do algoritmo de modo que a extensão não seja realizada para todos os pontos de controlo. A solução apresentada é idêntica à proposta para as superfícies racionais de Coons bilineares. Assim, para os pontos de controlo fronteiros e para os pontos de controlo adjacentes é aplicada a extensão ao espaço 4D, donde

$$P_{i,j}^w = \left(P_{i,j}^1 \right)^w + \left(P_{i,j}^2 \right)^w - \left(P_{i,j}^3 \right)^w,$$

para $\begin{cases} i=0,1,n-1,n \\ j=0,\dots,m \end{cases}$ e para $\begin{cases} j=0,1,m-1,m \\ i=0,\dots,n \end{cases}$, garantindo assim a

interpolação das curvas fronteiras bem como as condições impostas para as derivadas parciais ao longo dessas mesmas curvas. De forma a garantir que os pesos dos restantes pontos de controlo sejam sempre positivos é feita a seguinte adaptação

$P_{i,j}^w = (w_{i,j}^1 w_{i,j}^2 (P_{i,j}^1 + P_{i,j}^2 - P_{i,j}^3), w_{i,j}^1 w_{i,j}^2)$, onde $w_{i,j}^1$ e $w_{i,j}^2$ são os pesos associados aos pontos de controlo $P_{i,j}^1$ e $P_{i,j}^2$, respectivamente.

4. METODOLOGIA DE ANÁLISE DAS SUPERFÍCIES GERADAS E ANÁLISE DOS RESULTADOS

4.1 Visão geral sobre a análise realizada

Para comparar as superfícies obtidas com os diversos algoritmos implementados foi construído um conjunto de superfícies de teste. Estas superfícies, obtidas por revolução ou por varrimento, servem como superfícies de referência (superfícies que se desejam obter). Para cada uma destas superfícies foram determinadas as suas curvas fronteiras e, a partir destas, geradas as superfícies de Coons, com e sem ajustamento.

Após a geração de todas as superfícies obtiveram-se amostras dos erros cometidos pela aplicação de cada algoritmo, por comparação de cada superfície de Coons com a superfície original.

As amostras de erros foram, seguidamente, sujeitas a um tratamento estatístico e finalmente foi feita a análise dos resultados.

4.2 Obtenção das amostras de erros

Para a realização dos testes foi considerada uma amostra de 100 pontos obtidos a partir de um conjunto aleatório de parâmetros (u_i, v_i) , $i = 1, \dots, 100$. Dado que, em qualquer dos algoritmos de Coons, as curvas fronteiras são interpoladas, na amostragem dos pontos não foram considerados pontos pertencentes a estas curvas, o que em termos de coordenadas paramétricas significa que $u_i, v_i \in]0,1[$, $i = 1, \dots, 100$.

Assim, denominando por $S(u, v)$ os pontos da superfície original e por $S_C(u, v)$ os pontos de uma superfície resultante da aplicação de um algoritmo de Coons, o erro cometido aquando da aplicação deste algoritmo é dado por

$$E = \text{distância}(S_C(u, v) - S(u, v)), \forall u, v \in [0,1].$$

Consequentemente, as amostras dos erros cometidos quando se utiliza cada um dos algoritmos de Coons, são dadas por

$$E_C = \text{distância}(S_C(u_i, v_i) - S(u_i, v_i)), i = 1, \dots, 100.$$

Para calcular a distância entre dois pontos tridimensionais foi utilizada a definição de distância euclideana. Assim, o erro (distância) entre um ponto da superfície original $P_1 = (x_1, y_1, z_1)$ e um ponto da superfície resultante $P_2 = (x_2, y_2, z_2)$ é dado por

$$\text{distância}(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

As amostras de erros obtidas podem ser comparadas do ponto de vista estatístico usando o teste de Kolmogorov-Smirnov [Guimarães97, Reis99].

4.3 Teste de Kolmogorov-Smirnov (K-S)

O teste de Kolmogorov-Smirnov, ou abreviadamente, teste K-S, permite comparar a distribuição de duas amostras de erros.

Este teste é usado para comparar quaisquer dois conjuntos de erros gerados por algoritmos diferentes, permitindo decidir qual dos dois algoritmos é melhor para uma determinada superfície.

Considerando duas amostras aleatórias independentes $X_1, X_2, \dots, X_n$ e $Y_1, Y_2, \dots, Y_n$ (no nosso caso são as amostras de erros), o teste K-S testa a hipótese da distribuição dos erros X 's obtidos aquando da aplicação de um algoritmo ser estatisticamente menor, igual ou maior que a distribuição dos erros Y 's obtidos aquando da aplicação de outro algoritmo para um determinado nível de significância.

A estatística de teste K-S para duas amostras define-se por $D = \max_{-\infty < x < +\infty} |S_{N_1}(x) - S_{N_2}(x)|$, (6)

$$D_1 = \max_{-\infty < x < +\infty} (S_{N_1}(x) - S_{N_2}(x)) \text{ e} \quad (7)$$

$$D_2 = \max_{-\infty < x < +\infty} (S_{N_2}(x) - S_{N_1}(x)) \quad (8)$$

onde $S_{N_1}(x)$ e $S_{N_2}(x)$ são as funções de distribuição empíricas³ dos X 's e dos Y 's, respectivamente. A equação (6) é denominada por teste bilateral e as equações (7) e (8) por testes unilaterais. Pelas definições de D_1 , D_2 e D podemos concluir que $D = \max(D_1, D_2)$.

Os valores de D_1 , D_2 e D sugerem quando se deve aceitar ou rejeitar a hipótese para um determinado nível de significância. Comparando D com o valor crítico do teste bilateral temos a indicação se as amostras de erros X 's e Y 's são estatisticamente idênticas. Caso D seja maior que o valor crítico do teste bilateral então as amostras de erros não são estatisticamente idênticas. Além disso, se D_1 é maior que o valor crítico do teste unilateral então a função de distribuição dos X 's é estatisticamente maior que a dos Y 's. Esta situação permite-nos concluir que, do ponto de vista estatístico, a amostra X tem uma ocorrência de erros menores mais elevada que a amostra Y . De forma idêntica, se D_2 é maior que o valor crítico do teste unilateral então a função de distribuição dos Y 's é estatisticamente maior que a dos X 's. Caso D_1 e D_2 sejam maiores

³ Sejam $E_{C_1} = \{e_1, \dots, e_{100}\}$ e $E_{C_2} = \{e_{101}, \dots, e_{200}\}$ duas amostras de erros obtidas por algoritmos diferentes e seja $E = \{e_1, \dots, e_{200}\}$ a reunião dos dois conjuntos anteriores. Para determinar a função de distribuição empírica S_{N_1} , para cada $e_i \in E$, calcula-se o número de ocorrências de erros menores ou iguais que e_i no conjunto E_{C_1} . A função de distribuição S_{N_2} calcula-se de modo idêntico mas para o conjunto E_{C_2} .

que o valor crítico do teste unilateral, o resultado é, em princípio, inconclusivo.

Embora se pudesse recorrer a resultados da estatística descritiva, como a média e o desvio padrão, para comparar os erros cometidos, esta análise poderia ser pouco esclarecedora, uma vez que apenas se comparariam as médias e não toda a distribuição de erros, como acontece no teste K-S.

Outros testes também muito utilizados para comparar médias são o teste T (que exige normalidade) [Guimarães97] e o teste de Wilcoxon (que exige que as amostras sejam provenientes de uma população com a mesma distribuição) [Guimarães97]. Estes testes, além das exigências referidas (as quais podem não se verificar nos casos que se pretendem comparar), também só permitem comparar a média e não toda a distribuição de erros como no teste K-S. Desta forma o teste K-S é preferível a qualquer um destes testes.

Anteriormente discutiu-se a forma de analisar a amostragem de erros dos vários métodos, por comparação de D_1 , D_2 e D com os valores críticos do teste unilateral e bilateral. Para um nível de significância de 0,05 e uma amostra de 100 elementos, os valores críticos dos testes unilateral (V.C.T.U.) e bilateral (V.C.T.B.) são 0,172534055 e 0,192333044, respectivamente [Guimarães97].

Foi com base nestes valores que se analisaram as superfícies de teste.

4.4 Exemplo de uma superfície de teste

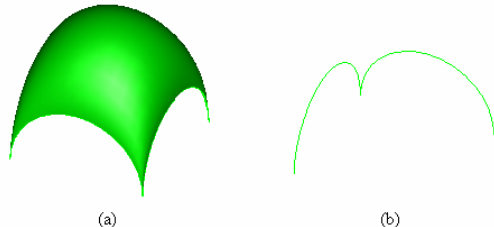


Figura 13 – Superfície original e curvas geradoras.

A superfície de teste representada na Figura 13 (a) consiste numa superfície de varrimento obtida a partir das curvas perfil e trajectória racionais representadas na Figura 13 (b). Esta superfície é uma superfície racional uma vez que as curvas perfil e trajectória também o são.

Na Figura 14 (a) apresentam-se as curvas fronteiras da superfície de teste e na Figura 14 (b), (c), (d) e (e) estão representadas, respectivamente, as superfícies racionais de Coons bilineares sem e com ajustamento e as superfícies racionais de Coons bicúbicas sem e com ajustamento.

Como facilmente se pode verificar por observação das Figuras 14 (b) e (d), as superfícies resultantes da aplicação dos algoritmos de Coons sem ajustamento apresentam irregularidades não esperadas, enquanto as superfícies obtidas com o ajustamento têm uma forma idêntica à da superfície original. No entanto, apenas com uma análise visual não é possível distinguir qual das duas superfícies de Coons com ajustamento se aproxima mais da

superfície original, daí a importância de recorrer a uma análise mais detalhada para poder responder a esta questão.

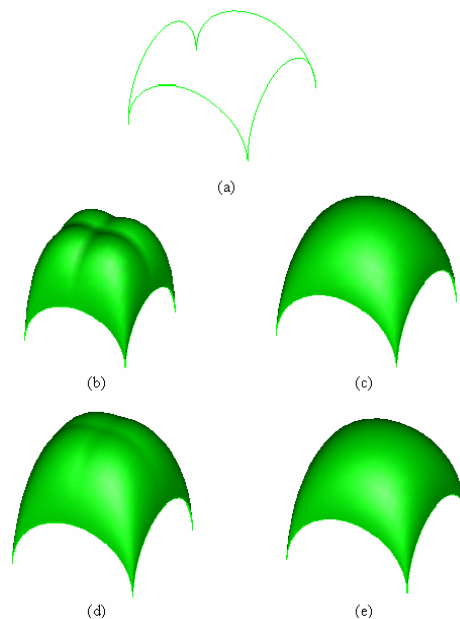


Figura 14 – (a) Curvas fronteiras. (b) e (c) Superfícies racionais de Coons bilineares sem e com ajustamento. (d) e (e) Superfícies racionais de Coons bicúbicas sem e com ajustamento.

Nesta análise construiu-se uma tabela que mostra os valores da estatística K-S e outra tabela que resume os resultados, mostrando a relação entre dois algoritmos (isto é, se um algoritmo tem uma distribuição de erro menor que o outro).

As tabelas 1 e 2 apresentam os resultados finais dos testes K-S realizados para esta superfície de teste.

	D_1	D_2	D
Bilinear / Bilinear com ajust.	0,17	1	1
Bilinear / Bicúbica	0	0,45	0,45
Bilinear / Bicúbica com ajust.	0,01	0,79	0,79
Bilinear com ajust. / Bicúbica	1	0	1
Bilinear com ajust. / Bicúbica com ajust.	1	0	1
Bicúbica / Bicúbica com ajust.	0,04	0,59	0,59

Tabela 1 – Resultados dos testes K-S.

	Bilinear	Bilinear com ajust.	Bicúbica	Bicúbica com ajust.
Bilinear	—	<	<	<
Bilinear com ajust.	—	—	>	>
Bicúbica	—	—	—	<
Bicúbica com ajust.	—	—	—	—

Tabela 2 – Comparação das distribuições de erros.

Para as superfícies racionais de Coons bilineares os resultados destes testes confirmam os resultados visuais, uma vez que ao comparar-se a superfície racional de Coons bilinear com as três restantes, a primeira apresenta uma distribuição de erros menor em relação a qualquer uma das outras (Tabela 2). Assim, a superfície obtida com o algoritmo de Coons bilinear apresenta erros mais elevados que as restantes. Consequentemente, para esta superfície de teste, o algoritmo de Coons bilinear mostra-se o menos indicado.

Relativamente à superfície racional de Coons bicúbica, embora se aproxime mais da superfície original que a superfície racional de Coons bilinear, é preterida a favor dos dois métodos com ajustamento. Nos testes de comparação da superfície racional de Coons bicúbica com os métodos com ajustamento foi obtido um valor para D superior ao V.C.T.B., permitindo afirmar que, estatisticamente, as distribuições de erros são diferentes. No teste “Bilinear com ajust./Bicúbica” obteve-se $D=1$ e $D_2=0$. Como $D_1 > \text{V.C.T.U.}$ e $D_2 < \text{V.C.T.U.}$ então a distribuição de erros da superfície racional de Coons bilinear com ajustamento é maior que a distribuição de erros da superfície racional de Coons bicúbica (Tabela 2). Assim, esta superfície apresenta erros mais elevados que a primeira e consequentemente o método de Coons com ajustamento é preferível. Quando se comparam as superfícies racionais de Coons bicúbicas sem e com ajustamento as conclusões são idênticas às anteriores.

As superfícies racionais de Coons obtidas com ajustamento são um bom exemplo para demonstrar a importância do estudo estatístico realizado, uma vez que as superfícies obtidas são aparentemente idênticas. Apenas com uma análise visual não é possível determinar qual das duas se aproxima mais da superfície original.

Aplicando o teste K-S às superfícies racionais de Coons bilineares e bicúbicas com ajustamento (por esta ordem), obtemos $D_1=1$ e $D_2=0$. Logo $D=1 > \text{V.C.T.B.}$ o que permite concluir que existe evidência estatística para afirmar que as distribuições de erros não são iguais, ou dito de outra forma, a análise das distribuições dos erros permite afirmar que os modelos gerados não são iguais. Além disso, $D_1 > \text{V.C.T.U.}$ e $D_2 < \text{V.C.T.U.}$ donde a distribuição de erros da superfície racional de Coons bilinear com ajustamento é maior que a distribuição de erros da superfície racional de Coons bicúbica com ajustamento (Tabela 2). Assim, conclui-se que destas superfícies a que mais se aproxima da superfície original é a superfície racional de Coons bilinear com ajustamento.

Após uma análise geral dos dados recolhidos nos testes K-S realizados para esta superfície de teste pode-se concluir que a superfície racional de Coons bilinear com ajustamento é a que mais se aproxima da superfície original.

Se se desejar uma comparação mais detalhada é ainda possível analisar os gráficos das distribuições de erros, como o que se mostra na Figura 15.

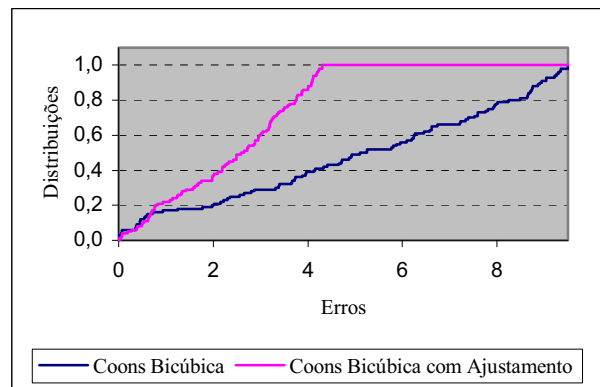


Figura 15 – Gráfico das distribuições de erros.

A análise destes gráficos pode ser bastante importante nos casos em que os valores obtidos pelo teste K-S sejam estatisticamente inconclusivos.



Figura 16 – Superfície racional de Coons bilinear com ajustamento.



Figura 17 – Superfície racional de Coons bicúbica com ajustamento.

Esta situação verifica-se, por exemplo, quando se comparam as superfícies racionais de Coons bilinear com ajustamento (Figura 16) e bicúbica com ajustamento (Figura 17) geradas a partir das curvas fronteiras apresentadas na Figura 4.

Neste caso, o valor de D é menor que o V.C.T.B., o que significa que não existe evidência estatística para afirmar que as distribuições de erros são diferentes. Saliente-se o facto deste resultado não indicar que algoritmos diferentes geram modelos iguais, o que apenas se pode concluir é que as diferenças entre as amostragens de erros não são estatisticamente relevantes. Nestas situações apenas uma análise mais detalhada das distribuições de erros pode conduzir a mais algumas conclusões.

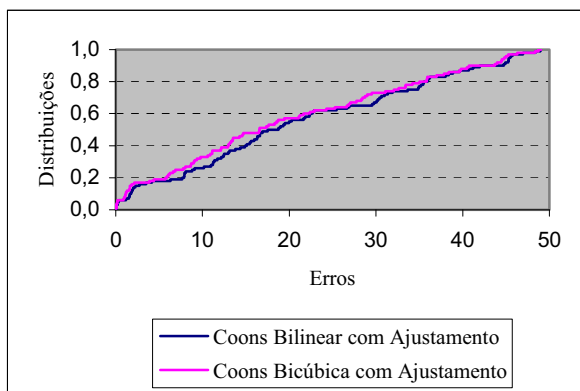


Figura 18 – Gráfico das distribuições de erros.

Para o caso concreto, a análise do gráfico das distribuições (Figura 18) permite concluir que a distribuição da superfície racional de Coons bicúbica com ajustamento é na maioria das vezes superior à distribuição da superfície racional de Coons bilinear com ajustamento. Assim, pelas amostragens realizadas somos levados a concluir que os erros da superfície racional de Coons bicúbica com ajustamento são quase sempre menores que os da superfície racional de Coons bilinear com ajustamento e consequentemente a superfície racional de Coons bicúbica com ajustamento aproxima-se ligeiramente mais da superfície original que a superfície racional de Coons bilinear com ajustamento.

5. CONCLUSÕES

Os testes realizados com as superfícies de Coons permitem confirmar que, em situações onde as curvas fronteiras são racionais, a superfície resultante da aplicação deste método pode apresentar oscilações não esperadas. Este facto resulta de uma desigual distribuição de pesos nas três superfícies que contribuem para a formação de uma superfície racional de Coons.

Para as superfícies de teste utilizadas verificou-se que a alteração no cálculo dos pesos dos pontos de controlo interiores, proposta por Lin e Hewitt [Lin94], evita a formação das referidas oscilações. No entanto, esta constatação não nos permite generalizar este resultado uma vez que apenas foram utilizadas superfícies de teste obtidas por revolução e por varrimento.

Relativamente às superfícies de Coons bicúbicas, estas apresentam ainda problemas subjacentes à definição de alguns parâmetros necessários à sua construção. Um dos problemas consiste na inexistência de informação sobre as derivadas parciais de primeira ordem ao longo das curvas fronteiras, o outro problema está relacionado com a ausência ou não compatibilidade das derivadas mistas.

Alguns dos resultados provenientes da análise estatística, realizada com o teste K-S, permitem concluir que a superfície de Coons bilinear se aproxima mais da superfície original que a superfície de Coons bicúbica. Este resultado justifica-se com o facto de na construção das superfícies de Coons bicúbicas se ter considerado o valor das derivadas mistas nos vértices das curvas a interpolar

nulo e as derivadas parciais ao longo das curvas fronteiras ser calculado por interpolação linear das derivadas parciais nesses mesmos vértices.

É ainda importante salientar que nem sempre a superfície resultante do ajustamento proposto por Lin e Hewitt é a que mais se aproxima da superfície original, como acontece com algumas das superfícies de teste. Assim, para o caso em que as curvas a interpolar são racionais, não é possível concluir qual dos algoritmos de Coons (com ou sem ajustamento) é mais apropriado. Tal escolha depende das características que se pretendam para a superfície final, uma vez que as oscilações resultantes podem ser, ou não, um problema.

6. TRABALHO FUTURO

Com o objectivo de poder generalizar os resultados obtidos com as superfícies de teste obtidas por revolução e por varrimento, pretendemos dar continuidade a este trabalho testando o algoritmo de Coons com ajustamento para outros tipos de superfícies.

Pretendemos ainda, desenvolver e testar métodos de geração de superfícies de Coons onde as derivadas parciais ao longo das curvas fronteiras e as derivadas mistas nos vértices dessas curvas possam ser calculadas por processos diferentes dos utilizados neste trabalho, nomeadamente por alguns dos métodos referidos em [Farin90] e [Hoschek93]. Deste modo, seria possível ter um maior controlo sobre a forma das superfícies de Coons bicúbicas e também obter uma maior diversidade de formas para este tipo de superfícies.

7. REFERÊNCIAS

- [Farin90] Farin, G., *Curves and Surfaces for Computer Aided Design-A Practical Guide*, 2ª edição, Academic Press, Inc, 1990.
- [Guimarães97] Guimarães, Rui Campos e Cabral, José A. Sarsfield, Estatística, McGraw-Hill, Portugal, 1997.
- [Hoschek93] Hoschek, J. e Lasser, D., *Fundamentals of Computer Aided Geometric Design*, A. K. Peters, Wellesley, MA 1993.
- [Lin94] Lin, F. e Hewitt, W.T., *Expressing Coons-Gordon surfaces as NURBS*, Comp. Aided Des., Vol.16, N°2, 1994, pp. 145-155.
- [Oliveira03] Oliveira, Eunice, *Modelação de superfícies de forma livre*, Dissertação de Mestrado sob orientação de J. C. Teixeira, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, Dezembro 2003.
- [Piegl97] Piegl, L. e Tiller, W., *NURBS Book*, 2ª edição, Springer, 1997.
- [Reis99] Reis, Elizabeth; Melo, Paulo; Andrade, Rosa e Calapez, Teresa, *Estatística Aplicada*, Vol.2, Edições Sílabo, LDA, 1999.
- [Rogers90] Rogers, D. F. e Adams, J. A., *Mathematical Elements for Computer Graphics*, 2ª edição, McGraw-Hill, New York, 1990.

Algoritmo de Subdivisão de Superfícies Deformáveis para Execução em GPU

Fernando Pedro Birra

Manuel Próspero dos Santos

Departamento de Informática
Faculdade de Ciências e Tecnologia da
Universidade Nova de Lisboa
Quinta da Torre, 2825-114 Caparica
{fpb,ps}@di.fct.unl.pt

Sumário

A subdivisão recursiva de superfícies é uma técnica que permite obter uma superfície final suave, com elevado nível de detalhe, partindo de uma superfície inicial, de controlo, muito pouco detalhada. Se é verdade que, por um lado, esta técnica se torna muito apetecível efectuando simulações demoradas apenas numa malha de controlo, por outro, se essa mesma malha tiver que ser subdividida pelo CPU antes de ser passada ao GPU, para visualização, então perde-se grande parte ou até mesmo a totalidade das vantagens na sua utilização. Este artigo propõe um algoritmo de subdivisão de malhas regulares formadas por triângulos, agrupados dois a dois em quadriláteros. O algoritmo é executado na sua totalidade no GPU, aliviando o CPU para as tarefas de simulação. O exemplo que serve de ilustração a este algoritmo é o da simulação de superfícies deformáveis (p.ex. tecidos) com variação dinâmica do nível de detalhe. Trata-se de um domínio de aplicação onde todo o tempo de CPU se revela necessário para a avaliação do modelo mecânico subjacente à dinâmica da superfície deformável.

Palavras-chave

Computação no GPU, subdivisão, simulação, superfícies deformáveis, níveis de detalhe.

1. INTRODUÇÃO

A subdivisão de superfícies é reconhecida como sendo a técnica que preenche o vazio entre malhas de triângulos simples e os sofisticadas representações com NURBS, sendo usadas sempre que se pretende aliar a eficiência das malhas de triângulos com a qualidade superior das superfícies de NURBS.

Exemplos de esquemas de subdivisão abundam na literatura, tendo alguns deles sido exaustivamente estudados e analisados, tais como a subdivisão de Loop [Loop87], Doo Sabin [Doo78] ou Catmull-Clark [Catmull78]. Ao subdividir-se uma superfície, de forma recursiva, vão-se obtendo superfícies cada vez mais refinadas e que, caso o método de subdivisão seja bom, se vão aproximando de uma superfície limite suave.

Não menosprezando outras aplicações tão ou mais válidas, a utilização da subdivisão recursiva de superfícies é particularmente tentadora no caso da simulação de tecidos ou outras superfícies deformáveis.

A simulação absorve quase todo o tempo de CPU disponível e passa, em primeiro lugar, pela avaliação de um modelo matemático complexo que descreve o comportamento mecânico do tecido, traduzindo-se no cálculo de forças aplicadas em diversos pontos de amostragem. Normalmente aproximam-se os tecidos com malhas de polígonos formando os seus vértices um sistema de partículas. Posteriormente, obtém-se novas

posições para os vértices da malha resolvendo um sistema de equações diferenciais.

A complexidade temporal dos algoritmos muito raramente é linear no número de partículas. Por estas razões é natural que se proceda à simulação de uma malha menos refinada.

Em [Bridson02] foi usado um passo de pós-processamento para suavizar as malhas usadas na simulação de tecidos. O algoritmo de subdivisão escolhido foi a subdivisão de Loop modificada [Loop01]. De modo a lidar com o problema das intersecções que se podem formar durante cada passo de subdivisão, as posições das partículas vão sendo ajustadas, caso necessário, de modo a garantir que a malha final de cada passo esteja isenta de intersecções com objectos sólidos. A malha inicial que é fornecida ao algoritmo de subdivisão é garantidamente uma malha isenta de intersecções ou seja, de colisões.

No entanto, a utilização de subdivisão de superfícies não garante, por si só, que a malha final possa capturar os detalhes que se obteriam usando uma malha mais refinada na simulação. É fundamental a utilização de malhas com suficiente nível de detalhe durante a simulação em determinadas zonas do tecido. Neste sentido é vantajoso usar-se simultaneamente um esquema de variação dinâmica do nível de detalhe, tal como em [Birra04].

Com o advento de processadores gráficos (GPUs) programáveis e com alto desempenho, tais como os disponíveis na quase generalidade das placas gráficas comercializadas hoje em dia para o mercado de jogos, torna-se possível explorar ainda mais as potencialidades associadas ao uso das superfícies de subdivisão. De facto, ao migrar os algoritmos de subdivisão de superfícies para os GPUs liberta-se o CPU desse fardo podendo o seu esforço ser inteiramente dedicado à simulação.

2. TRABALHO RELACIONADO

Durante muito tempo pensou-se que as superfícies obtidas por subdivisão recursiva não poderiam ser avaliadas analiticamente para valores arbitrários dos seus parâmetros. No entanto, Stam [Stam98] provou que tal é possível e usou como exemplo superfícies de Catmull-Clark. Apesar de se poder implementar a técnica num *fragment program*, o tratamento de todos os possíveis casos, relativamente a vértices extraordinários, fronteiras e cantos, resultaria num programa demasiado complexo [Zorin02].

Outros autores [Bolz02, Bolz03, Brickhill01] exploraram uma propriedade importante que permite a implementação de um algoritmo iterativo. Dado que a operação de *tessellation* de uma combinação linear de funções base é equivalente a uma combinação linear de *tesselations* de funções base, é possível, para qualquer nível de subdivisão, exprimir as posições dos vértices desse nível em função de combinações lineares dos pontos da malha inicial. Por outras palavras, o vector formado pelos vértices da superfície subdividida após vários passos de subdivisão recursiva pode escrever-se como o produto de potências da matriz de subdivisão pelo vector de pontos iniciais.

[Brickhill01] aplicou este princípio à subdivisão de malhas de triângulos usando a técnica de Loop, destinando-se a implementação à Playstation 2. Em [Bolz02] o mesmo princípio foi aplicado a quadriláteros, usando superfícies de Catmull-Clark. A primeira implementação corria no CPU, sendo posteriormente convertida para correr directamente no GPU, numa placa GeForce FX [Bolz03].

Comum às técnicas acima referidas, está a sua grande versatilidade, pois elas operam sobre malhas de topologia arbitrária, onde a valência dos vértices pode ser qualquer. No entanto, de modo a reduzir a complexidade da implementação é necessário um passo de pré-processamento para as malhas de topologia arbitrária. A ideia é a de isolar os vértices extraordinários (isto é, com valência diferente de 6 no caso da subdivisão de Loop ou diferente de 4 no caso das superfícies de Catmull-Clark) de tal forma que, para cada *patch* a subdividir, exista apenas um tal vértice não regular. Este processo requer a aplicação de 1 ou mais passos de subdivisão uniformemente em toda a malha (conforme o tipo de subdivisão em questão). Note-se que a aplicação de um passo de subdivisão apenas introduz vértices regulares permitindo assim o isolamento dos vértices irregulares. De seguida os

patches que incorporam vértices irregulares são classificados de acordo com o caso específico.

Ora, é precisamente este pré-processamento da malha que impede a sua fácil utilização num contexto em que a malha original varia de forma dinâmica ao longo do tempo. No exemplo aqui em estudo, a utilização da variação dinâmica do nível de detalhe da malha usada na simulação de tecidos impede que estas técnicas se possam usar de forma simples e eficiente.

Por outro lado, a aplicação de um elevado número de passos de subdivisão de uma única vez pode ser indesejada, pois não permite *feedback* exterior nos passos intermédios. Tomando como exemplo a animação de tecidos, a subdivisão da malha usada na simulação pode criar vértices que introduzam intersecções com outros objectos ou até mesmo com ela própria. É mais simples tratar estas novas intersecções passo a passo (o seu número é mais reduzido) do que efectuar o procedimento para um número potencialmente bastante superior. Por exemplo, nas superfícies de Catmull-Clark, em cinco passos de subdivisão um quadrilátero dá origem a 256 quadriláteros na superfície final.

Assim, propomos neste artigo um algoritmo de subdivisão para ser usado no contexto da animação de superfícies deformáveis com variação do nível de detalhe. O nosso algoritmo, corre integralmente no GPU e aplica-se a malhas 4-8 regulares. Apesar de seguirmos uma implementação recursiva tradicional, de facto a nossa abordagem pode considerar-se mista, pois em cada passagem efectuamos dois passos de subdivisão.

O esquema de subdivisão do qual partimos é o da subdivisão 4-8 [Velho01]. No entanto as regras foram ligeiramente adaptadas por questões de eficiência e simplicidade que veremos adiante.

3. ESQUEMA DE SUBDIVISÃO

Nesta secção ilustraremos os passos que nos conduziram à obtenção do nosso esquema modificado de subdivisão de malhas 4-8.

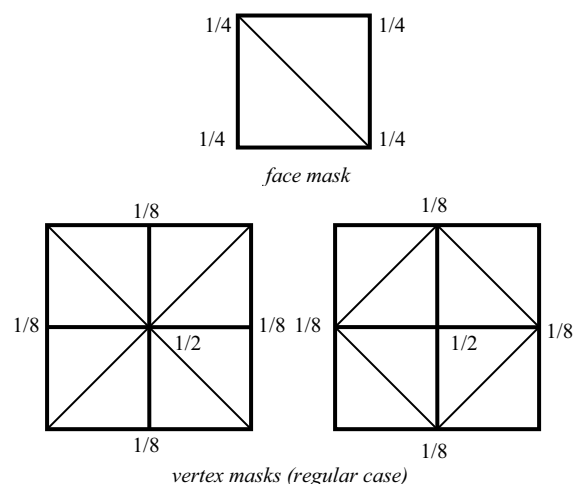


Figura 1 – Máscaras das regras de subdivisão 4-8 (caso regular)

As malhas 4- k [Velho00] são também conhecidas pelo termo *triquad meshes* e podem ser descritas por um agrupamento de triângulos, dois a dois, de modo a formarem um quadrilátero. Cada um destes quadriláteros define um bloco base e a operação de subdivisão elementar é a bissecção da sua aresta interior.

Fundamentalmente, os esquemas de subdivisão de superfícies decompõem-se em dois tipos de regras:

- *Face rules* – Adicionam novos vértices à superfície.
- *Vertex rules* – Filtram as posições dos vértices existentes.

Na Figura 1 podemos observar as máscaras associadas a cada uma das regras, acima mencionadas, para o caso de malhas 4-8 regulares (caso particular das malhas 4- k). As máscaras para as fronteiras de malhas fechadas serão analisadas mais adiante. A operação de bissecção corresponde à aplicação da *face rule* usando a primeira das máscaras da Figura 1 (*face mask*). Uma característica importante da subdivisão 4-8 é o facto da aplicação de dois passos de subdivisão corresponderem à quadrissecção dos quadriláteros que formam os blocos elementares, como se pode observar na Figura 2. Este facto é explorado na nossa implementação e conduz a programas muito simples, pois a disposição dos dados na memória do processador gráfico é muito natural. Refira-se, a propósito, que um dos grandes problemas na transposição de algoritmos para executarem nos GPUs é precisamente o da organização particular dos dados em memória. Na figura, os círculos cinzentos indicam os vértices introduzidos no respectivo passo da subdivisão.

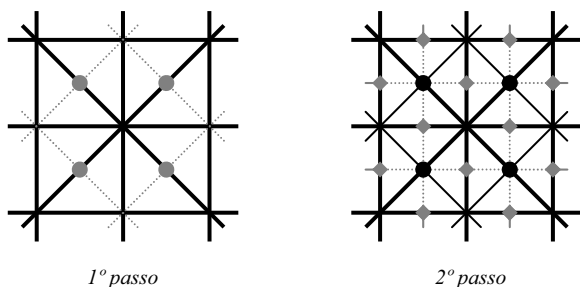


Figura 2 – Dois passos de subdivisão equivalem a uma quadrissecção dos blocos iniciais

Uma rápida análise da Figura 2 revela que, após os dois passos, podemos classificar os vértices resultantes da malha subdividida em três tipos:

- Tipo 1 - Vértices que substituem os vértices presentes na malha inicial. Estes vértices são o resultado da acumulação da dupla aplicação das *vertex rules*.
- Tipo 2 - Vértices que resultam da filtragem, no segundo passo, dos vértices criados durante o 1º passo (círculos pretos na Figura 2).
- Tipo 3 - Vértices introduzidos no segundo passo. Estes vértices estão assinalados a cinzento no lado direito da Figura 2.

Resta agora deduzir as máscaras para o esquema de subdivisão resultante da aplicação dos dois passos de seguida. O que necessitamos é de exprimir o cálculo das posições finais de cada tipo de vértice (dentro da classificação anterior) em função dos vértices da malha inicial.

A dedução das máscaras que permitem calcular cada um dos tipos de vértices é mostrada no Anexo A, apresentando-se aqui apenas o resultado final.

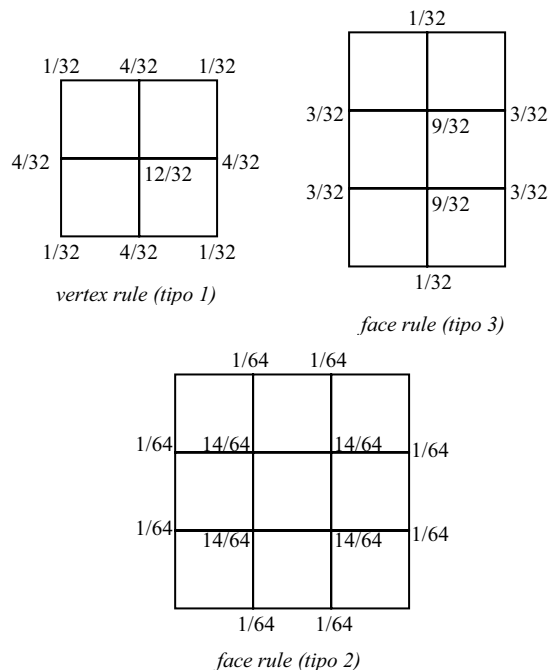


Figura 3 - Regras finais de subdivisão para o esquema de subdivisão 4-8 em dois passos

As regras são semelhantes, embora com pesos diferentes, às da subdivisão de Catmull-Clark para quadriláteros. No entanto, o suporte para algumas das regras é maior, no sentido de abranger mais pontos, o que se traduz numa desvantagem do ponto de vista da eficiência. Na secção seguinte mostraremos a nossa abordagem para contornar esta situação.

De modo a completar o esquema falta ainda deduzir as regras de subdivisão nos pontos fronteira e nos cantos. A dedução destas regras é efectuada tendo em conta o efeito final pretendido.

No contexto da simulação de tecidos ou outras superfícies deformáveis, o objectivo do uso da subdivisão de superfícies é simples: refinar, no final da simulação, a malha de controlo de forma a evitar a aparência angular que transparece quando surgem dobras ou rugas na superfície.

As regras de subdivisão a aplicar nas fronteiras da superfície terão que respeitar uma condição essencial, que é a de não provocarem o encolhimento do tecido a toda a sua volta.

Para além do mais, pode ser vantajoso refinar várias superfícies adjacentes (ou coladas nos bordos) de forma

independente sem que, na fronteira comum da superfície final subdividida, surjam buracos.



Figura 4 – Máscaras para as fronteiras

Com base nestas considerações, as nossas regras de subdivisão para as fronteiras resultam em combinações lineares onde apenas estão envolvidos os pontos da superfície original também situados na fronteira. Estas regras podem ser observadas na Figura 4.

No caso dos cantos, eles são inalterados pelas regras de subdivisão.

Na secção seguinte analisaremos a respectiva implementação no GPU juntamente com as modificações introduzidas no esquema de modo a obter programas mais eficientes e mais simples de implementar.

4. IMPLEMENTAÇÃO NO GPU

Os GPUs correntes possuem dois tipos distintos de processadores capazes de serem programados externamente:

- *Vertex processors* – permitem executar pequenos programas (*vertex programs* ou *vertex shaders*) que são activados por cada vértice das primitivas gráficas a desenhar.
- *Fragment processors* – permitem executar pequenos programas (*fragment programs* ou *pixel shaders*) que são activados por cada fragmento a escrever no framebuffer.

A grande vantagem dos processadores de fragmentos, no contexto da programação genérica nos GPUs, é o facto de poderem ser activados pelo CPU para operarem sobre grandes colecções de dados, gastando muito pouca largura de banda e tempo de CPU. A técnica usual consiste em desenhar um quadrilátero num framebuffer invisível (*offscreen*). Durante a fase de geração dos fragmentos (ou rasterização) será executado um programa, dentro dos processadores de fragmentos, por cada fragmento gerado.

Os vértices do quadrilátero poderão eventualmente activar a execução de programas nos processadores de vértices. O resultado de cada uma destas execuções é um vértice, provavelmente transformado. Numa situação em que se pretende processar um grande volume de dados não é de todo conveniente usar os processadores de vértices porque, por cada execução, é necessário fornecer um vértice do CPU para o GPU. Deste modo, o nosso algoritmo corre, naturalmente, nos processadores de fragmentos.

Um dos grandes problemas quando se pretende programar um algoritmo no GPU reside nas grandes

limitações do paradigma de *stream computing*. O mesmo programa é executado para múltiplos registos de activação (um *stream*). A execução é efectuada em paralelo nos diferentes processadores disponibilizados pela placa sem qualquer comunicação entre eles. Não existem registos ou qualquer outro tipo de memória global partilhada para escrita. No caso dos programas de fragmentos, o output de cada programa é um pixel no framebuffer de saída. Os diferentes processadores vão operando em paralelo gerando pixels distintos. Também não é possível ao programa decidir qual o pixel a escrever no final, pois tal é imposto pela rasterização da primitiva gráfica invocada.

Para além de um conjunto de instruções para operações numéricas em números de vírgula flutuante, existe um outro conjunto de operações dedicadas ao processamento de texturas. Quando pensamos em programação genérica as texturas servem, essencialmente, para guardar as suas estruturas de dados. Ora, a organização dos dados em texturas é, em muitos casos, um dos grandes obstáculos à programação dos GPUs.

O interesse crescente na programação de GPUs para tarefas normalmente atribuídas ao CPU faz desta área um turbilhão de novos desenvolvimentos, quer em termos de hardware, quer em termos de software. As capacidades dos processadores de vértices e de fragmentos está em permanente evolução. A cada geração de novas placas mais e melhor funcionalidade é incorporada nestes processadores. Consequentemente, as APIs que permitem programar os GPUs estão também em constante evolução e não é de estranhar a panóplia de extensões ao OpenGL, que permitem tirar partido do hardware mais recente, ou das constantes actualizações do Direct3D.

A nossa implementação foi realizada sobre OpenGL tendo os programas (*shaders*) sido escritos usando a linguagem Cg da nVIDIA. Algumas das extensões mais importantes do OpenGL usadas na nossa implementação são:

- `EXT_pbuffer` – permite a criação de frame buffers invisíveis para rendering *offscreen*.
- `NV_float_buffer` – permite a criação de frame buffers (*pbuffers*) e texturas formadas por valores em vírgula flutuante sem restrições de limites aos seus valores que não os da própria representação binária.
- `ARB_render_texture` – permite que o conteúdo de um *pbuffer* possa ser usado directamente como uma textura. Oferece assim um mecanismo de realimentação onde os dados escritos em passos anteriores possam ser processados por novos programas de fragmentos dentro do GPU sem haver necessidade de uma viagem de ida e volta até ao CPU.

4.1. Organização dos dados em texturas

No caso do nosso algoritmo, como ele funciona sobre malhas 4-8 regulares e estas têm uma estrutura exterior

formada por quadriláteros, o mapeamento das nossas malhas de vértices para a memória do GPU poderia ser a mais natural possível. A malha seria representada sob a forma de textura 2D armazenada no formato de vírgula flutuante. Cada *texel* representaria um vértice, encontrando-se os vértices vizinhos dos quadriláteros da malha nas posições vizinhas do respectivo *texel*.

No entanto, como se verá adiante, esta primeira disposição dos dados faz-nos incorrer numa grande penalização em termos de eficiência, obrigando-nos a pensar numa estratégia diferente para guardar as malhas 4-8 regulares.

A eficiência dos programas de fragmentos está directamente ligada ao número de instruções que os respectivos processadores têm que executar. Os acessos à memória (texturas) exibem uma latência elevada, mas o facto está escondido graças a um *pipeline* muito extenso. Deste modo, a preocupação fundamental está na redução do número de instruções que os programas terão que executar.

O programa mais simples funcionaria desenhando um *quad* de dimensões $(2w-1) \times (2h-1)$ pixels, para uma malha regular inicial de tamanho $w \times h$ guardada numa textura rectangular 2D. Dentro do programa far-se-ia a identificação do tipo de pixel (vértice) a gerar e utilizar-se-iam as máscaras apropriadas (ver Figura 3 e Figura 4).

O problema desta estratégia, é que o suporte para *branching* nos processadores de fragmentos é inexistente¹. Os saltos dinâmicos são simulados com escritas condicionais. As diversas execuções possíveis do programa são expandidas e todos os ramos são executados. No final, apenas se aproveitam para escrita no output os valores do ramo que seria efectivamente seguido. Se considerarmos que o número de instruções para calcular cada o tipo de vértice da malha refinada usa aproximadamente o mesmo número de instruções, a execução do *shader* seria cerca de 4 vezes mais lenta devido à inexistência de *branching*. O valor 4 obtém-se deixando de fora as regras para as fronteiras (caso contrário seria ainda pior) e considerando os 3 tipos de regras da Figura 3. Note-se que a regra do tipo 3 na realidade é um par de regras. Uma para os vértices gerados ao longo das arestas horizontais da malha e outra igual mas rodada 90º para os vértices gerados ao longo das arestas verticais.

Deste modo, propomos um *layout* das malhas regulares dentro de texturas diferente do acima mencionado. A ideia é usar um shader diferente para cada tipo de vértice a gerar, evitando-se, assim, o *branching*. No caso dos vértices interiores da malha, existem quatro tipos distintos, dando origem a outros tantos programas de fragmentos. Esta estratégia obriga-nos a concentrar no *pbuffer* de saída todos os vértices do mesmo tipo.

Note-se que a escrita no pixel buffer em coordenadas aleatórias é impossível. Cada programa escreve apenas numa localização pré-determinada pelo processo de varrimento da primitiva gráfica.

Considere-se uma malha regular de $w \times h$ vértices. O resultado final de um passo de subdivisão do nosso esquema modificado será disposto numa textura, de dimensão $(2w-1) \times (2h-1)$, exibindo a estrutura de blocos ilustrada na Figura 5.

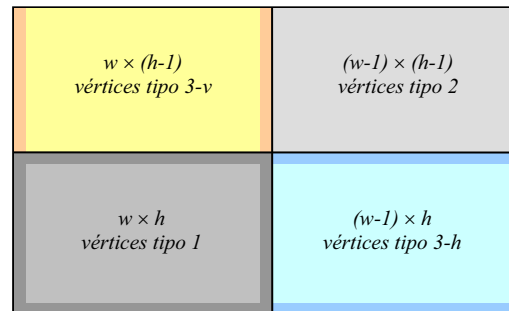


Figura 5 - Layout de uma malha regular sob a forma de textura

Os bordos mais escuros de cada zona correspondem a vértices que se situam na fronteira, pelo que é necessário usar as variantes apropriadas dos respectivos programas principais.

Obviamente, a malha de entrada de dimensão $w \times h$ é, também ela, representada da mesma forma dentro de uma textura caso contrário seria necessário, no final de cada passo, voltar a mudar a representação da malha na memória do GPU.

4.2. Fragment Programs

Não há qualquer dificuldade acrescida pelo facto de as malhas se encontrarem organizadas desta forma na memória. Do ponto de vista dos programas de fragmentos para cada uma das regras, eles serão invocados separadamente, desenhando um quadrilátero que cubra cada um dos blocos acima ilustrados (claro que as fronteiras também terão os seus próprios programas específicos).

Na tentativa de reduzir ainda mais a complexidade dos nossos programas de fragmentos procedemos ainda a uma simplificação das nossas regras. Na secção 3 mostrámos as máscaras associadas a cada regra de subdivisão. Estas máscaras exibem um suporte superior às do esquema de subdivisão de Catmull-Clark. O problema do suporte alargado é que o número de pontos envolvidos no cálculo é superior e isso traduz-se num maior número de acessos a texturas e em mais operações aritméticas. A juntar a isto, as regras obtidas na Secção 3 (Figura 3) introduzem uma complicação adicional ao criarem dois casos especiais para as fronteiras. De facto, a implementação directa das regras conduziria a programas distintos para as duas filas de vértices mais exteriores.

¹ Na realidade, as placas da 5ª geração tais como as GeForce 6000 já possuem alguma capacidade, embora limitada, para efectuar saltos dinâmicos nos *fragment programs*.

Decidimos por isso simplificar as regras do tipo 2 e do tipo 3 que são as que fazem explodir a complexidade do número de casos na fronteira da malha. Se repararmos, os vértices mais exteriores dessas máscaras têm um peso muito pequeno e no nosso esquema modificado resolvemos transferir esse peso para os vértices vizinhos mais interiores, mantendo a simetria das máscaras. O resultado final pode ser observado na Figura 6. Não observámos quaisquer desvantagens da utilização das regras modificadas em termos das propriedades que pretendíamos para as superfícies geradas.

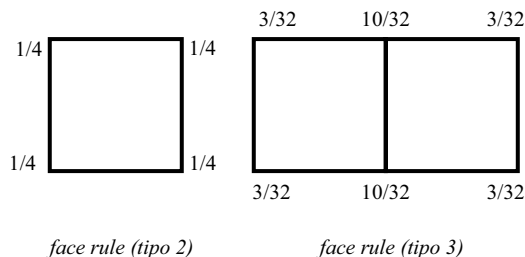


Figura 6 - Regras finais (após simplificação) do nosso esquema de subdivisão 4-8 modificado

A título meramente ilustrativo, dada a simplicidade dos programas obtidos, apresenta-se de seguida o programa que efectua os cálculos dos vértices tipo 2. Os outros casos são semelhantes, apenas envolvendo mais alguns pontos.

```
float4 main(float2 uv: TEXCOORD0,
            uniform samplerRECT mesh,
            uniform float2 dims): COLOR
{
    float4 p1 = texRECT(mesh, mapToQuadType1(uv, dims));
    float4 p2 = texRECT(mesh, mapToQuadType2(uv, dims));
    float4 p3 = texRECT(mesh, mapToQuadType3h(uv, dims));
    float4 p4 = texRECT(mesh, mapToQuadType3v(uv, dims));
    return (p1+p2+p3+p4)/4;
}
```

Nos nossos programas optámos por especificar as coordenadas de textura dos vértices dos quads de tal forma que elas coincidam com as coordenadas inteiras dos pixels (de modo a que a origem seja o canto inferior esquerdo do quad). As funções auxiliares `mapToQuadTypeX` são muito pequenas e simples e, para cada pixel a gerar no pbuffer, indexam um ponto de referência apropriado em cada um dos blocos. O vector 2D *dims* contém as dimensões da malha do passo anterior.

5. DETALHES DO ALGORITMO DO CPU

Nesta secção analisaremos a parte do algoritmo assistida pelo CPU. Na realidade, para além da detecção da condição de terminação e das ordens para gerar os sucessivos *quads*, pouco mais há a fazer do lado do CPU.

Considerando uma malha regular inicial de dimensões $R_w \times R_h$ e, uma resolução final pretendida, não superior a $M_w \times M_h$, o algoritmo começa por criar dois pbuffers com as dimensões máximas $M_w \times M_h$. Na realidade um dos pbuffers pode ser reduzido para um quarto desde que tenhamos o cuidado de colocar a malha inicial no buffer certo de modo a que no final o buffer que contém

o resultado seja o maior. Note-se que a cada passo do algoritmo se dá a troca dos papéis dos dois pbuffers.

```
// Activar os processadores de fragmentos para
// ambos os pbuffers

// As dimensões correntes da malha são as dimensões da
// malha regular de entrada
Fw = Rw; Fh = Rh

// Enquanto o pbuffer for suficientemente grande para
// mais um passo de subdivisão...
while((Fw < Mw/2) && (Fh < Mh/2)) {

    // Cálculo das dimensões da nova malha
    Fw = 2*Fw-1;
    Fh = 2*Fh-1;

    // Tornar o buffer de saída o framebuffer corrente
    ob->Activate();

    // usar o buffer de input como textura
    wglBindTexImageARB(ib->getHPBufferARB(),
                      WGL_FRONT_LEFT_ARB);

    // Chamar a função que desenha os quatro quads
    // usando os programas referidos na secção 4.2
    Subdivide2Steps(Fw, Fh, ib, ob);

    // libertar o buffer de input da textura
    wglReleaseTexImageARB(ib->getHPBufferARB(),
                          WGL_FRONT_LEFT_ARB);

    // Desactivar o buffer de output
    ob->Deactivate();

    // Trocar os papéis dos buffers
    // de input (ib) e de output (ob)
    Swap(ib, ob);

    steps++;
}
```

De notar que no nosso algoritmo não é obrigatório efectuar a subdivisão da malha de uma só vez para o nível de subdivisão final pretendido. Na realidade, pode-se efectuar a subdivisão por blocos, caso o nível de subdivisão pretendido seja tal que o GPU não possa suportar todas as texturas carregadas em simultâneo na sua memória. Mais ainda, devido à sua natureza recursiva, é sempre possível pegar em qualquer malha obtida por subdivisão e voltar a subdividi-la.

6. MALHAS COM LOD DINÂMICO

No caso do nosso simulador de tecidos, o modelo dinâmico utilizado tem por base o proposto por Baraff e Witkin [Baraff98]. É usada a mesma técnica de integração numérica que permite efectuar grandes passos na simulação de cada vez.

O nível de detalhe das nossas malhas varia dinamicamente ao longo da simulação [Birra04] e para tal tiramos partido da utilização de malhas 4-*k* que são uma generalização das malhas 4-8 regulares. Zonas da malha onde são detectados grandes efeitos angulosos são subdivididas pelo CPU, localmente. A ideia é usar um maior número de partículas onde de facto elas parecem ser mais necessárias para capturar as rugas e dobras dos tecidos.

De modo a podermos usar o algoritmo de pós-processamento das malhas 4-8 aqui proposto torna-se necessário subdividir a malha da simulação para que os triângulos pertençam todos ao mesmo nível de detalhe.

Uma solução barata, totalmente oferecida pelo hardware de interpolação do GPU, passa por efectuar o rendering da malha usando as coordenadas 3D dos respectivos vértices como se de coordenadas de textura se tratassem.

Como posição para os vértices usam-se as coordenadas paramétricas da malha.

Apesar de simples, a abordagem revelou ter aplicação prática quando em presença de uma gama relativamente pequena de níveis de detalhe. Se a amplitude de níveis de detalhe for muito grande acabamos por conseguir notar o efeito anguloso da superfície pois o processo de uniformizar o nível de detalhe vai gerar muitos pontos co-planares para cada zona inicialmente plana da malha original. O efeito é que a superfície é suavizada mas a curvatura tem um raio muito pequeno porque os pontos que provocam a aparência angulosa estão muito próximos espacialmente.

É importante realçar que este processo ocorre totalmente dentro do GPU. A malha resultante da uniformização está num pBuffer da placa gráfica, sendo este usado como textura no primeiro passo recursivo do nosso algoritmo de subdivisão.²

7. RESULTADOS

Nas Figuras 7, 8 e 9 podemos observar o resultado prático da aplicação desta técnica. Uma zona angulosa do tecido foi ampliada de modo a que se pudessem observar individualmente os pontos gerados pelo nosso algoritmo de subdivisão. Note-se a ausência de artefactos e a elevada suavidade da malha final. Na Figura 8 podemos observar com grande clareza o efeito suavização da malha original. Pequenas covas ou bicos são tapados ou cortados pela malha subdividida.

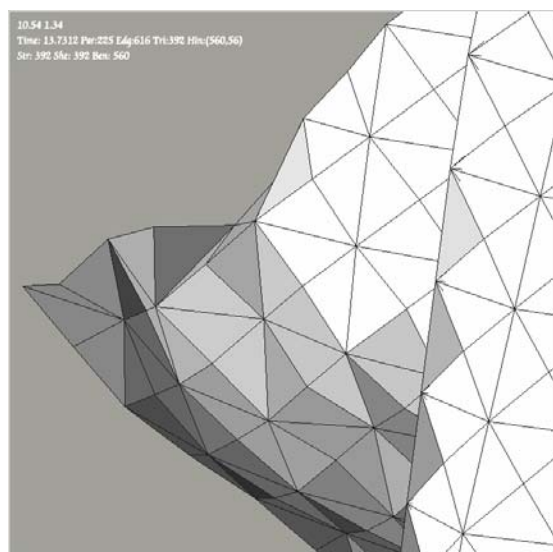


Figura 7 – malha poligonal 4-8 regular usada numa simulação

Em termos de performance medimos o débito de vértices dos nossos *shaders* no último passo de subdivisão que leva à criação de uma malha de 497x497 vértices. O mesmo *shader* foi executado 1000 vezes sem que houvesse mudanças de framebuffer ou outras tarefas pesadas de modo a medir com maior precisão a

² Na realidade é necessário invocar um shader que disponha os dados de acordo com o layout definido na subsecção 4.1.

performance. No caso de se usar um único programa ingénuo para processar todos os 4 tipos de vértices, o tempo obtido para o teste atrás referido foi de 80s. Usando os programas separados para cada tipo de vértice o tempo baixa para pouco mais de 1/4 (27s). Todos os testes foram realizados numa placa GeForce FX Go5200 (a menos rápida da penúltima geração de placas da nVIDIA). Estes valores representam um débito sustentado de aproximadamente 9 milhões de vértices por segundo.

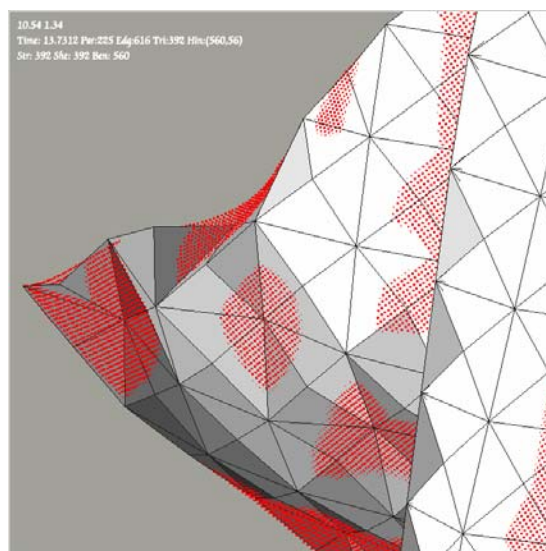


Figura 8 – A mesma malha da figura anterior mas agora com os pontos resultantes da subdivisão

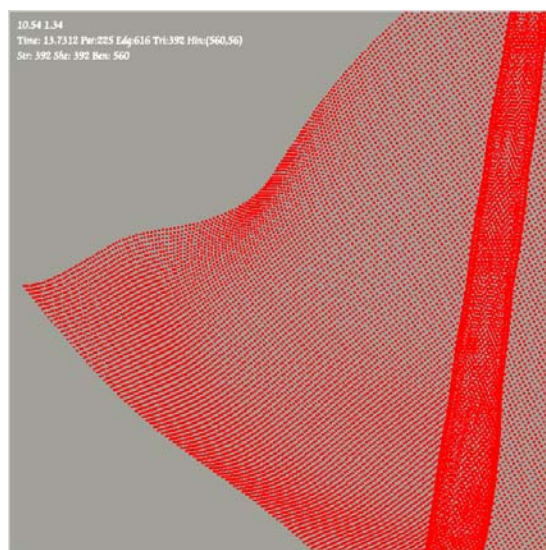


Figura 9 – Apenas os pontos gerados pela subdivisão para a malha da Figura 7

O mesmo algoritmo, com algumas optimizações de eficiência, demora 70s, para efectuar o mesmo teste, num Pentium 4 com Hyperthreading a 3.0Ghz e bus de memória a 800Mhz. Note-se que a placa gráfica usada é muito limitada, quer em termos de unidades de processamento paralelas quer em termos de memória

(tamanho e velocidade de acesso), sendo claramente batida por modelos mais recentes.

Na Figura 10, podemos observar uma malha, não regular, com variação dinâmica do nível de detalhe. Após o processo de uniformização da malha, conforme descrito na secção 6, procedeu-se à sua subdivisão, sendo o resultado ilustrado na Figura 11.

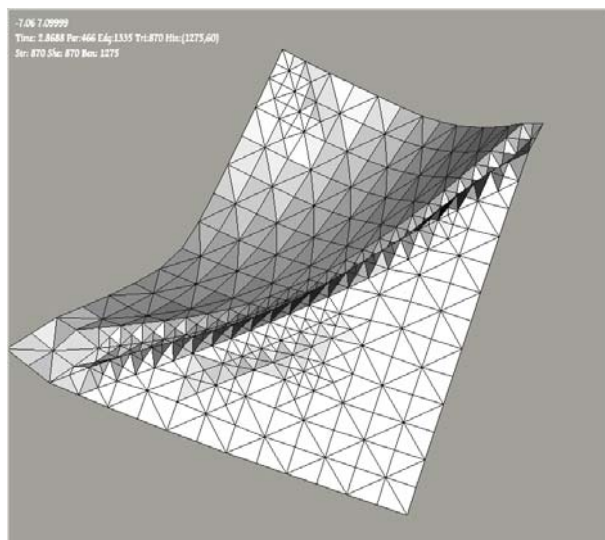


Figura 10 – Malha de simulação com variação dinâmica de LOD

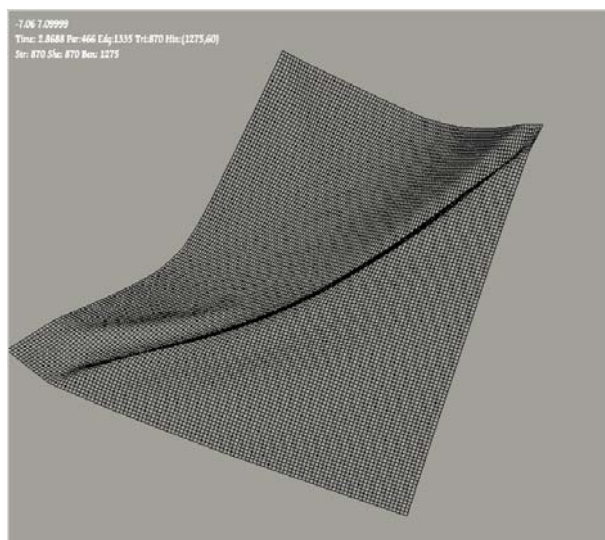


Figura 11 – A malha da figura anterior depois de subdividida

8. CONCLUSÕES E TRABALHO FUTURO

Neste artigo apresentámos um novo algoritmo recursivo de subdivisão de malhas 4-8 regulares. As características principais do algoritmo são resultantes da fusão de dois passos de subdivisão de cada vez. De modo a otimizar a eficiência do método deduzido procedemos à simplificação de algumas das regras de subdivisão sem aparente perda das características fundamentais pretendidas, principalmente a grande suavidade da malha final obtida.

Apesar de o algoritmo ser recursivo, resultando numa perda de desempenho quando comparado com o método proposto em [Bol03], tem a vantagem de não ser necessário carregar previamente o GPU com texturas gigantescas contemplando os diferentes casos de topologia. Além do mais o algoritmo apenas requer uma mudança de contexto do OpenGL (operação pesada aliada à activação dos puffers) entre cada aplicação recursiva do mesmo, podendo cada passo subdividir um grande número de patches. No caso do algoritmo proposto em [Bol03] é necessária uma mudança de contexto por cada patch, sendo esses tempos descontados pelos autores na análise da performance do seu algoritmo. A justificação, válida porém, é que no futuro os drivers otimizarão o processo. Outras extensões do OpenGL, emergentes, poderão tornar o processo de comutação de puffers numa operação mais leve, tais como a novíssima extensão EXT_framebuffer (apenas disponível em drivers beta).

Como trabalho futuro, propomos a investigação do problema da detecção de colisões nos passos de subdivisão sucessiva. Para tirar partido do algoritmo aqui apresentado é necessário que esta operação de validação do estado da malha - detectar que não existem novas colisões introduzidas pelo passo de subdivisão - se processe totalmente dentro do GPU. Existe já algum trabalho nesse sentido embora ainda esteja numa fase muito embrionária. O problema é de fácil solução se ignorarmos as auto-colisões e apenas existir um número reduzido de objectos na cena.

ANEXO A

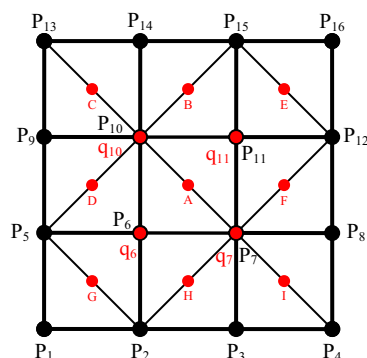


Figura 12 – Primeiro passo de subdivisão

Vértices criados durante o primeiro passo:

$$A = 1/4 (P_6 + P_7 + P_{10} + P_{11})$$

$$B = 1/4 (P_{14} + P_{15} + P_{10} + P_{11})$$

$$C = 1/4 (P_{13} + P_{14} + P_9 + P_{10})$$

$$D = 1/4 (P_9 + P_{10} + P_5 + P_6)$$

$$E = 1/4 (P_{15} + P_{16} + P_{11} + P_{12})$$

$$F = 1/4 (P_{11} + P_{12} + P_7 + P_8)$$

$$G = 1/4 (P_5 + P_6 + P_1 + P_2)$$

$$H = 1/4 (P_6 + P_7 + P_2 + P_3)$$

$$I = 1/4 (P_7 + P_8 + P_3 + P_4)$$

Vértices filtrados durante o primeiro passo:

$$q_{10} = 1/2 P_{10} + 1/8 (P_{14} + P_6 + P_9 + P_{11})$$

$$q_{11} = 1/2 P_{11} + 1/8 (P_{15} + P_7 + P_{10} + P_{12})$$

$$q_6 = 1/2 P_6 + 1/8 (P_{10} + P_2 + P_5 + P_7)$$

$$q_7 = 1/2 P_7 + 1/8 (P_{11} + P_3 + P_6 + P_8)$$

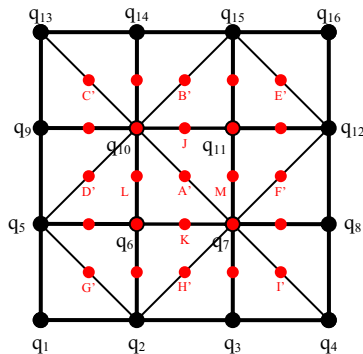


Figura 13 – Segundo passo de subdivisão

Vértices criados durante o segundo passo:

$$J = 1/4 (A + B + q_{10} + q_{11})$$

$$K = 1/4 (A + H + q_6 + q_7)$$

$$L = 1/4 (A + D + q_{10} + q_6)$$

$$M = 1/4 (A + F + q_{11} + q_7)$$

Vértices filtrados durante o segundo passo:

$$r_6 = 1/2 q_6 + 1/8 (D + H + A + G)$$

$$r_7 = 1/2 q_7 + 1/8 (A + I + H + F)$$

$$r_{10} = 1/2 q_{10} + 1/8 (A + B + C + D)$$

$$r_{11} = 1/2 q_{11} + 1/8 (A + B + E + F)$$

$$A' = 1/2 A + 1/8 (q_{10} + q_7 + q_6 + q_{11})$$

Expandindo as expressões para cada tipo de vértice:

$$r_6 = 1/32 (P_1 + 4P_2 + P_3 + 4P_5 + 12P_6 + 4P_7 + P_9)$$

$$K = 1/32 (3P_2 + 3P_3 + P_5 + 9P_6 + 9P_7 + P_8 + 3P_{10} + 3P_{11})$$

$$L = 1/32 (P_2 + 3P_5 + 9P_6 + 3P_7 + 3P_9 + 9P_{10} + 3P_{11} + P_{14})$$

$$A' = 1/64 (P_2 + P_3 + P_5 + 14P_6 + 14P_7 + P_8 + P_9 + 14P_{10} + 14P_{11} + P_{12} + P_{14} + P_{15})$$

REFERÊNCIAS

- [Baraff98] David Baraff and Andrew Witkin. Large Steps in Cloth Simulation. In *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 43-54. ACM SIGGRAPH, 1998.
- [Birra04] Fernando P. Birra, M. Próspero dos Santos. Dynamic Mesh Resolution and Stable Cloth Simulation. In *Advances in Computer Graphics in Portugal, Special Edition of the VIRTUAL Journal*, CD-ROM, November/December, 2004, ISBN 972-98464-5-6, ISSN 0873-1837. Online version: <http://virtual.inesc.pt/aicg04>
- [Bolz02] J.Bolz, and P.Schröder. Rapid Evaluation of Catmull-Clark Subdivision Surfaces. In *Proceedings of Web3D*, 11-17. 2002.
- [Bolz03] J.Bolz, and P.Schröder. Evaluation of Subdivision Surfaces on Programmable Graphics

Hardware, 2003.

<http://multires.caltech.edu/pubs/GPUSubD.pdf>

- [Brickhill01] Pratical Implementation Techniques for Multi-Resolution Subdivision Surfaces. In *Game Developers Conference*.
- [Bridson02] R. Bridson, R. Fedkiw, and J. Anderson. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Transactions on Graphics*, 21(3): 594–603, 2002.
- [Catmull78] E. Catmull, and J. Clark. Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes. *Computer Aided Design* 10(6): 350-355. 1978.
- [Doo78] D. Doo, and M.A.Sabin. Behaviour of Recursive Subdivision Surfaces Near Extraordinary Points. *Computer Aided Design* 10(6):356-360. 1978.
- [Loop87] C.T. Loop. *Smooth Subdivision Surfaces Based on Triangles*. Master's Thesis, Department of Mathematics, University of Utah, August 1987.
- [Loop01] C.Loop. Triangle Mesh Subdivision with Bounded Curvature and the Convex-Hull Property. *Tech.Rep.MSR-TR-2001-24*. Microsoft Research. 2001.
- [Stam98] Jos Stam. Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values. In *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 395-404. ACM SIGGRAPH, 1998.
- [Velho00] Luis Velho and Jonas Gomes. Variable Resolution 4-k Meshes: Concepts and Applications. *Computer Graphics Forum*, 19(4):195-214, 2000.
- [Velho01] Luis Velho and Denis Zorin. 4-8 Subdivision. *Computer Aided Geometric Design*, 18(5): 397-427, 2001. Special Issue on Subdivision Techniques.
- [Zorin02] D.Zorin, and D.Kristjansson. Evaluation of Piecewise Smooth Subdivision Surfaces. *The Visual Computer* 18, 5-6, 299-315. 2002.

Round Tables

Algoritmo para Rasterização *Run-length* de Formas Convexas

Helder M. A. Magalhães
Faculdade de Engenharia da Universidade do Porto
helder.magalhaes@fe.up.pt

RESUMO

No presente documento é apresentada uma abordagem diferente sobre algoritmos para rasterização de círculos e elipses. É avançada uma proposta que introduz modificações em algoritmos existentes, permitindo obter novas primitivas. No sentido de retirar partido de especificidades associadas ao problema, é criado um mecanismo tabelar de rasterização. Finalmente é apresentada uma implementação, utilizando DHTML (*Dynamic Hyper Text Markup Language*), para ilustração de um caso possível de aplicação a sítios *Web*. A implementação permite criar figuras sem recurso a imagens, porventura melhorando a eficácia da comunicação cliente-servidor e apresentando-se como uma alternativa viável a normas emergentes ainda em fase de aceitação.

PALAVRAS-CHAVE

Rasterização *run-length*, primitivas derivadas da circunferência, super-elipse, mecanismo de rasterização tabelar, DHTML.

1. INTRODUÇÃO

A problemática associada à rasterização de formas curvilíneas em malhas de pontos é uma questão que tem recebido a atenção de dezenas de investigadores nas últimas décadas. O trabalho de investigação realizado neste campo conduziu a diversas melhorias nos algoritmos existentes [Yao95] [Yao98].

Embora tenham surgido avanços, muitos dos algoritmos introduzidos assumiram compromissos ou pressupostos para se destacarem face a algoritmos pré-existent. Na maioria dos casos, foi dado mais ênfase ao binómio exactidão-desempenho, uma vez que melhorias no desempenho introduziram, nalguns casos, reduções na exactidão [Molla01]. Esta questão teve mais impacto na rasterização da elipse, uma vez que as necessidades de decomposição do algoritmo e de consideração de casos especiais levaram a falhas em muitos dos algoritmos anteriores [Golipour03].

Neste documento apresenta-se uma proposta diferente, com modificações introduzidas em algoritmos existentes [Foley94][Yao95] para possibilitar a derivação de novas primitivas e a obtenção de melhorias no

desempenho. Deste modo, são apresentados aspectos genuínos e consequentemente completadas as abordagens existentes.

Adicionalmente, é apresentado um mecanismo de rasterização tabelar que permite tirar partido não só da proposta algorítmica aqui apresentada mas também de um grande leque de potenciais aplicações.

É apresentada uma implementação, ligada ao desenho de páginas *Web*, que tenta simultaneamente combinar os melhores aspectos analisados e manter a abstracção necessária para ser utilizada como exemplo de aplicação a vários domínios. A motivação para direccionar a implementação neste sentido surgiu ligada à inexistência de um mecanismo para a declaração de formas nas normas HTML/CSS [Zorn05], uma questão por resolver neste âmbito. Para contornar o problema, tipicamente são utilizadas imagens, trazendo esta abordagem alguns inconvenientes. Em primeiro lugar, o uso de imagens é considerado frequentemente um aspecto negativo quer em termos de desempenho quer de percepção de qualidade por parte do utilizador. Em segundo lugar, no decorrer do carregamento das várias imagens que compõem a página, é necessário um

elevado número de contactos ao servidor *Web*.

Espera-se que a proposta apresentada venha a solucionar algumas das questões levantadas, especialmente ao nível da delimitação de tabelas e da inclusão de formas em páginas. A solução passa pela geração *client-side* das formas, minimizando substancialmente o problema.

1.1 Descrição do Problema

O problema em abordagem surge no decurso de duas questões. Por um lado, a ausência de um mecanismo adequado para rasterização de formas convexas, derivadas da elipse. Por outro lado, a elevada especificidade com que grande parte dos autores focaram repetidamente a utilização de dispositivos gráficos de baixo nível de abstracção.

As duas questões criaram espaço para novas abordagens, baseadas em rasterização de nível mais elevado. Neste documento apresenta-se uma abordagem em torno de ambos os aspectos, tentando estabelecer a ponte entre eles.

1.2 Estrutura do Documento

Este artigo inicia-se pela análise do estado da arte. É abordado trabalho relacionado ao nível conceptual, algorítmico e de implementação.

De seguida, é apresentado um mecanismo adequado para a rasterização de formas convexas, baseado na modificação de algoritmos existentes.

Finalmente, são apresentados dois estudos de caso, realizada a comparação com as mais importantes alternativas existentes e retiradas ilacções sobre o trabalho apresentado.

2. TRABALHO RELACIONADO

É dado especial destaque a três itens, pela sua aproximação e relação com o trabalho aqui descrito. Em primeiro lugar, ao nível conceptual, é dada uma visão geral sobre a norma CSS3 no âmbito deste artigo. De seguida, ao nível algorítmico, é feita uma análise sobre a rasterização *run-length*. Finalmente, ao nível de implementação, é analisada a biblioteca JSGraphics, uma proposta tecnologicamente próxima da apresentada.

2.1 Folhas de Estilo, Nível 3

A norma CSS (*Cascading Style Sheets*), promovida pelo W3C (*World Wide Web Consortium*), foi introduzida para possibilitar a separação de forma e conteúdo em páginas *Web*. A informação relativa a formatação visual seria declarada na folha de estilo (CSS), ao passo que o conteúdo seria mantido no ficheiro

HTML. Neste momento a norma está ainda em discussão, encontrando-se na fase de “rascunho elaborado” (*working draft*). Embora as normas 1 e 2 não tenham chegado totalmente à fase de produção (implementação completa por parte dos criadores de *software*), o consórcio decidiu avançar para a definição do terceiro nível (CSS3).

No contexto deste artigo, a norma poderá ser futuramente utilizada para obtenção de formas circulares e elípticas recorrendo à propriedade *border-radius*, definida no corrente *working draft* [Boland05]. A propriedade é definida declarando até 4 pares de medidas, correspondentes à dimensão dos semieixos de até quatro quartos de elipse distintos, utilizados no contorno da tabela. No exemplo da figura 1 é declarado apenas um par, pelo que as coordenadas são utilizadas para todos os quartos de elipse utilizados no contorno.

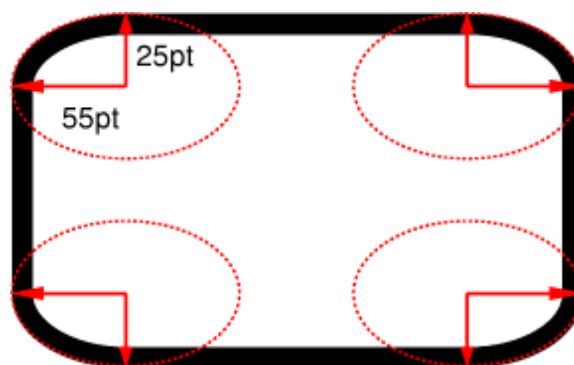


Figura 1: Tabela com contorno elíptico graças à propriedade *border-radius* (Fonte: [Boland05])

Caso os valores a utilizar sejam previamente determinados, é possível obter uma elipse convenientemente orientada. Actualmente, apenas é possível declarar o raio de quatro quartos de círculo, no navegador *Web Mozilla* [Belesis05]. Porém, é possível obter uma tabela baseada numa elipse orientada a 45 graus, caso os raios declarados para quartos opostos sejam iguais. Isto é exemplificado na figura 2, embora neste caso um dos raios declarados não cumpra o requisito.



Figura 2: Tabela com diferentes valores declarados para o raio dos quartos de círculo que compõem a borda (Fonte: [Belesis05])

Outra propriedade interessante desta norma emergente é a possibilidade de utilizar imagens no contorno de tabelas. Esta propriedade denomina-se *border-image* e a sua correcta aplicação fornece um meio alternativo de utilizar formas no contorno tabelar. Nesse sentido, é necessário utilizar uma imagem formatada convenientemente, como a da figura 3, onde é demonstrado o uso de uma elipse orientada para o efeito.

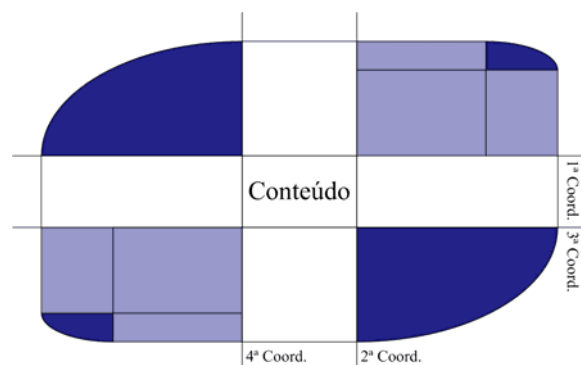


Figura 3: Exemplo de utilização de uma elipse orientada no contorno tabelar

Esta figura foi obtida tomando uma elipse orientada arbitrariamente (tonalidade escura) dividida em segmentos com derivada zero e infinita. Aos segmentos menores foi adicionado espaço intermédio (tonalidade clara) para manter a coerência visual. A área a branco é automaticamente ajustada de acordo com a dimensão do conteúdo. Para obter o efeito descrito, a imagem é acompanhada da declaração de 4 coordenadas utilizadas na sua decomposição.

2.2 Rasterização Run-length

O mecanismo de rasterização *run-length* é uma estratégia que consiste em substituir o desenho ponto-a-ponto por sequências de pontos com a mesma informação de cor, formando segmentos horizontais ou verticais.

Por um lado, esta abordagem é vantajosa tendo em conta a configuração do *hardware* gráfico disponível actualmente. O ganho de desempenho obtido é significativo, devido à optimização da utilização dos barramentos de entrada/saída: desenhar ponto-a-ponto entre dois pontos-limite é geralmente menos eficiente do que desenhar uma linha entre esses dois pontos.

Por outro lado, a modelação de um problema de rasterização com base nesta estratégia poderá melhorar a compreensão ao nível conceptual, facilitando a introdução de melhorias ao nível algorítmico, como foi o caso de [Yao95].

Em termos históricos, o mecanismo surgiu inicialmente ligado à rasterização da recta (uma referência

recente é, por exemplo, [Stephenson00]). Um exemplo de utilização está patente na figura 4.

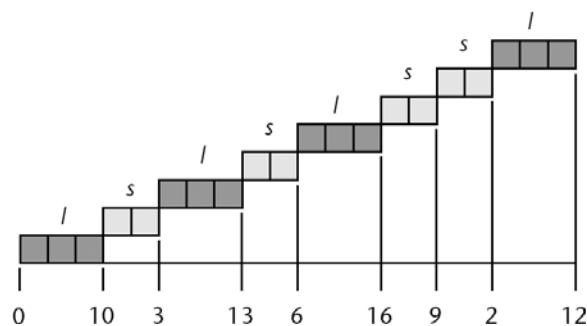


Figura 4: Rasterização da linha $y = \frac{17}{41}x$ evidenciando a existência de várias sequências de pontos alinhadas (Fonte: [Stephenson00])

As letras apresentadas na figura referem-se ao comprimento das sequências de pontos: “l” significa uma sequência longa, ao passo que “s” significa uma sequência curta. Na base da imagem apresenta-se a codificação *run-length* da linha.

O mecanismo foi aplicado por Yao e Rokne à rasterização da circunferência [Yao95] e, mais tarde, generalizado para a elipse [Yao98]. No caso específico da rasterização de curvas, o ganho de desempenho aumenta se for combinado com a abordagem tradicional ponto-a-ponto em certas circunstâncias, nomeadamente quando a rasterização se aproxima de zonas de declive unitário em módulo. Nestes casos, o comprimento das sequências de pontos é frequentemente igual a 1 e o mecanismo *run-length* poderá inclusivamente trazer perdas (não significativas, no entanto). A combinação das duas técnicas levou a que fosse praticamente duplicado [Yao95] o desempenho algorítmico obtido, tomando como elemento de comparação o algoritmo de Bresenham (ou o seu equivalente *midpoint circle* [Foley94]). As melhorias de desempenho foram medidas sem contar com a poupança em operações de entrada/saída, que previsivelmente traz ganhos adicionais acentuados.

2.3 JavaScript VectorGraphics

Esta biblioteca, doravante referida como JSGraphics, foi criada com o intuito de permitir o uso de algumas primitivas gráficas em páginas Web [Zorn05], através do uso de DHTML. Uma aplicação possível da biblioteca é a utilização como alternativa ao SVG (*Scalable Vector Graphics*), tecnologia actualmente em fase de aceitação [Zorn05].

Ao nível da rasterização, a biblioteca baseia-se em elementos rectangulares flutuantes (DIV), dispostos segundo um sistema de coordenadas absolutas. Por um lado, uma vantagem decorrente desta abordagem é

uma maior liberdade no posicionamento de cada um dos elementos. Por outro lado, esta é também uma desvantagem uma vez que não tira partido da relatividade entre os elementos (coerência espacial).

Em termos tecnológicos, a utilização de DHTML vai também ao encontro do trabalho apresentado. Porém, as capacidades desta biblioteca, em termos de primitivas suportadas, a par da sua filosofia de posicionamento absoluto, não permitem a utilização nas circunstâncias previstas na presente implementação, nomeadamente o contorno de tabelas.

A figura 5 exemplifica a obtenção de algumas formas recorrendo a esta biblioteca.



Figura 5: Exemplo de formas geradas pela biblioteca JSGraphics

Na figura, destaca-se a rasterização do círculo, tomada como elemento de comparação ao nível tecnológico com a abordagem apresentada neste artigo.

3. O ALGORITMO *ROUND TABLES*

A implementação levada a cabo neste trabalho pressupõe que sequências de pontos em escada (*pixel stair-step*) apenas se tocam nos cantos. Esta formulação, denominada *8 way stepping* (devido à possibilidade de um *pixel* se “deslocar” em 8 direcções), está patente no algoritmo de *midpoint* utilizado [Foley94] e uma abordagem similar é utilizada nos trabalhos de [Yao95] e [Zorn05].

3.1 A Tabela como Mecanismo de Rasterização

A tabela pode ser utilizada como um mecanismo de rasterização de alto nível. A sua utilização traz uma nova abordagem à problemática da rasterização e um novo conjunto de aplicações possíveis.

Algumas das virtudes deste mecanismo são as seguintes:

- sistema de coordenadas relativas;
- unidades atómicas adaptativas;
- optimização da utilização da memória;
- possibilidade de rasterização de qualquer tipo de figura;
- largamente suportado em dispositivos gráficos.

Cada uma das vantagens é seguidamente detalhada.

Em primeiro lugar, uma das principais vantagens do mecanismo é a relatividade das coordenadas: estas não são absolutas mas relativas às células anteriormente declaradas e estas, por sua vez, ao início da tabela. Esta propriedade é particularmente interessante na exploração de coerência espacial, existente na maioria das primitivas.

Este mecanismo apresenta um potencial acrescido pelo facto do seu sistema de coordenadas permitir a adaptação do tamanho da célula. Esta pode ser utilizada como unidade atómica (como se fosse um *pixel*) ou pode ser distorcida propositadamente para declarar sequências de pontos (*run-length*) de forma eficiente. Isto corresponde a um impulso importante na exploração de simetrias. Embora o artigo apresentado se debruce sobre condições excepcionais decorrentes da rasterização de circunferências, este mecanismo pode ser aplicado com ganhos de eficiência a grande parte dos algoritmos que contenham sequências de pontos.

A optimização da utilização de memória poderá ser obtida de duas formas: explorando a propriedade de expansão de linhas/colunas ou utilizando algoritmos iterativos. Na implementação proposta, ambas as estratégias são postas em prática. Por um lado, tira-se partido do facto de que uma célula pode ser configurada para ocupar o espaço destinado a um conjunto de células, previa ou posteriormente declaradas. A expansão pode dar-se sobre as linhas (expansão vertical) ou sobre as colunas (expansão horizontal). Por outro lado, o algoritmo utilizado na rasterização é um algoritmo iterativo, que possibilita a declaração de linhas/colunas à medida que são determinadas, não sendo necessário mantê-las em memória.

A possibilidade de rasterizar qualquer tipo de figura decorre das vantagens já enunciadas. À partida, este facto não apresenta vantagens sobre abordagens “tradicionais” (como o recurso a imagens). Porém, esta abordagem poderá ser vantajosa caso a imagem contenha grandes regiões com a mesma intensidade de cor. O mecanismo pode ser modelado para se assemelhar a uma codificação RLE (*Run-Length Encoding*), uma forma simples e eficaz de compressão de dados.

Finalmente, uma das grandes apostas deste mecanismo decorre do facto de ser largamente suportado em dispositivos gráficos, uma vez que a tabela é um dos formatos mais utilizados para exibição de conteúdos.

3.1.1 Nível de Abstracção

A abordagem tabelar envolve, à partida, alguma perda de desempenho devida ao seu alto nível de abstracção. A figura 6 pretende clarificar esta afirmação.

Como é evidenciado na figura, a tabela em páginas *Web* está ligada a um patamar de abstracção elevado,

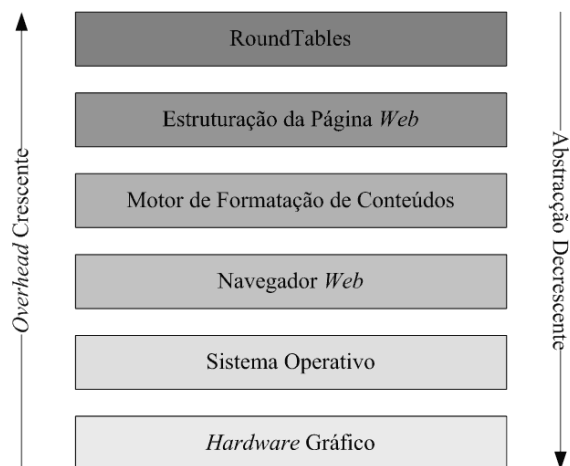


Figura 6: Camadas de abstracção e consequente overhead

trazendo este facto algumas virtudes, já referidas, e alguns inconvenientes. A principal desvantagem numa abordagem a este nível é a perda de desempenho face a abordagens nativas, ao nível da aplicação, sistema operativo ou *hardware*. A figura 6 evidencia que esta perda tem origem no *overhead* criado pelo atraso das camadas subjacentes.

3.1.2 Exemplo de Aplicação

A figura 7 revela a rasterização de um círculo de raio 17, dividido por quadrantes, utilizando o mecanismo tabelar descrito.

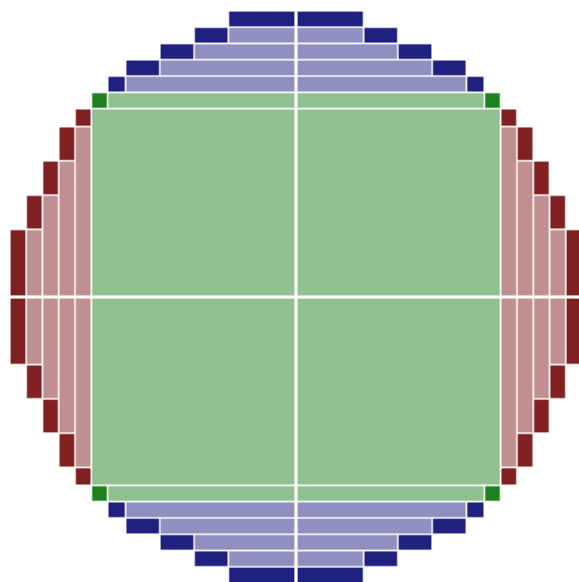


Figura 7: Rasterização de um círculo com o algoritmo proposto

A estrutura tabelar é evidenciada na figura através da ampliação da dimensão das células, adição de espa-

çamento entre as células e utilização de tonalidades diferenciadas: tonalidades escuras nas células pertencentes à circunferência e tonalidades claras no preenchimento.

O facto de ter sido utilizado o valor 17 para o raio salienta um caso especial de rasterização de circunferências, em que existe um ponto comum a ambos os octantes. Este ponto, denominado ponto-fronteira, é adicionado ao espaço central de ajuste.

3.2 DHTML como Ferramenta para Rasterização de Alto Nível

DHTML, acrónimo de *Dynamic Hyper Text Markup Language*, é uma combinação de três tecnologias: *JavaScript*, *CSS (Cascading Style Sheets)* e *HTML (Hyper Text Markup Language)*, utilizadas em conjunto para conferir carácter dinâmico a páginas Web.

Na implementação proposta, cada uma das tecnologias tem uma função específica:

- *JavaScript*: utilizado para gerar a informação representativa da tabela;
- *HTML*: alberga a informação gráfica;
- *CSS*: utilizado para a formatação visual (dimensão e cor) da informação gerada.

A separação de forma e conteúdo obtida é praticamente total, como seria desejável. Porém, alguma informação de forma é embebida no código HTML gerado, uma vez que neste é declarado o comprimento das sequências de pontos a rasterizar. A razão para não fazer a total separação entre forma e conteúdo prende-se com razões práticas e históricas da realidade dos navegadores Web. O suporte dinâmico de *CSS* ainda não é total e o suporte nativo de grande parte dos navegadores era muito limitado ou inexistente até há pouco tempo atrás. Uma vez que estatísticas revelam um período de transição razoável [W3Schools05], a utilização de uma abordagem baseada em normas actualmente pouco suportadas seria menos útil a curto/médio prazo.

Adicionalmente, a versão de teste do algoritmo utiliza valores para a dimensão das células que são múltiplos dos valores utilizados pelo algoritmo. Os valores arbitrados deverão ser coerentes com os especificados na folha de estilo (*CSS*), uma vez que é esta que mantém o controlo da espessura da linha a rasterizar. Com esta facilidade é assim possível obter *pixels* superiores à unidade, facilitando a leitura e depuração da tabela, como no exemplo da figura 7.

3.3 O Algoritmo de Rasterização

O algoritmo de rasterização criado corresponde a uma adaptação do algoritmo *midpoint* para a geração de circunferências, devido à sua elevada eficiência e fácil

compreensão [Foley94]. A sua implementação na linguagem JavaScript é apresentada de seguida:

```
//inicialização de variáveis
//(notar a utilização do factor de transformação T)
var x=0, y=R, p=(1-R)*T, iE=3, iSE=(5-2*R)*T;

while(x < y) {
  if(p < 0) {
    //seguir para este
    //(na direcção horizontal)
    p+=iE;
    iSE+=2;
    dx++;
  }
  else {
    //seguir para sudoeste
    //(direcções horizontal e vertical)
    drawRun(dx);
    dx=1;
    p+=iSE;
    iSE+=4;
    y--;
  }
  x++;
  iE+=2;
}

//para considerar excepções e lidar com transformações
//(por exemplo, raio 17 e/ou factor de transformação 20)
if(x == y) {
  if((dx % 2) == 0) {
    dx--;
    drawRun(dx);
    dx=1;
  }
  else {
    if(2 < dx) {
      dx-=2;
      drawRun(dx);
      dx=1;
    }
    drawCenter();
  }
}
```

Uma das modificações em relação ao algoritmo inicial relaciona-se com a utilização da tabela como mecanismo de rasterização, tirando partido da abordagem *run-length*. Como se pode comprovar no código apresentado, não se efectua o desenho ponto-a-ponto, memorizando-se um segmento horizontal até que ocorra uma mudança de linha (quando o algoritmo determina que seguirá para sudoeste), momento em que é traçado o segmento.

A função `drawRun(X)` faz o desenho de uma linha de comprimento X, tirando partido das simetrias no círculo. Uma vez que esta simetria existe ao nível dos octantes, cada chamada à função `drawRun(X)` leva a cabo a criação de 8 linhas, 4 na direcção horizontal e 4 na direcção vertical (evidenciadas na figura 7). A função `drawCenter()` destina-se a considerar os casos especiais em que surge um ponto-fronteira entre octantes com declive de módulo unitário (secção 3.1). Neste sentido, a função desenha esse ponto na porção central de espaço adicionado aos octantes.

Introduziu-se também a noção de factor de transformação, T, com o objectivo de operar alterações nos parâmetros de decisão. Estas alterações acarretam consequências imediatas sobre o comportamento do algo-

ritmo, sendo obtidas formas que variam com o factor de transformação utilizado.

A variação dos valores de transformação e a sua relação com as formas obtidas pode observar-se na tabela 1.

Forma	Nome	Transformação	Empregue
a	losango	$T \simeq 0$	0.00001
b	losango arredondado	$0 < T < 1$	0.5
c	círculo	$T = 1$	1
d	quadrado insuflado	$T > 1$	2
e	“quadrado”	$T \gg 1$	100

Tabela 1: Valores genéricos para o factor de transformação e valores empregues na figura 8

Outra modificação a salientar foi criada pela necessidade de considerar casos especiais e lidar com as alterações introduzidas no comportamento do algoritmo. Neste sentido foi criado código adicional com a função de detectar linhas não desenhadas no decorrer do ciclo de rasterização. Esta situação poderá ocorrer quando existirem passos finais do algoritmo que apenas sigam na horizontal (não sendo considerados pela função `drawRun(X)`) e quando o factor de transformação causar saltos horizontais significativos.

4. DISCUSSÃO DE RESULTADOS

Nesta secção são demonstrados os resultados obtidos. Em primeiro lugar são analisadas as formas obtidas, bem como outras formas obtidas por permutação de quartos de forma. São focadas questões de eficiência no código gerado e efectuado um pequeno estudo de desempenho.

4.1 Novas Primitivas Obtidas

A escolha de diferentes factores de transformação (secção 3.3) no algoritmo desenvolvido levou à criação de duas novas primitivas: o “losango arredondado” e o “quadrado insuflado”, presentes na figura 8.

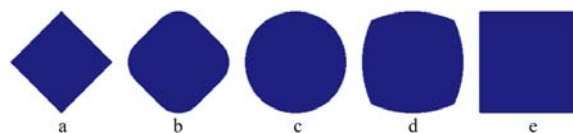


Figura 8: Formas (convexas) geradas pelo algoritmo: (a) losango, (b) losango arredondado, (c) círculo, (d) quadrado insuflado e (e) “quadrado”

Numa primeira análise, o losango arredondado pode ser obtido através da criação de um quadrado arredondado nas bordas com uma rotação a 45 graus. O quadrado insuflado também pode ser aproximado pela conjugação de quatro arcos.

No entanto, as primitivas gráficas apresentadas na figura 8 foram obtidas directamente por aplicação do algoritmo proposto, com grande vantagem ao nível do desempenho. Os factores de transformação utilizados estão patentes na tabela 1, sendo que formas intermédias surgem pela interpolação do factor de transformação, estando apenas limitadas pelo aspecto visual pretendido.

É de notar que o factor de transformação não afecta o ponto de passagem em ângulos múltiplos de $\frac{\pi}{2}$, que se mantém igual ao raio. Isto permite a combinação de diferentes quartos de forma entre quadrantes, aumentando o grau de liberdade de forma exponencial.

Alguns artifícios visuais são ainda passíveis de aplicação neste contexto, possibilitando, por exemplo, a obtenção de formas côncavas, evidenciadas na figura 9.

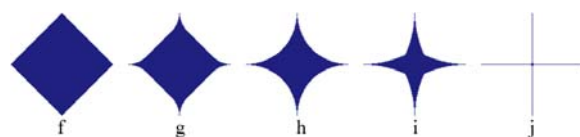


Figura 9: Formas côncavas obtidas por permutação: (f) losango “invertido”, (g) almofada, (h) losango contraído, (i) estrela e (j) soma

As formas foram obtidas através da permutação de quartos de forma opostos, dois a dois, alargando o horizonte de possibilidades. De facto, o losango “invertido” é semelhante ao losango original, uma vez que esta forma apresenta simetria segundo eixos orientados a 45 e 135 graus. De notar que os casos (g), (h) e (i) são conceptualmente diferentes entre si, uma vez que o caso (g) contém segmentos de recta, (h) é obtido com arcos de círculo e (i) é obtido com arcos de círculo que formam, entre cada quarto de forma, um vértice entre si.

4.2 Código Gerado

O código HTML gerado para o primeiro quadrante de um círculo de raio 17 é de seguida apresentado e clarificado. A estrutura tabular obtida é semelhante à da figura 7.

```
<table border="0" cellpadding="0" cellspacing="0">
  <!--INÍCIO DO SEGUNDO OCTANTE-->
  <tr>
    <td class="h_b" style="width:4px"></td>
  </tr>
  <tr>
    <td class="h_f"></td>
    <td class="h_b" style="width:2px"></td>
  </tr>
  <tr>
    <td class="h_f" colspan="2"></td>
    <td class="h_b" style="width:2px"></td>
  </tr>
  <tr>
    <td class="h_f" colspan="3"></td>
  </tr>
```

```
    <td class="h_b" style="width:2px"></td>
  </tr>
  <tr>
    <td class="h_f" colspan="4"></td>
    <td class="h_b" style="width:1px"></td>
  </tr>
  <!--FIM DO SEGUNDO OCTANTE-->

  <!--INÍCIO DO ESPAÇO CENTRAL-->
  <tr>
    <td class="c_f" colspan="5"></td>
    <td class="c_b"></td>
  </tr>
  <tr>
    <td class="c_f" colspan="6" rowspan="5"></td>
  <!--FIM DO ESPAÇO CENTRAL-->

  <!--INÍCIO DO PRIMEIRO OCTANTE-->
  <td class="v_b" style="height:1px"></td>
</tr>
  <tr>
    <td class="v_f" rowspan="4"></td>
    <td class="v_b" style="height:2px"></td>
  </tr>
  <tr>
    <td class="v_f" rowspan="3"></td>
    <td class="v_b" style="height:2px"></td>
  </tr>
  <tr>
    <td class="v_f" rowspan="2"></td>
    <td class="v_b" style="height:2px"></td>
  </tr>
  <tr>
    <td class="v_f"></td>
    <td class="v_b" style="height:4px"></td>
  </tr>
  <!--FIM DO PRIMEIRO OCTANTE-->
</table>
```

A utilização do mecanismo *run-length* permite minimizar o número de células através do seu alongamento, simulando a existência de conjuntos de pontos. A análise da figura e do código apresentado, a par da consideração da existência de contorno, permite afirmar que é obtido o resultado óptimo para cada caso, ou seja, o código gerado para a tabela contém o número mínimo de células possível para o quarto de círculo rasterizado.

Adicionalmente, a estrutura tabular faz referência a classes CSS (class="X"), enunciadas de seguida.

```
/*INFORMAÇÃO DE DIMENSÃO*/

/*altura de linhas horizontais*/
TD.h_b, TD.c_b {
  height:1px;
}
/*largura de linhas verticais*/
TD.v_b, TD.c_b {
  width:1px;
}

/*INFORMAÇÃO DE COR*/

/*cor do contorno*/
TD.h_b, TD.c_b, TD.v_b {
  background-color:rgb(192,192,255);
}
/*cor de preenchimento*/
TD.h_f, TD.c_f, TD.v_f {
  background-color:rgb(128,128,255);
}
```

A utilização de CSS tem o objectivo de centralizar a formatação gráfica. No código apresentado, é visível a separação entre informação de dimensão e de cor.

Por um lado, na informação de dimensão, está patente a declaração da altura de linhas horizontais (*h_b*, *horizontal border*) e a largura de linhas verticais (*v_b*, *vertical border*). A dimensão do ponto comum (*c_b*, *center border*), utilizado nos casos especiais de rasterização, é declarada em conjunto com as dimensões horizontal e vertical, sendo que as suas propriedades resultam da união. Estes valores serão normalmente iguais a 1, excepto quando for desejável criar *pixels* superiores à unidade, para efeitos visuais e/ou de demonstração (como no exemplo da figura 7). Os valores utilizados terão de ser coerentes com o comprimento das sequências de *pixels* declaradas no algoritmo.

Por outro lado, na informação de cor são utilizadas apenas duas tonalidades que, tal como é indicado no código, correspondem a diferentes cores utilizadas para o contorno e preechimento.

A utilização de classes CSS permitiu fazer uma optimização ao nível do comprimento do código e da utilização de memória, sendo também um passo importante na separação entre aspecto e conteúdo.

O código gerado não tira partido de algumas propriedades de simetria existentes no círculo, nomeadamente o facto de cada sequência de pontos poder ser declarada com o dobro do comprimento/largura, rasterizando o círculo linha-a-linha (abordagem levada a cabo na implementação de [Zorn05], por exemplo). Tal facto deve-se a um compromisso assumido de funcionalidade, uma vez que permite utilizar os quartos de forma independente. No entanto, a possibilidade de obtenção de formas completas mantém-se, através do agrupamento das partes: declaração dos quartos de forma dentro de uma tabela com quatro células dispostas duas a duas.

Para além disto, a utilização de quartos independentes também não rentabiliza algumas propriedades do mecanismo tabelar: uma vez declarada informação de largura/altura para uma dada célula, a sua célula “simétrica estruturalmente” (na mesma coluna entre células horizontais e vice-versa) necessita apenas de ser declarada. O motor de formatação encarrega-se de esticar a nova célula até à dimensão anteriormente declarada [Boland05]. Este método permite reduzir alguma redundância de informação.

O compromisso eficácia *versus* funcionalidade acabou por seguir no sentido da funcionalidade uma vez que, sendo possível obter formas completas por agrupamento, a rasterização de formas completas pelo mecanismo faria com que não fosse possível utilizá-las na delimitação de conteúdos, uma das desvantagens da implementação proposta por [Zorn05].

A figura 10 demonstra uma das principais vantagens da implementação levada a cabo: a utilização de quartos de círculo para conferir um aspecto arredondado a tabelas.

Breve excerto d'Os Lusíadas, de Luís de Camões:

As armas e os Barões assinalados
Que da Ocidental praia Lusitana
Por mares nunca de antes navegados
Passaram ainda além da Taprobana,
Em perigos e guerras esforçados
Mais do que prometia a força humana,
E entre gente remota edificaram
Novo Reino, que tanto sublimaram;

Uma vez que se trata de uma tabela, pode ser utilizado qualquer tipo de conteúdo como imagens (a título de exemplo):



Figura 10: Exemplo de aplicação a páginas Web

No exemplo, é criada uma tabela com 3×3 células, sendo uma célula destinada ao conteúdo (a célula central) e as restantes destinadas ao contorno. Das células destinadas ao contorno, as que correspondem aos cantos contêm, por sua vez, tabelas correspondentes aos quartos de círculo previamente rasterizados. A criação de uma tabela única contendo o contorno e o conteúdo invalidaria o mecanismo de *backwards-compatibility*, utilizado para permitir o suporte tradicional a imagens caso não se encontre presente o suporte a DHTML.

4.3 Análise de Desempenho

A presente secção pretende comparar a implementação proposta com duas alternativas existentes. Em primeiro lugar é efectuada a comparação com a abordagem utilizando imagens, muito usada na prática, seguida da comparação com a biblioteca JSGraphics, semelhante ao nível tecnológico.

Os resultados foram obtidos num PC, equipado com um processador AMD Athlon à velocidade de relógio de 1,3 GHz e com 512 MB de memória RAM. Em contactos remotos foi utilizada uma ligação por cabo com uma largura de banda de 2 Mbps. Os testes foram realizados recorrendo aos três navegadores Web predominantes (*Internet Explorer*, *Mozilla* e *Opera*) [W3Schools05], sendo que para os testes envolvendo

a variável tempo foi realizada a média pesada dos tempos obtidos. O sistema operativo utilizado foi o *Windows XP*. Embora os resultados temporais possam variar de acordo com o navegador *Web* e máquina utilizados (entre outros factores) a proporção dos valores apresentados deverá manter-se.

4.3.1 Abordagem utilizando imagens

Nos casos de estudo efectuados foram consideradas as variáveis relevantes apresentadas, que se encontram descritas tomando a comparação com a abordagem recorrendo a imagens:

- número de contactos ao servidor: pretende comparar a variação do número de ligações, quando são carregadas imagens, com a geração de formas a nível local;
- transferência total: utilizada para estabelecer a comparação entre o tamanho do código necessário à geração de imagens e o tamanho das imagens que o código se propõe a substituir. Aqui são contabilizados todos os ficheiros auxiliares transferidos (imagens, HTML, JavaScript, CSS);
- tempo (total) de resposta: esta variável tem o objectivo de medir o atraso total decorrido desde o início do carregamento da página até que é completada a sua exibição. O atraso foi medido em situações para as quais o navegador *Web* necessita de carregar as imagens remotas, tratando-se da situação mais desfavorável (ausência na *cache*);
- atraso fixo: comparação do atraso provocado pela geração *client-side* de imagens *versus* a utilização de imagens residentes a nível local e/ou previamente carregadas por meios de *caching*. O valor obtido corresponde ao tempo necessário para o navegador *Web* efectuar totalmente a apresentação da página, uma vez transferidos todos os ficheiros necessários;
- *overhead* de *markup* introduzido: medida do código HTML adicional necessário para permitir a utilização do mecanismo proposto, mantendo *backwards-compatibility* para proceder ao carregamento alternativo de imagens caso não exista suporte a DHTML.

Para a abordagem recorrendo a imagens, o estudo de caso preparado consistiu na montagem de uma página *Web* com o conteúdo das figuras 8, 9 e 10, acompanhadas de texto descritivo para não limitar o conteúdo a formas rasterizadas.

Relativamente às variáveis consideradas, os resultados numéricos obtidos são enunciados na tabela 2.

Com base nos resultados obtidos, é demonstrado que na implementação proposta o número de contactos é

Variável	Imagens	RoundTables
contactos ao servidor (n)	51	5
transferência total (KB)	48	40
tempo de resposta (ms)	4642	2786
atraso fixo (ms)	2201	1259
<i>overhead</i> de HTML (KB)	–	12

Tabela 2: Resultados numéricos da comparação com a abordagem recorrendo a imagens

reduzido substancialmente. Isto permite reduzir atrasos (detalhados adiante) e aliviar a carga nos servidores. A reutilização do mecanismo na obtenção de várias formas, uma vez transferidos os ficheiros correspondentes, tem aqui um dos seus principais trunfos.

O tamanho total de transferência da implementação proposta, face à abordagem recorrendo a imagens, aparenta uma relativa aproximação devida ao *overhead* introduzido que ofusca a diminuição de imagens conseguida.

Os dados obtidos para o tempo de resposta são francamente positivos, uma vez que esta vantagem tem origem nos dois itens anteriores. Por um lado, a diminuição do número de contactos (n) reduz a latência no estabelecimento ligações necessárias. O valor do decréscimo, estimado em $\Delta t_{ligacoes}$, é obtido com a relação expressa na equação 1.

$$\Delta t_{ligacoes} = (n_{imagens} - n_{roundtables}) \times t_{latencia} \quad (1)$$

Este é o principal facto para a redução de tempo uma vez que a latência é, por norma, significativa tendo em conta a deslocalização e globalização dos servidores *web*. Por outro lado, a diminuição de dados transferidos (d), não tão significativa, efectua uma redução dependente da largura de banda do canal (L). A estimativa do valor $\Delta t_{transferencia}$ é evidenciada na equação 2.

$$\Delta t_{transferencia} = \frac{d_{imagens} - d_{roundtables}}{L_{canal}} \quad (2)$$

Uma vez que o tempo de resposta é a variável com mais percepção por parte do utilizador, é frequentemente utilizado na medição da qualidade de serviço (QoS, *Quality of Service*) de páginas *Web*. Neste sentido, é tomada como uma das variáveis mais relevantes analisadas.

O atraso fixo é também favorável à implementação apresentada, uma vez que embora os navegadores possuam código especializado para lidar com imagens (elementos muito comuns em páginas *Web*), a poupança no carregamento de um elevado número de imagens compensa o atraso provocado pela geração de imagens a nível local.

Em relação ao *overhead* introduzido pela utilização da implementação proposta, os dados obtidos revelam que, proporcionalmente ao tamanho da página transferida, se trata de uma perda significativa. Será um dos aspectos a considerar em termos de trabalho futuro.

Finalmente, a possibilidade de utilização de mecanismos de *caching* poderia permitir evitar o recarregamento de imagens dentro de algumas restrições. Porém, é normalmente necessário estabelecer uma ligação com o servidor para averiguar se a imagem ainda se encontra actualizada. Isto levanta de novo a questão do elevado número de contactos ao servidor (e latência associada). Esta variável não é considerada na avaliação, uma vez que diversos navegadores e sistemas operativos utilizam diferentes estratégias (de *caching*).

4.3.2 Comparação com a biblioteca JSGraphics

A biblioteca JSGraphics [Zorn05] é o principal elemento de comparação ao nível tecnológico. Porém, em termos de aplicação afasta-se do trabalho apresentado, como já foi apontado na secção 2.3.

Foi preparado um teste no qual a biblioteca e o mecanismo apresentado rasterizaram apenas um círculo, sendo variado o raio e medido o atraso correspondente a cada situação.

Utilizando um raio de 64 *pixels* foram obtidos os resultados numéricos apresentados na tabela 3.

Variável	JSGraphics	RoundTables
contactos ao servidor (n)	2	3
transferência total (KB)	20	6
tempo de resposta (ms)	84	248
atraso fixo (ms)	53	232
overhead de HTML (KB)	1	2

Tabela 3: Resultados numéricos da comparação com a biblioteca JSGraphics

Pela análise dos dados, o número de contactos ao servidor pode considerar-se equivalente.

A transferência total utilizando a biblioteca JSGraphics é maior, como seria de esperar, uma vez que esta permite a rasterização de um conjunto alargado de formas para além do círculo (linhas, polígonos, ...). O seu autor refere que o tamanho do código foi desprezado em prol de melhorias no desempenho.

Em termos de atraso, o desempenho da biblioteca é significativamente superior face ao trabalho apresentado. A justificação para este facto está presente na secção 4.2, onde é referido que a implementação tomou a opção de aumentar o grau de flexibilidade, prevenindo perdas no desempenho. De salientar que a biblioteca JSGraphics não permite contorno de cor diferente no círculo gerado, apresentando um número significativamente mais baixo de elementos a rasterizar.

O tempo de resposta total é ligeiramente menos desfavorável ao trabalho apresentado, uma vez que a transferência de dados adicional requerida pela biblioteca JSGraphics adiciona algum atraso na fase inicial, reduzindo ligeiramente a disparidade no desempenho.

O *overhead* de código HTML é semelhante, uma vez que as etapas necessárias, para ser obtida a funcionalidade pretendida, são semelhantes em ambas as abordagens.

A recolha de dados levada a cabo encontra-se representada graficamente na figura 11. Foi utilizada a escala logarítmica para facilitar a leitura, uma vez que existem discrepâncias acentuadas nos valores.

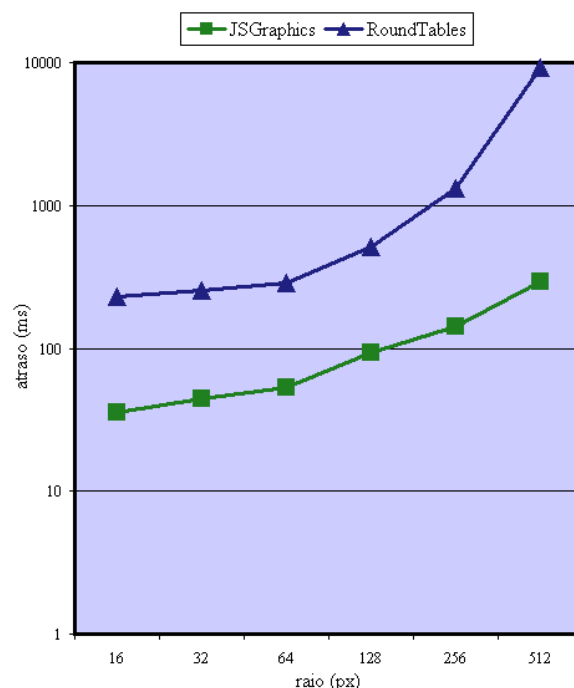


Figura 11: Gráfico comparativo do tempo de resposta entre a biblioteca JSGraphics e a implementação proposta

A figura evidencia uma elevada diferenciação para raios superiores a 128 *pixels*. Uma análise detalhada dos dados recolhidos permitiu verificar que teve origem, principalmente, nos maus resultados obtidos com o navegador *Opera*, que desequilibraram a média de modo significativo. A partir dos 256 *pixels*, o navegador *Mozilla* também começou a revelar atrasos significativos. A disparidade face à biblioteca JSGraphics, a par da análise já efectuada em 4.2 sobre a optimização da estrutura tabelar, permite avançar que na origem dos maus resultados estarão motores de formatação de conteúdo menos eficientes, no que diz respeito a tabelas. Este facto é comprovado pelo desempenho superior do navegador *Internet Explorer*, com valores obtidos mais próximos da biblioteca JSGraphics.

Em relação à comparação com a biblioteca JS-Graphics, resta salientar que apenas foi possível comparar a rasterização de círculos, o que não permitiu tirar partido das capacidades adicionais da implementação apresentada. Por outro lado, a respeito da compatibilidade retroactiva, a implementação proposta permite a degradação (carregamento de imagens) caso alguma das tecnologias necessárias não se encontre disponível, sendo superior neste aspecto.

5. CONCLUSÕES

No presente documento foi apresentada uma nova abordagem para a rasterização de formas, utilizando um mecanismo de coordenadas relativas, a tabela. A nova abordagem permitiu tirar partido de especificidades ligadas ao problema da rasterização, nomeadamente a exploração de coerência espacial. De salientar que o mecanismo tabular de rasterização apresentado não se esgota na implementação proposta (desenho de páginas *Web*), podendo ser utilizado em aplicações onde não exista nenhum mecanismo de rasterização disponível, ou onde as primitivas de desenho sejam muito limitadas. Aplicações futuras poderão incluir, a título de exemplo, folhas de cálculo e processadores de texto.

Na proposta apresentada, foram levadas a cabo alterações em algoritmos existentes, permitindo derivar um conjunto de novas formas convexas, baseadas no círculo. A utilização do factor de transformação permitiu aumentar significativamente o grau de flexibilidade, uma vez que pode assumir valores fraccionários. A utilização de artifícios visuais baseados em simetrias e permutações de quadrantes permitiu igualmente aumentar o leque de formas obtidas.

Ao nível da implementação, foi efectuada uma análise de desempenho que revelou grandes potencialidades. A abordagem tradicional (com recurso a imagens) foi largamente ultrapassada, sendo obtidas reduções notórias ao nível da carga nos servidores e do tempo de resposta. As melhorias tiveram origem na diminuição da quantidade de dados transferidos e na diminuição significativa do número de contactos efectuados. O caso de estudo ao nível tecnológico (comparação com a biblioteca JSGraphics) apontou um desempenho com melhorias em aberto.

Relativamente a alternativas adicionais à implementação apresentada, existem algumas para além das descritas com funcionalidade semelhante. As principais razões pelas quais não foram analisadas relacionam-se com a sua fraca presença no terreno, com questões proprietárias e com a falta de suporte retroactivo. Pelo contrário, as tecnologias utilizadas na implementação proposta (desenho de páginas *Web*) são livres e encontram-se em praticamente todos os navegadores

actualmente disponíveis.

Resta acrescentar que a implementação presente neste artigo não se destina de forma alguma a substituir normas emergentes e que estas poderão mesmo permitir, no futuro, contornar algumas das questões levantadas. O objectivo da proposta apresentada foi o de fazer a aplicação prática dos conceitos descritos, demonstrando um caso possível de utilização para o presente, com tecnologias consideradas omnipresentes.

6. TRABALHO FUTURO

Encontra-se em fase de estudo a aplicação do algoritmo e mecanismo desenvolvidos à elipse. A elipse alinhada nos eixos ([Aken85], [Hearn97] e [Golipour03] para o caso da rasterização ponto-a-ponto e [Yao98] para a rasterização *run-length*) foi já abordada e neste momento está em estudo a sua generalização (elipse orientada arbitrariamente [Fellner91]), com base em trabalho desenvolvido por [Bond02], estando em fase de implementação.

O caso da aplicação dos conceitos à super-elipse, generalização da elipse proposta por Lamé e estudada por [Weisstein99], está também em fase de análise. Esta última abordagem não existe em nenhuma das normas ou implementações apontadas, sendo pois inovadora.

O algoritmo *run-length* proposto por Yao e Rocke [Yao95] para a rasterização do círculo é similar, do ponto de vista da rasterização *run-length*, ao algoritmo descrito neste artigo. Está em curso um estudo de comparação com o algoritmo apresentado, quer ao nível funcional quer ao nível do desempenho. No estudo desenvolvido até ao momento, a abordagem alternativa não considera os casos especiais de rasterização (raio 17, por exemplo) e não foi implementado ainda no algoritmo a utilização do factor de transformação. Finalmente, não foram explorados alguns dos benefícios da rasterização tabular, nomeadamente a exploração de simetrias, por motivos de compatibilidade retroactiva. No futuro, maior suporte tecnológico por parte dos navegadores a par da respectiva actualização por parte dos utilizadores poderão abrir perspectivas de melhorar a implementação levada a cabo. A optimização adicional do código gerado permitirá obter melhorias significativas no comprimento do código gerado e consequentemente no desempenho geral do mecanismo apresentado.

7. AGRADECIMENTOS

Gostaria de expressar o meu profundo agradecimento ao Prof. A. Augusto de Sousa pela disponibilidade revelada no esclarecimento de dúvidas e pelo elevado valor dos comentários efectuados ao conteúdo apre-

sentado. Gostaria também de agradecer ao Prof. José Armando Rodrigues de Almeida pelas elucidações e debate teórico em torno da passagem conceptual da elipse para a super-elipse. Finalmente, gostaria de agradecer a Benjamin Mitchell pela disponibilização da sua implementação do algoritmo de rasterização de elipses generalizadas, criado por [Bond02], e ao Departamento de Engenharia Electrotécnica e de Computadores da FEUP pelo patrocínio na participação no encontro.

REFERÊNCIAS

- [Aken85] Aken, J. Van e Novak, Mark, *Curve-Drawing Algorithms for Raster Displays* ACM Transactions on Graphics, Volume 4, Número 2, Páginas 147-169, Abril de 1985.
- [Belesis05] Belesis, Peter, *Fun with Mozilla Border Radii*, Fevereiro de 2005
<http://www.webreference.com/dhtml/column70/>
(acedido a 4 de Maio de 2005)
- [Boland05] Boland, Tim e Bos, Bert, *CSS3 Backgrounds and Borders Module*, Fevereiro de 2005
<http://www.w3.org/TR/2005/WD-css3-background-20050216/>
(acedido a 4 de Maio de 2005)
- [Bond02] Bond, C., *A New Algorithm for Scan Conversion of a General Ellipse*, Janeiro de 2002
<http://www.crbond.com/papers/ellipse.pdf>
(acedido a 1 de Maio de 2005)
- [Fellner91] Fellner, Dieter W. e Helmberg, Christoph, *Fast Rendering of General Ellipses* Eurographics '91, Elsevier Science Publishers B.V., 1991.
- [Foley94] Foley, James D. e outros, *Introduction to Computer Graphics* Addison-Wesley Publishing Company, Inc., 1994.
- [Golipour03] Golipour, M. e Banissi, E., *Performance, and accuracy Improvements in Algorithms for centered, Axes-aligned Ellipses* Proceedings of the 2003 International Conference on Geometric Modeling and Graphics (GMAC'03), IEEE, 2003.
- [Hearn97] Hearn, Donald e Baker, M. Pauline, *Computer Graphics – C Version*, Second Edition Prentice Hall, Inc., 1997.
- [Molla01] Mollá, Ramón e Vivó, Roberto, *Fixed-Point Ellipse Drawing Algorithm* Polytechnical University of Valencia, 2001.
- [Stephenson00] Stephenson, Peter e Litow, Bruce, *Why Step When You Can Run?* IEEE Computer Graphics and Applications, Novembro/Dezembro de 2000.
- [W3Schools05] W3Schools, *Browser Statistics* Refsnes Data, Julho de 2005
http://www.w3schools.com/browsers/browsers_stats.asp
(acedido a 17 de Julho de 2005)
- [Weisstein99] Weisstein, Eric W., *Superellipse*, 1999 MathWorld – A Wolfram Web Resource.
<http://mathworld.wolfram.com/Superellipse.html>
(acedido a 6 de Maio de 2005)
- [Yao95] Yao, Chengfu e Rokne, Jon, *Hybrid Scan-Conversion of Circles* IEEE Transactions on Visualization and Computer Graphics, Volume 1, Número 4, Dezembro de 1995
- [Yao98] Yao, Chengfu e Rokne, Jon, *Run-Length Slice Algorithms For The Scan-Conversion Of Ellipses* Comput. & Graphics, Vol. 22, No. 4, pp. 463-477, 1998, Elsevier Science Ltd, 1998
- [Zorn05] Zorn, Walter, *High Performance JavaScript Vector Graphics Library*, Março de 2005
http://www.walterzorn.com/jsgraphics/jsgraphics_e.htm
(acedido a 4 de Maio de 2005)

Calculating the Stereo Pairs of a Mirror-Based Augmented Reality System

Filipe Marreiros^a

^aCentro de Computação Gráfica

R. Teixeira de Pascoais, 596, 4800-073 Guimarães
{filipe.marreiros, aderito.marcos}@ccg.pt

Adérito Marcos^{a,b}

^bDSI, Universidade do Minho

Campus de Azurém, 4800-058 Guimarães
marcos@dsi.uminho.pt

Abstract

This paper describes a solution for the calculation of the correct stereo pairs of a Mirror-Based Augmented Reality System (MBARS).

To achieve augmentation half-silvered mirrors are applied due to their dual reflection characteristic that allow simultaneous seeing-through of real objects and the viewing of reflected virtual ones. This way it is possible to merge virtual content reflected by the mirror with real objects that the user can see through it. The virtual content can be either generated by a monitor or a projector. In this paper we will only refer to the monitor-based systems. These systems are view dependent, meaning that depending upon the viewer's position the reflected images have to be recomputed. This is why a Head-Tracking System is needed to supply the current viewer's position. Also we need to consider the main system components and their location, to correctly compute the stereo pairs.

Furthermore we present how the result of the mathematical calculations can be used in combination with commands of the OpenGL API to achieve the desired transformations and to allow interactivity.

Keywords

Augmented Reality, Mirror-Based Display Systems, Stereo Pairs.

1. INTRODUCTION

The Virtual and Augmented Reality technology has been a subject of intensive research and study in the last years. Some results have already been deployed and used in end-user environments.

MBARS, in particular, are used to merge real scenes that are viewed through the mirror, together with the computer generated images, that are projected on it. The virtual content can be either generated by a monitor or a projector. In this paper we will only refer to the monitor-based systems. Using a head-tracking system, the correct images can be computed according to the observer's point of view, this way, allowing the viewer to freely interact with the system.

The systems that we are considering are stereoscopic, so it is necessary to compute the virtual images for the stereo (viewing) pair. The reason for the images to be rendered in stereo is because depth information is required in the final application scenario. When the viewer sees through the mirror he/she will need to perceive the object in its correct location, at a predefined distance from him/her.

This paper is structured as follows: Section 2 introduces the related work, while our conceptual approach is de-

scribed in section 3. In section 4 the transformations performed to the several basic system elements are demonstrated, section 5 presents the OpenGL transformations performed to achieve the correct stereo pairs and section 6 presents our application scenario. Finally section 7 includes a summary and the conclusions of the work.

2. RELATED WORK

The *de-facto's* major work developed for the subject of MBARS is related with the well-known Virtual Showcase device. This work constitutes a seminal contribution to the field, it has been carried out by a international team of researchers leaded by Oliver Bimber whose results have been used for some years now to support further development and research on MBARS (see [Bimber01], [Bimber02a], [Bimber02b], [Bimber03], [Beckhaus03]).

Furthermore the European project intituled Virtual Showcases [VirtualShowcases], also produced a large amount of papers based on the same concepts. We have been working, integrated in this project, to develop our own Showcase – the Augmented Room. For further detail on this device see [Matos03], [Pereira03], [Pereira04], [Marreiros05]. The work of our partners can also be found in [Ledermann03], [Turkish] and in the Virtual Showcases web site [VirtualShowcases].

Regarding the Mirror-Based Systems it is worth mentioning here the work of [Mulder05], which addresses the problems of interaction with the virtual objects in this type of systems. In particular to perform realistic occlusion effects.

In addition to MBARS, there is also an important related work targeting specially the calculation of Stereo Pairs. In this subject we have to refer the brilliant work of Paul Bourke [14] and [15] that approaches some concrete solutions to correctly compute stereo pairs, using OpenGL and GLUT.

Our approach integrates the some of the contributions above-mentioned however adapted for a stereoscopic application scenario where the viewer's point of view requires to be constantly computed to allow a smooth combination of real and virtual artefacts at the eyes of the end-user. Our solution can be defined as a specialization for the common MBARS.

3. OUR APPROACH

The approach used to compute the stereo pairs is based on the following observation: if we have a monitor in a known position over a mirror, then this mirror will reflect an image of the monitor according to the mirror's reflection equations. We call this projected image the "Virtual Monitor" as depicted in Figure 1, because it really does not exist, despite the viewer perceives it as being real. Since the mirror is semi-transparent, computer generated images of the "Virtual Monitor" can be combined with the real objects creating an overall *augmentation* effect.

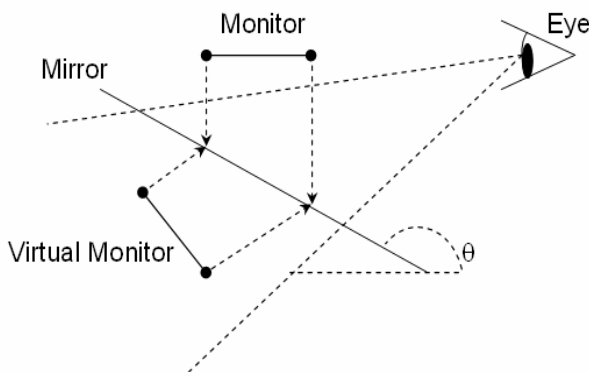


Figure 1: Mirror reflection of the Monitor

The lights play here an important role. If the scene is too much illuminated the viewer observes the virtual object as transparent. If the scene is too dark he/she will not be able to correctly view the real objects. This requires a mid-term solution where the scene is neither too much illuminated nor dark. Due to the fact that semi-transparent mirrors are being used to combine real together with virtual objects, one should be careful where to place the virtual objects. In fact, if a virtual object is placed over a real one that is too much illuminated or has a bright colour, the user will still see the real object behind the virtual one. To overcome this problem, it has

been developed a so-called "Occlusion shadows" [Bimber03], where the lights are explicitly controlled. In the spots where virtual objects are placed, the area (shadow) is simply not illuminate. However, this solution requires an extra projector only to control the lights. Since we deal here with monitor-based this approach does not satisfactorily fulfils our needs. This means that extra attention is necessary when placing the virtual objects.

To allow the viewer to perceive the virtual objects in their correct locations, the stereo images displayed in the Virtual Monitor have to be computed. However, the Virtual Monitor does not really exist; it is just an abstraction tool used to aid the understanding of the overall transformation processes. This means that the image displayed in the real monitor has to be transformed for the Virtual one. There are several steps involved in this procedure that will be described in the following sections.

4. TRANSFORMATIONS OF THE SYSTEM ELEMENTS

The system is composed by basic elements, which are: the Monitor, the Mirror and the Viewer. To be able to compute the images, the positions of these elements have to be known.

The monitor is defined by its edges points and the mirror by its plane that can be obtained using points directly measured over its surface.

To compute the virtual monitor position, the mirror reflection matrix R [Bimber01] is used. Equation 1, exemplifies how to compute the position of the virtual monitor.

$$P_{vm} = R \times P_m \quad (1)$$

To obtain the eyes position (viewer), a head tracking system is used which tracks a point in the middle of both eyes. This will introduce some error, especially if the user rotates his head to certain angles. It is also necessary to take into account the position of the tracking system, since the positions of the eyes have to be converted from the tracking system coordinate system, centred at the device, to the environment coordinate system. Also, note that the positions of the monitor and mirror have as reference the environment coordinate system. The origin of the coordinate system should be at a point viewed through the mirror, for example the at a table top where are placed real and virtual objects.

The second step involved in the process is to simulate that we are using a conventional Monitor to view the stereo images. This way we can use all the general formulation for calculating stereo pairs found in [Bourke99a] and [Bourke99b]. But this involves a series of transformations that will be discussed further ahead.

4.1 Tracking System transformations

As mentioned earlier we have to transform the eyes position that is acquired by the tracking system in its own coordinate system to the system's coordinate system. In

Figure 2 are depicted the elements that take part in this process.

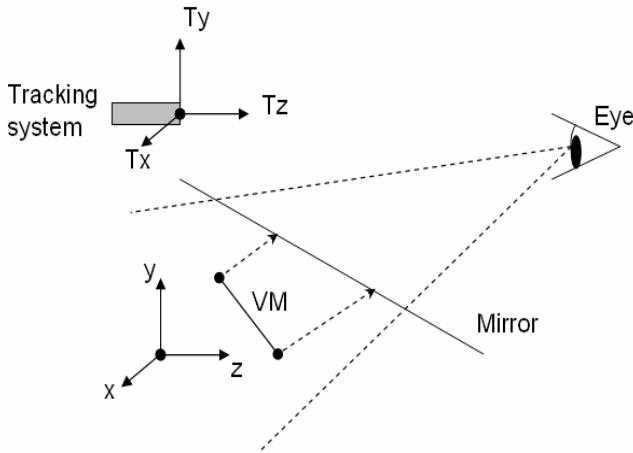


Figure 2: System's and Tracking coordinate systems

To convert a point taken in (Tx, Ty, Tz) to the (x, y, z) coordinate system we first have to translate the point in (Tx, Ty, Tz) to (x, y, z) .

Equation 2, exemplifies how to compute this transformation.

$$P_{(x,y,z)} = P_{(Tx,Ty,Tz)} + T \quad (2)$$

The T matrix is just the origin point of the tracking system, in the main system's coordinate system. Let's consider an example: the origin of the tracking system is at $(0, 10, 0)$ and the eye position taken in the tracking system coordinate system is $(0, 0, 10)$ so the eye position in the system's coordinate system is $(0, 0, 10) + (0, 10, 0) = (0, 10, 10)$.

The next steps would be to rotate the axis to align the tracking coordinate system to the main system's coordinate system. Normally to simplify this task they are already aligned. This is why we don't give further details about this subject. But one can use, if necessary, the formulation presented in further sections, because the concepts are alike.

4.2 Transformation applied to simulate a conventional monitor

As mentioned earlier we need to simulate that we are using a conventional Monitor to view the stereo images. This involves a series of transformations that will be discussed in this section. Figure 3, presents the result of these transformations.

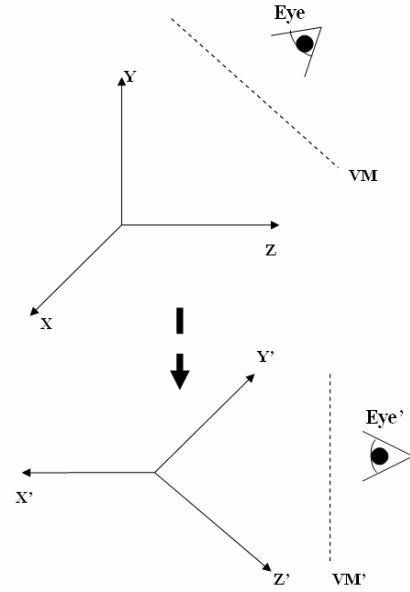


Figure 3: Coordinate System, Virtual Monitor and Eye Transformation

We can observe that several elements of the system are affected by these transformations. We will present now how these transformations are performed.

The first step of this set of operations is to obtain the centre point of the virtual monitor. Considering that we have the four edges points, Pvm_1 , Pvm_2 , Pvm_3 and Pvm_4 , computed using Eq. 1, the centre can be easily obtained using Eq. 3.

$$Vvm_{aux(1,2)} = ((Pvm_{(2,3)x} - Pvm_{1x}) / 2, \\ (Pvm_{(2,3)y} - Pvm_{1y}) / 2, \\ (Pvm_{(2,3)z} - Pvm_{1z}) / 2)$$

$$Pvm_C = Pvm_1 + Vvm_{aux1} + Vvm_{aux2} \quad (3)$$

Then, the translation to the origin is calculated. The translation matrix is given in 4.

$$TM = -Pvm_C \quad (4)$$

The rotations necessary for the virtual monitor plane to be normal to Z axis are obtained performing a set of rotations. The following figures and equations will present step by step these transformations.

The first step is to rotate the virtual monitor that is now centred at the origin, so that the projection of the vector $Vvmx$ on the XZ plane coincides with the X axis. Figure 4 depicts this transformation and the equations are presented in 5.

Y axis rotation:

$$\begin{aligned} \text{If } (Vvmx_z > 0) \quad & \theta = \arccos\left(\frac{Vvmx_x}{D1}\right) \\ \text{If } (Vvmx_z < 0) \quad & \theta = -\arccos\left(\frac{Vvmx_z}{D1}\right) \\ D1 = & \sqrt{(Vvmx_x)^2 + (Vvmx_z)^2} \end{aligned} \quad (5)$$

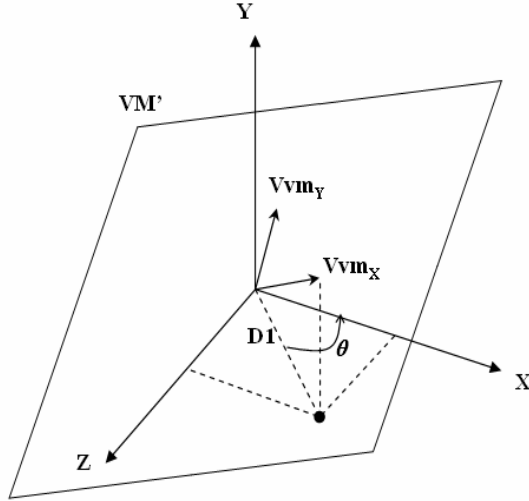


Figure 4: Initial Rotation for the virtual monitor to be normal to the Z axis

The second rotation will then transform vector $V'vmx$ so that it coincides with the X axis, like presented in figure 5 and using Eq. 6.

Z axis rotation:

$$\begin{aligned} \text{If } (V'vmx_y > 0) \quad & \Phi = \arccos\left(\frac{V'vmx_x}{D2}\right) \\ \text{If } (V'vmx_y < 0) \quad & \Phi = -\arccos\left(\frac{V'vmx_z}{D2}\right) \\ D2 = & \sqrt{(V'vmx_x)^2 + (V'vmx_z)^2} \end{aligned} \quad (6)$$

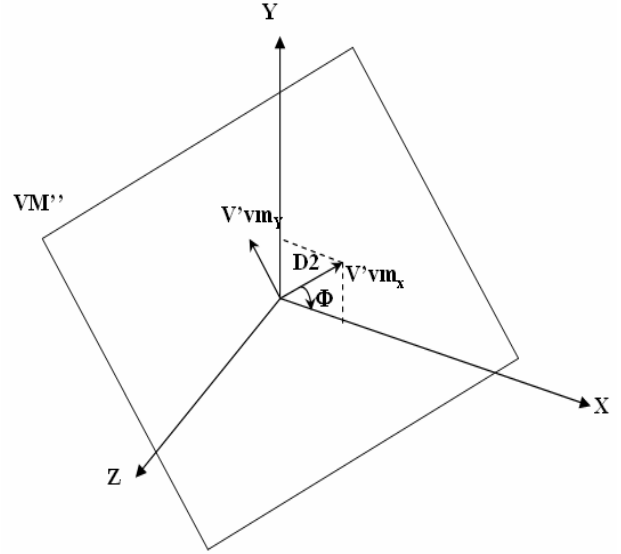


Figure 5: Second Rotation for the virtual monitor to be normal to the Z axis

Finally the last rotation will place $V''vmx$ and $V''vmy$ in the respective axis the X and Y, using Eq. 7. Then we have the transformed virtual monitor normal to the Z axis, like depicted in figure 6.

X axis rotation:

$\text{If } (V''vmy_z \geq 0)$	$\alpha = -\arccos\left(\frac{V''vmy_y}{D3}\right)$	(7)
$\text{If } (V''vmy_z < 0)$	$\alpha = \arccos\left(\frac{V''vmy_y}{D3}\right)$	
$D3 = \sqrt{(V''vmy_z)^2 + (V''vmy_y)^2}$		

One has to take in attention that these transformations are applied not only to the virtual monitor but also to the main system's coordinate system and to the eye. This way we guarantee that the image viewed in the cases presented in figure 3 are always the same.

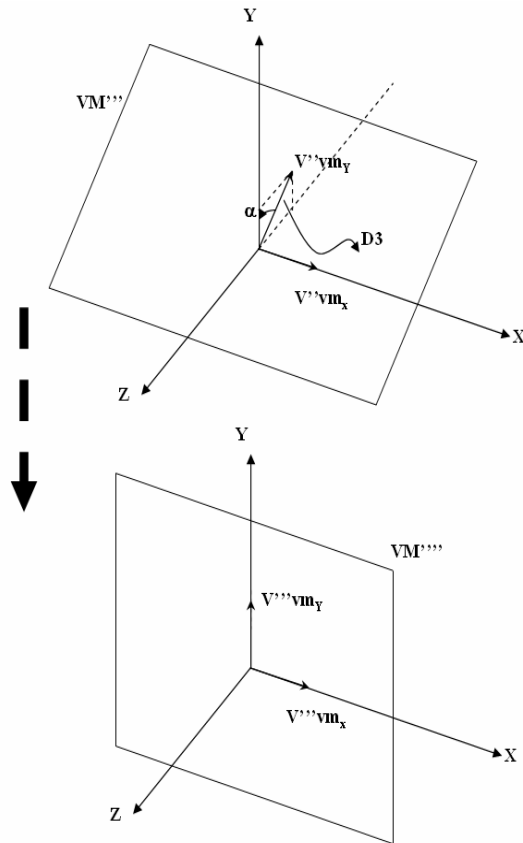


Figure 6: Final Rotation for the virtual monitor to be normal to the Z axis

Having the angles of the rotations and the translation displacement, we can apply them to the several elements. The virtual monitor only needs to be transformed once because its position is a fixed one. The eyes positions need to be computed each time because they can change their position. The operations that must be performed are exemplified on the following pseudo-code:

```
//performed only at the initialization
Pvm[1..4] = Pvm[1..4] - Pvmc;
Pvm[1..4] = RotationY (Pvm[1..4],  $\theta$ );
Pvm[1..4] = RotationZ (Pvm[1..4],  $-\Phi$ );
Pvm[1..4] = RotationX (Pvm[1..4],  $\alpha$ );
```

```
//performed every time
//Left Eye
EyeLeft = EyeLeft - Pvmc;
EyeLeft = RotationY (EyeLeft,  $\theta$ );
EyeLeft = RotationZ (EyeLeft,  $-\Phi$ );
EyeLeft = RotationX (EyeLeft,  $\alpha$ );
```

```
//Right Eye
EyeRight = EyeRight - Pvmc;
EyeRight = RotationY (EyeRight,  $\theta$ );
EyeRight = RotationZ (EyeRight,  $-\Phi$ );
EyeRight = RotationX (EyeRight,  $\alpha$ );
```

Since the coordinate system is also changed the virtual objects will also be affected by these transformations.

```
//performed only at the initialization
//Left Eye
VO = VO - Pvmc;
VO = RotationY (VO,  $\theta$ );
VO = RotationZ (VO,  $-\Phi$ );
VO = RotationX (VO,  $\alpha$ );
```

Because we didn't want to apply these transformations to the object every time, we just transform them at the beginning and use OpenGL commands for updating the positions if they change.

5. OPENGL TRANSFORMATIONS

Now that we have all the necessary information to create the stereo pairs, we just need to apply the OpenGL commands.

The correct way to create stereo pairs is to use an Off-axis projection. It requires a non symmetric camera frustum; fortunately this is supported by OpenGL.

In [Bourke99a] the author demonstrates that this is effectively the correct way to compute the stereo pairs by giving picture examples that prove it. In [Bourke99b] the same author presents how this can be achieved with OpenGL and GLUT. Our work is based on the referred author's work, but applied to our particular case.

5.1 Calculating the OpenGL Model View Transformations

The OpenGL model view matrix, is easily created knowing the transformations presented in section 4. For each eye we will have the following pseudo-code:

```
//Right Eye
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt (EyeRight.x, EyeRight.y,
           EyeRight.z, EyeRight.x,
           EyeRight.y, 0, 0, 1.0, 0);
glRotated( $\theta$ , 0, 1, 0);
glRotated( $-\Phi$ , 0, 0, 1);
glRotated( $\alpha$ , 1, 0, 0);
glTranslated(-Pvmc.x, -Pvmc.y, -Pvmc.z);
```

```
//Left Eye
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt (EyeLeft.x, EyeLeft.y,
           EyeLeft.z, EyeLeft.x,
           EyeLeft.y, 0, 0, 1.0, 0);

glRotated( $\theta$ , 0, 1, 0);
glRotated( $-\Phi$ , 0, 0, 1);
glRotated( $\alpha$ , 1, 0, 0);
glTranslated(-PvmC.x, -PvmC.y, -PvmC.z);
```

One has to be careful with the order of the transformations; these should be supplied as they are.

5.2 Calculating the OpenGL Projection Transformations

The final step of this process is to obtain the OpenGL Projection Matrix. The pseudo-code used in this case is:

```
//Right Eye
glDrawBuffer(GL_BACK_RIGHT);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glFrustum(Pvm1.x-EyeRight.x,
          Pvm2.x-EyeRight.x,
          Pvm3.y-EyeRight.y,
          Pvm1.y-EyeRight.y,
          EyeRight.z, 100000.0);

//Left Eye
glDrawBuffer(GL_BACK_LEFT);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glFrustum(Pvm1.x-EyeLeft.x,
          Pvm2.x-EyeLeft.x,
          Pvm3.y-EyeLeft.y,
          Pvm1.y-EyeLeft.y,
          EyeLeft.z, 100000.0);
```

5.3 The resulting camera Frustum

The last operations will produce the camera frustums for both eyes (the stereo pairs). Figure 7 will be used to better exemplify the result of these operations and to verify that we really have a non symmetric camera frustum.

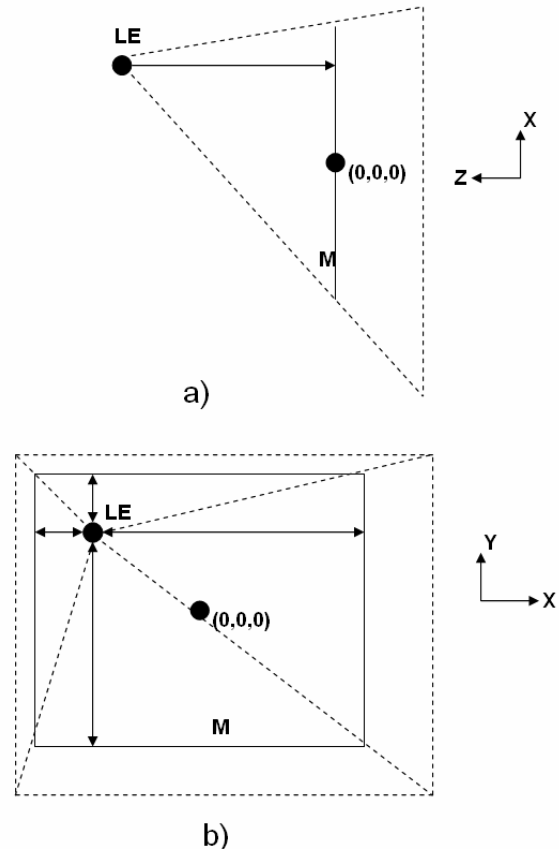


Figure 7: Camera Frustum a) Top View b) Front View

In Figure 7 we only present the camera frustum for the left eye but the same considerations apply to the right eye. The top view clearly shows us that the camera is non symmetric. This is because the `gluLookAt` forces it to be. Let's analyze again the parameters of the `gluLookAt`. The first 2 parameters are the most important they are the Eye and Center. In this case the Eye is (LE_x , LE_y , LE_z) and the Center (LE_x , LE_y , 0). Only the Z changes between them, this forces the camera to point in a straight line towards $Z=0$, this is showed by the arrow in a). In b) we can see the distances between the transformed eye and the virtual monitor; they are used by the `glFrustum`. The `glFrustum` command requires also the near and far plane. We put the far plane at a position very far away. The nearest plane has to be placed at the transformed virtual monitor; its value will be LE_z because this is the distance between the virtual monitor and the eye.

6. APPLICATION SCENARIO

The concepts presented here were used to develop an application scenario for Museum D. Diogo de Sousa in Braga, Portugal. As mentioned earlier we have been working, integrated in the Virtual Showcases project, and developed our own Showcase – the Augmented Room, which stands at the referred museum. For further details see [Matos03], [Pereira03], [Pereira04], [Marreiros05].

In order to give an overview of the system we present images that help to understand the work developed.

The scenario consist of an Ancient Roman Tomb were virtual archaeological artefacts are placed over it. Figure 8 depicts the real component of the scenario, a 1:2 model of the remainders found of the Tomb.



Figure 8: Real component of the scenario

To exemplify how the augmented environment is displayed, we use the montage present in figure 9.



Figure 9: Augmented scenario

It is important to notice that the user will not view the environment exactly like it is presented in figure 9. Notice that the position of the objects is not accurate. We made this illustration due to the fact that; to capture stereo images special hardware is required, which is not available.

The augmented environment is located inside the structure of the Showcase. Figure 10 presents our showcase-the Augmented Room.

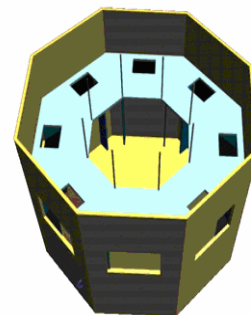


Figure 10: Augmented Room

The user can look through the showcase windows and see the augmented environment. One of these windows is presented in figure 11.



Figure 11: Viewer's window

In figure 11, we can also see the joysticks used for interaction with the virtual content, the shutter glasses and a part of a stereo virtual image used to explain the joysticks usage.

Due to the fact that Museu D. Diogo de Sousa is still closed to the public, further tests like usability tests couldn't be made. But we plan to do so as soon as the showcase is in use by the general public [Marreiros05].

7. SUMMARY AND CONCLUSIONS

We have presented in detail the several steps involved in the calculation of the Stereo Pairs of a Mirror-Based Augmented Reality System.

We started by introducing the specific transformations used in a MBARS, that uses Monitors for image generation. These transformations were used in combination with OpenGL commands to correctly obtain the Stereo Pairs.

The result of this work was tested during our work in the Virtual Showcase project, proving the concepts presented in this paper.

8. ACKNOWLEDGEMENTS

We would like to acknowledge Oliver Bimber and Paul Bourke continuous support during the development of this work.

This work was partially supported by the European Union through the project n. IST-2001-28610.

9. REFERENCES

- [Bimber01] Bimber, O., Fröhlich, B., Schmalstieg, D., and Encarnação, L.M.. The Virtual Showcase, *IEEE Computer Graphics & Applications*, vol. 21, no. 6, pp. 48-55, 2001.
- [Bimber02a] Bimber, O., Fröhlich, B.. Occlusion Shadows: Using Projected Light to Generate Realistic Occlusion Effects for View-Dependent Optical See-Through Displays, *Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR'02)*, 2002.
- [Bimber02b] Bimber, O., Gatesy, S.M., Witmer, L.M., Raskar, R. and Encarnação, L.M.. Merging Fossil Specimens with Computer-Generated Information, *in IEEE Computer*, September, pp. 45-50, 2002.
- [Bimber03] Bimber, O., Encarnação, L.M. and Schmalstieg, D.. The Virtual Showcase as a new Platform for Augmented Reality Digital Storytelling9. *EurographicsWorkshop on Virtual Environments*, The Eurographics Association, 2003.
- [Beckhaus03] Beckhaus, S., Ledermann, F., Bimber, O.. Storytelling and Content presentation with Virtual and Augmented Environments in a Museum Context, *CIDOC 2003*, September 2003, Sankt Petersburg, Russia.
- [Ledermann03] Ledermann, F., Schmalstieg, D.. Presenting an Archaeological Site in the Virtual Showcase, *Proc. of the 4th International Symposium on Virtual Reality, Archaeology, and Intelligent Cultural Heritage (VAST 2003)*, pp. 119-126, Brighton, UK, Nov. 2003. Available as technical report TR-188-2-2003-17, Vienna University of Technology.
- [Matos03] Matos, N., Pereira, P., Grave, L., Marcos, A.. ARK: Augmented Reality Kiosk, *In: Human-Computer Interaction – Theory and Practice (PART II)* Edited by Constantine Stephanidis, Julie Jacko, pp. 168-172. (Vol.2 of the Proceedings of HCI International 2003, 22-27 June, Crete, Greece). Lawrence Erlbaum Associates Publishers Mahwah, New Jersey, London, ISBN: 0-8058-4931-9.
- [Pereira03] Pereira, P., Matos, N., Grave, L., Marcos, A., ARK multi-utilizador, *12º EPCG*, Porto, Portugal, October 8-10 2003, pp. 119-124.
- [Pereira04] Pereira, P., Matos, N., Grave, L., Marcos, A.. Augmented room: a device for visualization of museum artefacts, *In: Proc. of 1st International Conference Virtual Design and Automation*, Poznan, 3-4 Jun. 2004, Poland (published in CD-ROM – ISBN: 972-98872-2-5).
- [Turkish] The Demonstrator Scenario “Turkish Chess-Player”.
<<http://www.ims.tuwien.ac.at/~flo/vs/chessplayer.html>>
- [VirtualShowcases] The Virtual Showcases Consortium
<<http://www.virtualshowcases.com>>
- [Marreiros05] Marreiros, F., Figueiredo, P., Pereira, P., Matos, N., Marcos, A.. An augmented reality showcase to support a cultural heritage scenario, *In: Proc. of Training, Education & Simulation International (TESI) 2005*, Maastricht, 22-24 March 2005, Netherlands (published in CD-ROM).
- [Mulder05] Mulder, J.. Realistic Occlusion Effects in Mirror-Based Co-Located Augmented Reality Systems, *In: VR 2005*, Bonn, 12-16 March 2005, Germany.
- [Bourke99a] Calculating Stereo Pairs
<<http://astronomy.swin.edu.au/~pbourke/stereographics/stereorender/>>
- [Bourke99b] 3D StereoRendering Using OpenGL (and GLUT)
<<http://astronomy.swin.edu.au/~pbourke/opengl/stereogl/>>

Zarabatana, uma interface tangível para ambientes de entretenimento em realidade aumentada

José Miguel Salles Dias
Miguel.Dias@iscte.pt

António Pocinho
ahpocinho@sapo.pt

ADETTI/ISCTE, Associação para o Desenvolvimento das Telecomunicações e Técnicas de Informática, Edifício ISCTE, 1600-082 Lisboa, Portugal, www.adetti.iscte.pt

Resumo

Este artigo descreve a construção de uma interface tangível para ambientes de entretenimento em realidade aumentada e mista, bem como de um jogo demonstrador da nova interface, designado por BlowPipe. O ambiente demonstrado disponibiliza ao utilizador um jogo imersivo, que tem lugar dentro de uma arena, um espaço de interacção em realidade aumentada e mista. Utilizando uma zarabatana física e dardos virtuais, possibilita-se a interacção com um cenário aumentado constituído por alvos em movimento. O objectivo do jogo é abater o maior número possível de alvos num determinado período de tempo. Discute-se a construção da interface tangível Zarabatana bem como a sua integração na Arena, no contexto do jogo BlowPipe.

Palavras-chave

AR Toolkit, MX Toolkit, Realidade Aumentada, Realidade Mista, Interfaces Tangíveis, Arena, Cubo Mágico, Zarabatana.

1. INTRODUÇÃO

O trabalho apresentado neste artigo, tem como objectivo a exploração das potencialidades lúdicas dos ambientes imersivos baseados em realidade aumentada e mista, concretizado com o desenvolvimento de uma nova interface tangível e a construção de um jogo demonstrador da mesma, assente sobre o ambiente de desenvolvimento MxToolkit [Dias03a]. O jogo tem como elemento central, uma interface tangível baseada numa zarabatana física com capacidade de seleccionar e disparar dardos virtuais, com o intuito de atingir alvos, também virtuais, em movimento (Figura 1). O jogo desenrola-se, dentro de uma arena física tridimensional com forma de paralelepípedo [Dias04a] (Figura 2), onde é possível fazer o seguimento em tempo real da posição de um utilizador que se situe no seu interior, através de tecnologia de seguimento baseada em ultra-sons, a qual providencia dois graus de liberdade (XY) em posição (admite-se que a altura do utilizador é constante). O jogador utiliza, para além da referida zarabatana, um dispositivo de visualização *video see-through Head Mounted Display* (H.M.D.), acoplado com um cubo de inércia (Figura 3). Este cubo de inércia adiciona à arena, três graus de liberdade em orientação [isense], perfazendo um total de 5 graus de liberdade, a partir dos quais é possível fazer o seguimento dos movimentos de rotação da cabeça e da translação do corpo do jogador.

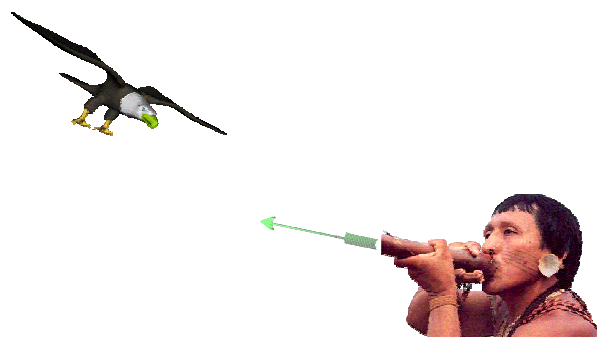


Figura 1 Esboço conceptual do jogo.

Para além do seguimento do utilizador com 5 graus de liberdade, a Arena, inclui um Toolkit orientado a objectos, apropriado para o programador de aplicações em Realidade Aumentada e Mista, o MxToolkit. Este Toolkit cria um nível de abstracção ao programador que providencia a gestão uniforme da entrada e saída de vídeo (para H.M.Ds e ecrãs), distribuição de recursos computacionais numa LAN, som espacial 3D, interacção multimodal em Realidade Aumentada e Mista (interfaces tangíveis, detecção de gestos e compreensão de fala).

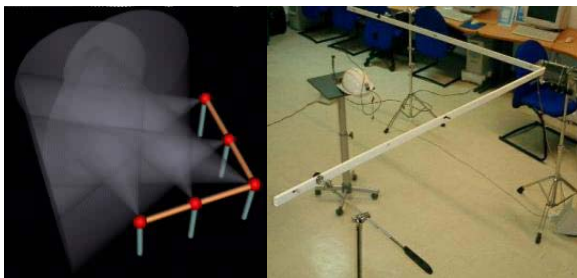


Figura 2 A Arena: Um espaço interior de interacção em Realidade Aumentada e Mista.

O sistema de detecção da pose (posição e orientação) da zarabatana, face à câmara do utilizador, é realizado com o auxílio de uma das faces de um dispositivo de seguimento baseado em marcas fiduciais designado por Cubo Mágico [Dias 03b], que utiliza como sistema de seguimento o AR Toolkit [artoolkit], o qual está integrado na plataforma de desenvolvimento MxToolkit. Associados à Arena e a cada alvo, existem sons posicionados no referencial da Arena (cuja percepção espacial é suportada pelo MxToolkit), que além de proporcionarem um ambiente envolvente para o jogo, facilitam a tarefa de localização dos alvos em movimento. O utilizador tem um tempo limite para atingir o máximo de alvos possíveis e a pontuação obtida, está directamente relacionada com o número de alvos abatidos e o tipo de dardo usado para o efeito. Este trabalho teve como inspirador, o artigo “Jellyfish Party: Blowing Soap Bubbles in Mixed Reality Space” [Yasuhiro03]. Neste artigo é apresentado um sistema de realidade aumentada de carácter lúdico, onde os utilizadores interagem com um ambiente virtual usando um dispositivo físico sensível ao sopro, baseado numa pequena turbina. Pretendeu-se desenvolver uma solução alternativa, de mais baixo custo e com vantagens na sua integração, num ambiente de desenvolvimento de aplicações em realidade aumentada e mista disponível (a Arena).

O artigo está organizado da forma seguinte: na secção 2, está referenciado o estado da arte no tema das tecnologia de suporte à realidade aumentada, especialmente, às tecnologias de seguimento. Na secção 3, apresentamos o diagrama geral do sistema de suporte ao novo interface tangível. A secção 4 trata os assuntos específicos do sistema contendo os pormenores de *hardware* e *software* do protótipo desenvolvido. A secção 5 apresenta os resultados obtidos. Finalmente, a secção 6 contém algumas conclusões e direcções futuras de desenvolvimento.

2. ESTADO DA ARTE EM TECNOLOGIAS DE SUPORTE À REALIDADE AUMENTADA

O paradigma da realidade aumentada é alcançado pela colocação de objectos virtuais 3D, ou sugestões informativas, no mundo real, o que se torna possível fazendo “calibração” ou “seguimento” da câmara virtual, isto é, calculando em tempo real os parâmetros da câmara virtual, cuja posição e orientação coincidem com a do observador na cena real [Azuma 97]. Com esta técnica, objectos “virtuais” podem então ser registados em relação a objectos “reais”, o que significa que estes objectos podem ser vistos na mesma posição e orientação de

outros objectos físicos da cena real. Isto pode ser alcançado utilizando, por exemplo, óculos equipados com pequenos ecrãs de vídeo e com uma ou duas câmaras de vídeo acopladas (*video see-through*). Este dispositivo de visualização pode ser integrado num computador normal, com capacidades gráficas 3D, ou num *wearable computer* (um computador embebido no vestuário), igualmente dotado de capacidades gráficas 3D e com ligação a uma rede de computador sem fios (Figura 3).

2.1 Dispositivos de Visualização

Os dispositivos H.M.D. disponíveis no mercado apresentam já resoluções da ordem de 1280x1024 *pixel*, o que corresponde a uma imagem de muito boa definição. Alguns modelos permitem ainda uma visão estereoscópica



Figura 3 Uma unidade Wearable, com H.M.D, e respectivas baterias embebidas no Vestuário.

(o que exige duas câmaras de vídeo), que pode ser importante para a criação dos efeitos de realismo e imersão pretendidos. Este tipo de dispositivos apresentam ainda uma limitação quanto ao ângulo máximo de visão permitida, que está restringida a cerca de 50°. Quando comparado com o valor máximo de 200° possível com a visão humana, este é um factor muito restritivo na criação de um ambiente imersivo.

2.2 Dispositivos e Técnicas de Seguimento

De um ponto de vista geral, os dispositivos de seguimento permitem, em tempo real, seis graus de liberdade (três em translação e três em rotação). Alguns suportam resoluções inferiores ao milímetro para os graus de liberdade

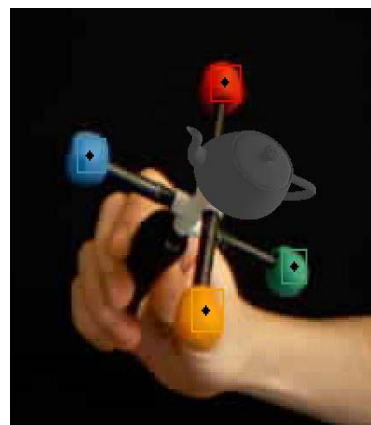


Figura 4 O Sistema de Seguimento ARTIC baseado em Visão e num Artefacto Físico com Marcas Coloridas.

em translação e muito abaixo do grau, para os graus de

liberdade em rotação. As soluções tecnológicas disponíveis são variadas. Com técnicas de visão por computador, a posição e orientação, relativamente à câmara real, de certas marcas pré-colocadas na cena real, são calculadas em tempo real, o que permite a calibração da câmara virtual. É essa a técnica utilizada por Hirokazu Kato no seu sistema AR Toolkit [artoolkit]. Outras técnicas, baseadas ainda em visão, permitem o seguimento da pose de artefactos físicos coloridos e em movimento [Dias04b] (Figura 4). Os sistemas de seguimento por acústica de ultra-sons, determinam a posição de um ponto fixo pela análise do tempo de ida e volta ou da coerência de fase de ondas ultra-sónicas. Obtém-se assim uma superfície esférica sobre a superfície da qual se encontra o alvo. Com vários pares emissor/receptor calcula-se a posição tridimensional do alvo pela intersecção das várias superfícies esféricas. A frequência destas ondas está acima da gama audível por humanos e ronda tipicamente os 20000 Hz. Esta é a técnica utilizada na Arena [Dias 04a] (e neste artigo) para seguir a posição XY do utilizador.

Os sistemas de seguimento electromagnético, usam pares de sensores emissor/receptor para medir alterações em campos magnéticos. Estas variações têm uma repercussão eléctrica cujas propriedades permitem o cálculo da posição e orientação no espaço. Este tipo de sistema requer ambientes controlados, que são normalmente dispendiosos e muito vulneráveis a distorções e interferências de objectos metálicos. É o sistema utilizado por Yasuhiro no trabalho “Jellyfish Party: Blowing Soap Bubbles in Mixed Reality Space” [Yasuhiro 03].

Os sistemas de seguimento inerciais recorrem a dois tipos de componentes nomeadamente, acelerómetros e giroscópios [isense], disponibilizando uma boa resolução nos graus de liberdade, apesar de existirem desvios com o tempo, que necessitam de correcção. A determinação da aceleração permite o cálculo da posição no espaço (integrando a primeira grandeza duas vezes no tempo), tendo como base a 2ª lei de Newton ($f = ma$), o que é feito com uma precisão milimétrica. De forma análoga, os giroscópios permitem calcular a orientação no espaço pelo princípio da conservação do momento angular ($M = I\alpha$), o que é realizado com uma precisão menor que meio grau em todos os graus de liberdade de orientação. Esta tecnologia de seguimento está incluída na Arena e, como tal, foi utilizada neste trabalho, para seguir a orientação do utilizador na Arena. O seguimento pode ser conseguido analisando a radiação electromagnética no espectro dos infravermelhos. Este tipo de sistema localiza objectos reflectores de raios infravermelhos no mundo real. Os emissores irradiam periodicamente num espaço restrito, enquanto câmaras sensíveis a luz infravermelha identificam pontos de reflexão criteriosamente colocados no referido espaço. Com a identificação de vários pontos de reflexão torna-se possível determinar a localização e orientação tridimensional de objectos e/ou utilizadores. Estes sistemas apresentam um custo relativamente elevado, no entanto a precisão é sub-milimétrica e sub-angular num espaço limitado [ARTtrack2]. Existem ainda sistemas híbridos, os quais combinam as vantagens de diver-

sas tecnologias (por exemplo, integrando sensores de ultra-son com métodos inerciais [isense]), de modo a providenciarem os seis graus de liberdade referidos.

3. ARQUITECTURA DO SISTEMA

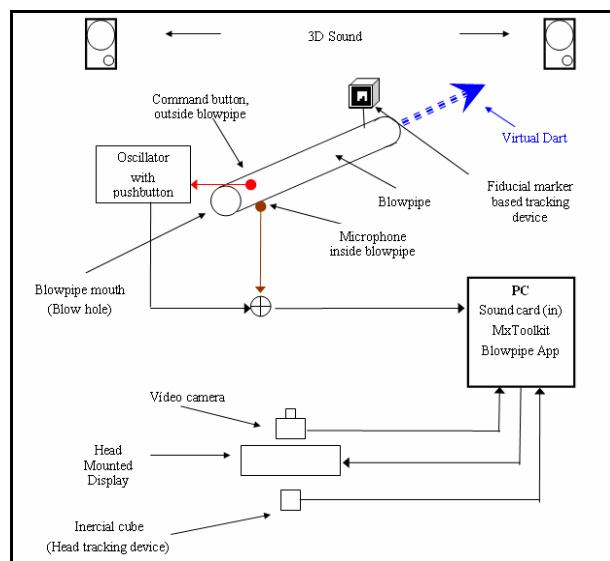


Figura 5 Arquitectura do Sistema de suporte à Interface Tangível Zarabatana.

Na Figura 5, podemos visualizar um diagrama do sistema proposto. É dada maior ênfase à integração da interface tangível Zarabatana, pois o MxToolkit já fornece o suporte para os restantes elementos que fazem parte deste diagrama. Este sistema assenta sobre uma Arena e o software é desenvolvido em C++ (.net), utilizando a plataforma MxToolkit.

A construção da Zarabatana, requer um microfone, para detectar e estimar o fluxo de ar soprado e um circuito oscilador que será activado quando se premir um botão de controlo, que irá existir no corpo da zarabatana, servindo como botão de controlo. Este dispositivo é uma alternativa viável à turbina utilizada no sistema de Yasuhiro [Yasuhiro03]. O sinal produzido por este oscilador é misturado com o sinal proveniente do microfone e, de seguida, enviado por um cabo até à entrada “Mic IN” da placa de som do computador. No extremo de saída da zarabatana, vai existir uma marca fiducial, a qual é uma das faces do “cubo mágico”, e com a qual é possível controlar a orientação da zarabatana, via seguimento baseado em visão. É com base nesta marca que é feita a colocação e alinhamento do dardo virtual sobre a zarabatana física.

No PC de suporte à aplicação, para além do MxToolkit que fornece suporte à captura e tratamento das imagens adquiridas pela câmara de vídeo, ao cubo de inércia e ao H.M.D, entre outros, vai ainda existir uma aplicação designada por “BlowPipeApplication” que vai proporcionar o controlo da zarabatana e o desenrolar do jogo. Esta aplicação existe sob a forma de uma DLL e pode ser carregada e descarregada dinamicamente na Arena, através dos serviços disponibilizados pelo MxToolkit.

3.1 Configuração do Sistema

As plataformas de *hardware* e *software* necessárias para o nosso sistema são as seguintes:

Hardware:

- *Video See-Through Head Mounted Display*: Olympus Eye-trek FMD 700, Multimedia Glasses (800x600 pixel), com uma ligação de vídeo sem fios entre o computador de suporte e a câmara de vídeo miniatura: “2Micro Cam wireless ultra miniature video camera” da Swann Security. (O sistema também pode suportar uma Web Cam, no entanto os cabos de ligação, provocam perda de mobilidade).
- *Wearable Unit*: Blusão equipado com uma ligação bidireccional de vídeo analógico sem fios e baterias.
- PC de suporte: *Hardware* - CPU: Intel Pentium IV 3.0 GHZ; RAM: 1 Gbyte; Placa gráfica card: ATI RADEON 9700 64 Mbyte; Placa de som Sound Blaster ou compatível.
- Interface Tangível Zarabatana

Software:

- MS Visual Studio .Net enterprise edition; MxTool-kit; Video Input - Direct X 9.0b; Graphics- Open GL and Open VRML.
- BlowPipeApplication.dll.

4. A INTERFACE TANGÍVEL ZARABATANA

A Zarabatana é constituída por um tubo oco onde estão montados um microfone e um oscilador controlado por um botão de pressão montado no exterior da mesma. Na Figura 6 podemos observar o diagrama de blocos do *hardware* da zarabatana.

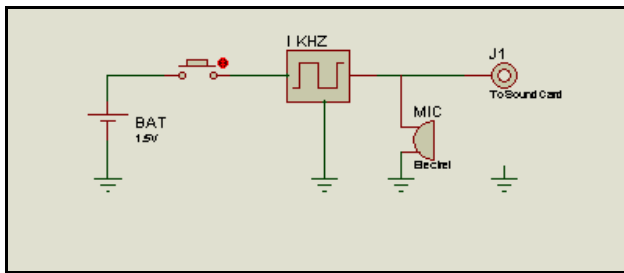


Figura 6 Diagrama do *Hardware* da Zarabatana.

O microfone tem como objectivo detectar e estimar o fluxo de ar soprado pelo jogador. Esta detecção e estimativa são efectuadas pelo *software*, recorrendo à extracção das componentes de baixa frequência, do sinal capturado pelo microfone, através da “Fast Fourier Transform” (F.F.T.) e posterior cálculo do seu valor médio. Com base na amplitude do valor médio calculado, é determinada a velocidade de partida do dardo virtual. Em paralelo com o microfone, existe um oscilador com uma frequência fixa de 1KHz que é activado sempre que se prime o botão de controlo existente na zarabatana, visando a criação de um botão de controlo. Este botão de pressão fecha o circuito que fornece energia ao oscilador, logo quando o botão está na sua posição de descanso, o circuito não consome energia (Ver Figura 7). Este botão, no decorrer

da aplicação, tem como principal função comandar o início do jogo e permitir a mudança de tipo de dardo quando o jogo está a decorrer, sem que o utilizador tenha de se deslocar até ao teclado do computador.

Uma vez que o sinal produzido pelo oscilador está dentro da gama vocal, o acto de premir do botão, para dar início ao jogo, ou para mudar de dardo, pode ser substituído por um assobio cuja frequência fundamental esteja nas proximidades da frequência do oscilador (1 KHz). É também com base na F.F.T. que é detectada a entrada em funcionamento deste oscilador/assobio, sendo calculados os valores médios em torno da zona do espectro que corresponde a 1KHz. Este modo de funcionamento do botão de controlo, pode apresentar alguns inconvenientes, mas permite simplificar a ligação entre a zarabatana e o computador, e dar maior liberdade de movimentos ao utilizador na Arena. Os inconvenientes mencionados, estão relacionados com a eventual activação do comando por uma fonte exterior ao jogo e também com os diversos problemas de compatibilidade que podem ocorrer, dado a existência de sensibilidades diferentes nas placas de som usadas em cada computador.

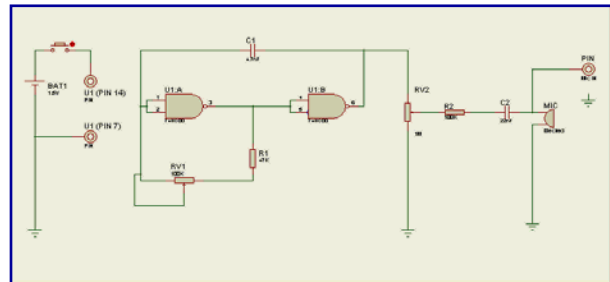


Figura 7 Esquema Electrónico do *Hardware* da Zarabatana.

4.1 O Esquema Eléctrico da Zarabatana

Atendendo à grande simplicidade do hardware envolvido, é aqui apresentado o seu esquema electrónico (ver Figura 7) e, ainda, uma pequena descrição do circuito, efectuada da esquerda para a direita. Assim, na zona mais à esquerda, podemos constatar que o botão de pressão de controlo da zarabatana, está ligado em série com os terminais de alimentação do circuito integrado que suporta os elementos activos do circuito. Este facto permite que o circuito em repouso não consuma energia, permitindo assim uma maior longevidade para a pilha utilizada. Esta é uma pilha com formato de “botão”, com uma tensão de 1,5 V que por sua vez alimenta um circuito integrado de tecnologia CMOS 74HC00 [74HC00]. Este circuito tem de ser obrigatoriamente da família HC, pois esta é a única que pode funcionar com a tensão fornecida pela pilha usada. O consumo deste tipo de circuito é muito baixo, na ordem dos micro amperes, o que contribuirá para uma grande longevidade da pilha. O circuito oscilador é formado por um circuito muito conhecido baseado em dois inversores interligados, um condensador (C1) e uma resistência (R1+ RV1). A frequência resultante deste tipo de montagem, pode ser calculada de forma aproximada pela expressão $f_0 = \frac{1}{2,3RC}$. O resto do circuito, à direita do oscilador, apresenta um divisor de tensão,

constituído por RV2 onde é possível regular a amplitude do sinal quadrado que vai ser injectado no circuito do microfone, enquanto R2 e C2 limitam os valores máximos admissíveis e efectuem uma filtragem de corrente continua respectivamente. Não foram considerados outros cuidados de construção, nomeadamente ao nível de uma filtragem mais refinada do sinal produzido, pois este irá posteriormente ser alvo de uma filtragem digital pelo nível de *software*. De seguida, no circuito, aparece montado em paralelo com o oscilador, o elemento microfone, que não é mais do que um microfone capacitivo *padrão*, com características unidireccionais. O sinal resultante deste circuito é então enviado para a porta de entrada Mic IN [Sound Blaster, Mic IN, 04] da placa de som do computador de suporte à aplicação.

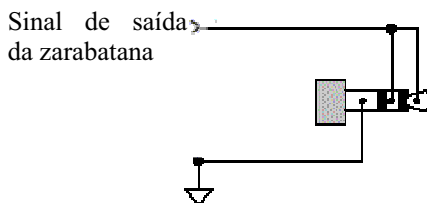


Figura 8 Diagrama da Ligação da Zarabatana à Placa de Som.

Na Figura 8 podem ser observadas as ligações da zarabatana à placa de som “Sound Blaster” através do conector “Mic. IN”.

4.2 Construção do Protótipo da Zarabatana

Na parte superior da Figura 9 podemos visualizar o aspecto final do circuito electrónico da zarabatana. O aspecto alongado da placa, deve-se ao facto de esta estar destinada a ser colocada no interior da zarabatana.

A zarabatana tem um tamanho de aproximadamente 50 cm e foi pensada para se adequar de forma satisfatória às dimensões da Arena onde vai ser utilizada.

4.3 O Seguimento da Zarabatana

O seguimento da zarabatana dentro do campo de visão da câmara é efectuado recorrendo a apenas uma das faces do cubo mágico.

Baseado na imagem capturada pela câmara, a API MX Toolkit do referido cubo, fornece ao programador a pose relativa da sua face mais visível, face à orientação e posição perpendicular da câmara. Matematicamente, essa pose é a produto de uma matriz de transformação de translação (cujos parâmetros equivalem à distância entre os dois referenciais em causa, o da marca e o da câmara), com uma matriz de transformação de rotação (cujos parâmetros equivalem aos ângulos formados entre os versores correspondentes de ambos os referenciais).

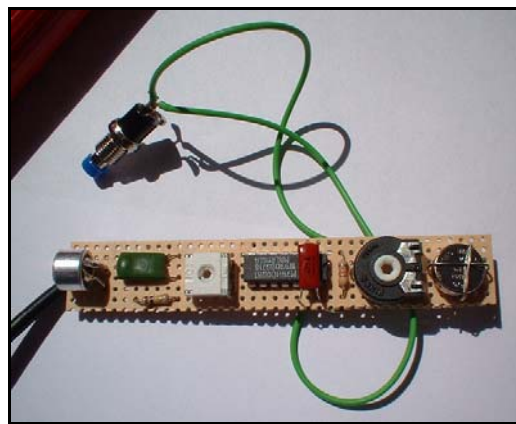


Figura 9 Construção do Hardware do Protótipo da Interface Tangível Zarabatana.

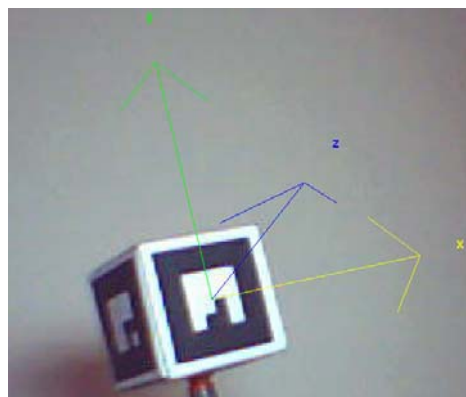
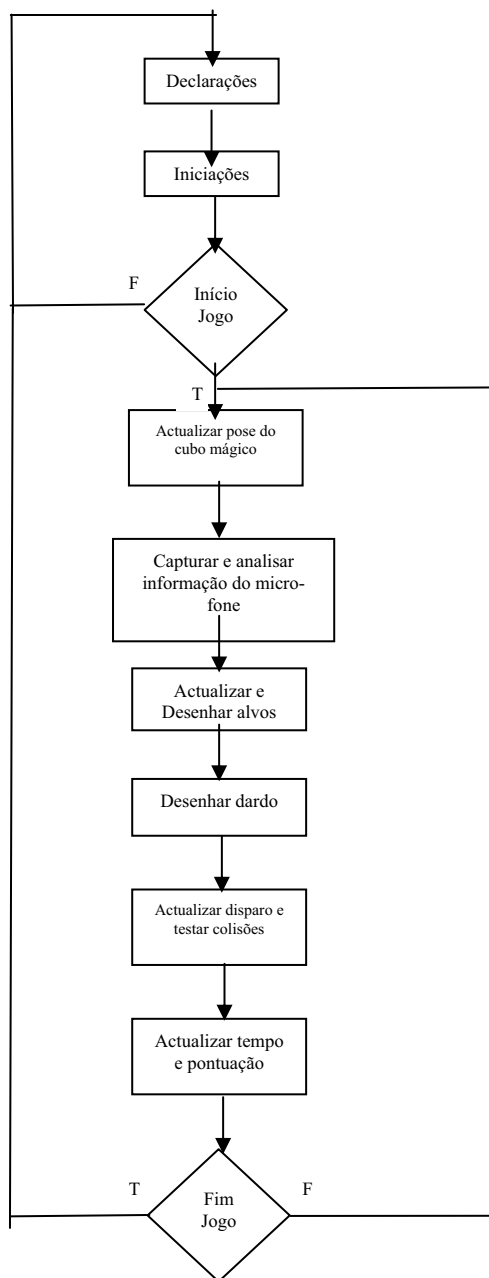


Figura 10 O Cubo Mágico.

Com base nesta informação é possível fazer o posicionamento do dardo virtual sobre a zarabatana física. De modo a facilitar as tomadas de vista sobre os alvos e estimar as trajetórias, a face do cubo utilizada é posicionada na lateral direita da zarabatana e não na sua parte superior como se poderia supor numa primeira abordagem.

O ciclo principal do algoritmo da Zarabatana é baseado no diagrama que de seguida se apresenta. Este ciclo de jogo deveria em condições óptimas acontecer 25 vezes por segundo.



alvos existentes, com o tipo de dardo que considerar mais adequado para abater esses alvos, sabendo que o dardo sensível à gravidade, apresenta valores de pontuação mais elevados, quando do abate de um mesmo alvo. A pontuação conseguida pelo jogador é contabilizada em tempo real, sendo apresentada na linha de estado do jogo. O objectivo deste jogo, como já foi referido, é o de conseguir obter a maior pontuação possível, num determinado tempo fixo.

De referir que, no computador, o volume associado ao microfone deve ser regulado para 50% do seu valor máximo e devem ser desactivadas as opções de *Boost* e *Microfone* de painel frontal.

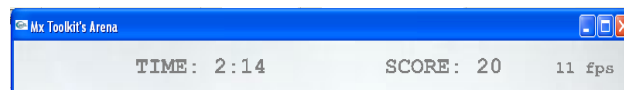


Figura 11 Linha de Estado do Jogo.

A linha de estado do jogo é visualizada na zona superior do ecrã, o que permite ao jogador obter, em tempo real, informações sobre o estado de início, fim de jogo e estado do desenrolar do jogo, nomeadamente o tempo que ainda dispõe para jogar e a pontuação obtida até ao momento. O indicador do número de imagens por segundo é apenas uma medida de capacidade de processamento e não faz parte do jogo,



Figura 12 Interagindo com o Jogo *BlowPipe*.

5. RESULTADOS E DISCUSSÃO

5.1 O Jogo *BlowPipe*

A Zarabatana disponibiliza dois dardos intermutáveis, com o auxílio do botão de controlo, com cores e comportamentos diferentes (Figura 9). O dardo verde é insensível à gravidade, e possui uma velocidade de deslocamento que não depende do valor do volume de sopro. O dardo vermelho é sensível à aceleração da gravidade e a sua velocidade depende da quantidade de ar soprado pelo jogador.

Quando se carrega o jogo na Arena, sob o controlo do MxToolkit, este fica em espera até que se carregue no botão de controlo da Zarabatana. Quando este botão for premido é dado início ao jogo, sendo iniciada uma contagem decrescente de tempo com um valor de 3 minutos. Durante este tempo o jogador pode optar por atingir os



Figura 13 Pormenor do Jogo *BlowPipe*.

As Figuras 12 e 13 mostram um ecrã de jogo, onde é visível um fundo e uma zarabatana físicos, sobre os quais estão representados um pássaro e um dardo virtual. A posição do dardo é controlada pela manipulação da Zarabatana, com a limitação que a marca na ponta da mesma deve estar visível.

5.2 Discussão

Uma das principais limitações do trabalho realizado deve-se à pouca fiabilidade que o microfone apresenta para a medição eficaz do volume de ar soprado. Devido à concepção do *software*, constata-se que esta medição é tanto menos fiável, quanto menor for o valor do número de imagens por segundo apresentadas.

Por outro lado, devido à técnica de seguimento adoptado para a Zarabatana (baseado no AR Toolkit), sempre que a marca fique parcialmente oculta, é perdida a posição do dardo.

A colocação na Arena de mais de 4 alvos, com qualidade gráfica média (correspondente a modelos VRML com 1 Mbyte cada), não permite um número de imagens por segundo aceitáveis para efectuar um jogo, na configuração utilizada para o testar. Tal deve-se ao facto de, tanto o controlo da Zarabatana, como o seguimento do utilizador na Arena, ou ainda, o cálculo das imagens sintéticas que se misturam com as imagens reais, terem sido calculadas no mesmo processador. Esta condicionante não tem necessariamente de ser assim e não se refere a nenhuma limitação da arquitectura dado que o MX Toolkit possibilita uma arquitectura distribuída [Dias03a], com a qual podemos distribuir os cálculos referidos por computadores distintos interligados numa rede local ETHERNET, o que faremos em testes e estudos de usabilidade futuros.

6. CONCLUSÕES E TRABALHO FUTURO

O sistema apresentado permite a exploração de um novo conceito de diversão baseado, na utilização de uma zarabatana física e dardos virtuais, a qual possibilita a interacção com um cenário aumentado constituído por alvos em movimento. A interface tangível Zarabatana, foi testada num ambiente de realidade mista, concretizado num jogo de lançamento de dardos ao alvo.

Com este trabalho ficam demonstradas as capacidades do MxToolkit para o desenvolvimento rápido de aplicações de realidade aumentada e mista, cuja interacção decorre na Arena. Também fica demonstrado que com a capacidade de processamento disponível num computador actual, é possível construir sistemas lúdicos simples, baseados em realidade aumentada e mista, o que abre boas perspectivas para o futuro deste tipo de sistemas.

O enriquecimento do ambiente em termos de maior qualidade gráfica e a utilização de um número mais elevado de alvos em movimento apresenta no entanto ainda algumas limitações.

Como trabalhos futuros, podemos considerar:

- A construção de um jogo de tiro ao alvo para vários jogadores em simultâneo;

- A separação do processamento por vários computadores para que o sistema desenvolvido possa integrar bastantes mais alvos.
- A avaliação da usabilidade do jogo, numa plataforma computacional distribuída e de maior desempenho gráfico, visando a avaliação de métricas de sucesso atingidas pelos utilizadores durante o jogo, tais como a % de alvos atingidos por sessão, a duração do jogo ou ainda a avaliação subjectiva da sensação de imersão no uso da zarabatana como interface tangível.

Com a integração de um sistema de seguimento no interior da zarabatana e com a substituição do microfone, por um elemento mais eficaz na captura do sopro é possível considerar a construção de um protótipo com fins comerciais.

7. AGRADECIMENTOS

Os autores, gostariam de agradecer a Pedro Santos, um dos elementos criadores do MX Toolkit, sem o qual este trabalho não teria sido realizado, bem como a Marisa Aurélio e Pedro Nande, todos da ADETTI-ISCTE, pela colaboração prestada com o desenvolvimento do Cubo Mágico.

8. REFERÊNCIAS

[artoolkit] <http://www.hitl.washington.edu/artoolkit/>

[ARTtrack2] <http://www.ar-tracking.de>

[Azuma 97] Azuma R. T.. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, Aug. 1997.

[Dias03a] Dias, J., M., S., Santos, P., Monteiro, L., Silvestre, R., Bastos, R., “Developing and Authoring Mixed Reality with MX Toolkit”, ART03, The Second IEEE International Augmented Reality Toolkit Workshop, Tokio, Japão, 6 Outubro 2003.

[Dias 03b] Dias, J. M. S., Aurélio, Rosário, Carvalho, H., L., M., Rocha, I., Carvalho, L. S., “Augmented Reality Visualisation of the Left Ventricle of the Heart, using a Tangible Interface”, Reunião Científica Anual da Sociedade Portuguesa de Biologia, Instituto Bento da Rocha Cabral, Novembro de 2003.

[Dias 04a] Dias, J. M. S., Bastos, R., Santos, P., Monteiro, L., Canhoto, J., “The Arena: An Indoor Mixed Reality Space”, Actas da 1ª Conferência Nacional em Interação Pessoa-Máquina, Faculdade de Ciências, Universidade de Lisboa, Lisboa, 12-14 Julho 2004.

[Dias04b] Dias, J. M. S., Jamal, N., Silva, P., Bastos, R., “ARTIC: Augmented Reality Tangible Interface by Color Evaluation”, Actas da 1ª Conferência Nacional em Interação Pessoa-Máquina, Faculdade de Ciências, Universidade de Lisboa, Lisbon, 12-14 Julho 2004.

[Yasuhiro03] Okuno, Hiroyuki Kakuta, and Tomohiko Takayama: “Jellyfish Party: Blowing Soap Bubbles in Mixed Reality Space”, ISMAR03, Tokio, Japão, 7-9 Outubro 2003.

[Sound Blaster, Mic IN], texto online em Julho 2004,

http://www.epanorama.net/links/pc_sound.html#soundcard_general

[74HC00], texto *online* em Julho 2004,
http://www.semiconductors.philips.com/acrobat/datasheets/74HC_HCT00_3.pdf
[isense] <http://www.isense.com>

TARCAST: Uma Taxonomia para Sistemas de Realidade Aumentada

José Braz Pereira
Dep. Sistemas e Informática, EST Setúbal / IPS
Setúbal
jbraz@est.ips.pt

João Madeiras Pereira
Dep. Engª Informática, IST / UTL
Lisboa
jap@inesc.pt

Resumo

O presente trabalho discute as definições de Realidade Aumentada (RA) mais comuns assim como a arquitectura de um sistema de RA típico e conclui pela necessidade de a expandir através de componentes que permitam uma interacção háptica com o ambiente aumentado. Em seguida propõe-se uma taxonomia para caracterização e classificação de sistemas de RA que permite compará-los em termos de tecnologias e metodologias utilizadas. A taxonomia proposta baseia-se no princípio de que qualquer sistema de RA é potencialmente composto por seis subsistemas e selecciona os critérios de caracterização para cada um deles de acordo com taxonomias existentes. Finaliza-se descrevendo a ferramenta de suporte à utilização da taxonomia através da web, ferramenta esta que permite caracterizar um sistema de RA através de uma interface baseada em menus e consultar a informação assim disponibilizada.

Palavras Chave

Realidade Aumentada, Realidades Misturadas, Taxonomia, Caracterização.

1 INTRODUÇÃO

1.1 Motivação

Com o objectivo de propor um ambiente de Realidade Aumentada (RA), para visualização dos dados biomecânicos recolhidos pelo Laboratório de Biomecânica da FMH, ambiente esse já implementado e descrito em [Braz, 2005] efectuou-se um estudo dos sistemas de RA até então desenvolvidos.

Ao tentar sistematizar esse estudo, tornou-se evidente a inexistência de uma taxonomia apropriada para a comparação dos diferentes sistemas se pretendessemos ter como critérios comparativos as tecnologias e metodologias envolvidas. Tal verificou-se ser essencial para tomar decisões quanto ao sistema a implementar, sentimento este também realçado em [Dubois et Al., 1999a] onde se propõe um Espaço Classificativo para Cirurgia Aumentada.

1.2 Objectivos

De acordo com [Milgram et Al., 1994], o objectivo de uma taxonomia é oferecer uma classificação de acordo com a qual possam ter lugar discussões de ordem não só teórica mas também prática, avaliados os desenvolvimentos de uma determinada área, conduzir e orientar a investigação para as sub-áreas pertinentes e comparar informação de forma útil.

Mas, acrescentaríamos que, principalmente, ela tem de ser usada. Para tal parece-nos que essa taxonomia deve:

1. à semelhança das mais conhecidas taxonomias, como sejam a Tabela Periódica dos Elementos ou a Taxonomia Numérica para classificação das espécies, permitir classificar os sistemas de acordo com critérios “objectivos” e, a partir da classificação obtida, inferir características qualitativas,
2. ser suficientemente simples para ser entendida por todos os intervenientes de forma intuitiva, baseada em critérios de classificação já aceites, citados e usados, por forma a permitir enquadrar as descrições dos sistemas desenvolvidos da forma mais directa possível nos critérios caracterizadores da taxonomia,
3. considerando o carácter multidisciplinar da RA, reflectir e incorporar o conhecimento dos especialistas das diferentes áreas abordadas pela RA,
4. preferencialmente ser suportada por uma ferramenta que facilite a caracterização dos diferentes sistemas.

Mais do que classificar os sistemas existentes de acordo com uma nova taxonomia, o TARCAST pretende ajudar a comunidade de investigação e desenvolvimento a extrair a informação relevante sobre sistemas de RA já existentes.

2 DEFINIÇÃO E TERMINOLOGIA

Em [Milgram et Al., 1994] é proposta uma Taxonomia para Sistemas de Apresentação de Realidades Misturadas que introduz o conceito de “Continuum de Virtualidade” (Figura 1), ao longo do qual se distribuem diferentes tipos de Sistemas de Realidades Misturadas.

Considerando a RA como um caso particular no

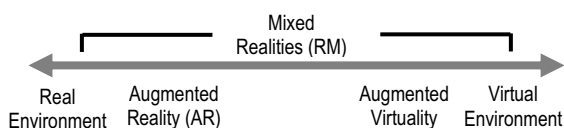


Figura 1: O “Continuum de Virtualidade” de Milgram.

universo das Realidades Misturadas, Milgram insere-a no lado esquerdo desse continuum, mais perto do ambiente real do que dos mundos virtuais. Este enquadramento permite-lhe justificar a clara separação entre as áreas da RA e da Realidade Virtual, e, em simultâneo, abrir campo para o conceito genérico de Realidades Misturadas que ocupa assim todo o espaço entre o Ambiente Real e o Ambiente Virtual.

2.1 Definição

Aceitando explicitamente o enquadramento da RA proposto em [Milgram et Al., 1994], Ronald Azuma, em [Azuma, 1997] e [Azuma et Al., 2001], acaba por definir a RA como «uma variante da Realidade Virtual [...]». As tecnologias de Realidade Virtual imergem completamente o utilizador num ambiente sintético e, enquanto imerso nesse ambiente, o utilizador não pode ver o mundo real que o rodeia. Pelo contrário, a RA permite ver o mundo real com objectos virtuais sobrepostos ou integrados nesse mundo real. Desta forma, a RA complementa a realidade em lugar de a substituir.[...]».

Evitando a conotação da RA com qualquer tipo de tecnologia, Ronald Azuma aponta três características fundamentais e distintivas para um sistema de RA, nomeadamente:

1. Combina o real e o virtual.
2. É interactivo em tempo real.
3. As imagens reais e virtuais são alinhadas espacialmente e sincronizadas temporalmente num espaço tridimensional (*registered*).

Note-se que esta definição, se bem que genericamente aceite, não é absolutamente consensual. Refira-se, a este propósito, que [Dubois et Al., 1999a] defende que a variedade de domínios aplicacionais torna difícil chegar a uma definição consensual de RA e que diferentes pessoas, com diferentes objectivos, usam o termo RA aplicando-o a diferentes sistemas com diferentes características, nem todos englobados na definição de Azuma.

De notar ainda que, devido à «adequabilidade do termo, [...] é comum empregar o termo RA para caracterizar qualquer tipo de apresentação em que um mundo, de outra forma real, é aumentado com informação virtual. Integram-se nesta classe os sistemas em que os objectos virtuais são sobrepostos a imagens de vídeo do mundo real, captadas por uma câmara estática ou não e apresentadas num écran bidimensional. Autores mais preocupados em respeitar a semântica vulgarmente

aceite, optam por denominar esta última classe de sistemas como sistemas de “Realidade Melhorada” (*enhanced reality*)» [Milgram et Al., 1994].

Pela nossa parte, relevando a preocupação em analisar os mais variados tipos de sistemas, preferimos considerar a definição restritiva de Azuma como uma “utopia” a perseguir e englobar nos sistemas de RA qualquer forma de sobreposição de imagens virtuais a imagens reais seguindo a interpretação mais alargada de, entre outros, [Milgram et Al., 1994] «Como definição operacional de RA, tomamos o termo como referência a qualquer caso em que um ambiente, de outra forma real, é “aumentado” através de objectos virtuais (gerados por computador) [...]».

2.2 Terminologia

Ao definir a semântica de vários termos vulgarmente usados em RA, Paul Milgram, numa abordagem dicotómica, propõem três tipos de distinções:

1. Objectos Reais (OR) versus Objectos Virtuais (OV).
2. Visualização Indirecta (VI) e Visualização Directa (VD) (a *Unmediated Reality* de [Naimark, 1991].)
3. Imagens Reais e Imagens Virtuais

Neste âmbito define:

1. Os OR como sendo objectos que possuem uma existência objectiva, por oposição aos OV, os quais existem em essência ou efeito, mas não de facto. Desta forma, os objectos reais tanto podem ser observados directamente, como re-sintetizados num dispositivo de apresentação a partir de uma amostragem do objecto real, enquanto os objectos virtuais, só podem ser visualizados a partir de uma simulação/modelo e sintetizados num dispositivo de apresentação.
2. A VD como tendo lugar através do vidro ou ar e que se aplica apenas a objectos reais, por oposição à VI que recorre obrigatoriamente a um dispositivo de amostragem e a um dispositivo de apresentação.

Também a terminologia proposta por Milgram nos parece passível de alguma re-elaboração, nomeadamente no que concerne às dicotomias entre OR e OV e entre VD e VI.

Ao distinguir Objectos Reais de Objectos Virtuais, nota-se que Milgram tenta fugir à conotação dos objectos reais com o mundo físico, atómico se quisermos, optando antes pelo conceito de objectividade. Quanto a nós, esta segunda abordagem acaba por introduzir tanta, ou mais, “indeterminação” quanto uma definição baseada na dicotomia átomo/bit. Desta forma, e tanto mais que outros autores, como [Dubois et Al., 1999a] preferem também esta abordagem, propomos uma distinção entre objectos reais e virtuais, em tudo semelhante, mas baseada em diferentes critérios:

Propomos assim que se considerem os Objectos Reais como sendo aqueles que possuem uma estrutura atómica¹, por oposição aos Objectos Virtuais, que existem apenas sob a forma de informação² armazenada num qualquer sistema.

Seguindo este critério distintivo, quanto a nós mais “objectivo”, mantém-se o fundamental do conceito definido por Paul Milgram e explicitado no segundo parágrafo do ponto 1.

Também no que respeita à distinção entre VD e VI, o recurso a conceitos físicos fundamentais permite-nos obter uma definição semanticamente equivalente, mas, quanto a nós, mais fortemente sedimentada. Podemos desta forma definir a Visualização Directa, como sendo produto da luz proveniente directamente do objecto real, por oposição à Visualização Indirecta que requer dispositivos de amostragem e de apresentação.

2.3 Um Sistema Típico de RA

Considerando, por um lado, a relevância do conceito de “Extensão da metáfora de presença” usado por Paul Milgram, como um dos critérios de classificação para sistemas de apresentação de realidade misturada e, por outro lado, que a tecnologia usada para apresentar a fusão dos universos, o real e o virtual, é determinante para a qualidade da imersão proporcionada por um sistema de RA, não é de estranhar que os sistemas de RA tenham vindo a ser classificados de acordo com o sistema de apresentação usado.

Vallino na sua “*Introduction to Augmented Reality*” [Vallino, 2001], distingue três tipos de sistemas de acordo com o dispositivo de apresentação usado:

1. Os baseados num monitor (Window on the World ou Fish Tank),
2. os baseados num HMD a que é acoplada uma câmara de vídeo (Video See Through)
3. e os baseados em HMD de lentes semireflexivas (Optical See Through)

A estes três tipos de sistemas de RA, podemos já acrescentar um quarto:

4. Os sistemas que recorrem a projectores para apresentar os objectos virtuais directamente no mundo real, como os sistematizados em [Billinghurst et Al., 1999], essencialmente vocacionados para sistemas colaborativos como as “*Augmented Surfaces*” [Rekimoto et Al., 1999]. (Projectivos)

É ainda em [Vallino, 2001] que encontramos a representação de um sistema típico de RA onde se podem distinguir claramente cinco subsistemas:

1. Sistema de captação de imagem (SCI).

2. Sistema de seguimento da localização e orientação do SCI (SSSCI).
3. Sistema Gráfico para geração dos objectos virtuais (SG).
4. Sistema Misturador de Imagem (SMI).
5. Sistema de Apresentação (SA).

No entanto, se pretendermos alargar a noção de interactividade para além da mera navegação no mundo aumentado e considerar a existência de objectos reais em movimento na cena aumentada, o que acontece em vários dos sistemas de RA analisados, parece-nos relevante acrescentar outros três subsistemas, nomeadamente:

1. Sistema Real de Manipulação de Objectos. Que tanto poderá manipular objectos reais como virtuais.
2. Sistema de seguimento do SRMO.
3. Sistema de seguimento de objectos reais que se movimentem no ambiente aumentado.

Acrescentando o primeiro ao sistema típico de Vallino e incorporando os dois últimos num sistema de seguimentos global, um sistema típico de RA passará a apresentar uma composição do tipo da apresentada na Figura 2.

3 TARCAST: TAXONOMIA PROPOSTA

Uma possível abordagem para a definição de uma taxonomia dos sistemas de RA, basear-se-á assim, na criação de um quadro em que os seis subsistemas identificados possam ser caracterizados de acordo com taxonomias pertinentes.

Abordam-se em seguida vários aspectos da taxonomia proposta, nomeadamente os princípios em que se baseia, a sintaxe proposta e os valores semânticos de cada característica identificada.

3.1 Princípios:

1. Apesar de a “ponta visível do iceberg” consistir nos diferentes elementos classificados de acordo com a taxonomia, a parte relevante de uma taxonomia consiste nas características identificadas que permitem fazer a distinção entre sistemas e portanto conduzem à sua classificação. Por outras palavras, o fundamental são os critérios que dão lugar à classificação.
2. Qualquer sistema de RA é potencialmente composto pelos seis subsistemas identificados na Figura 2.
3. Nos casos em que os seis subsistemas não sejam fisicamente independentes, é sempre possível separá-los conceptualmente, identificá-los e caracterizá-los.
4. Mais do que em classificar, o TARCAST preocupa-se em extrair de cada subsistema os parâmetros necessários a uma caracterização o mais objectiva possível das potencialidades e limitações de cada tecnologia ou metodologia usadas.

¹ Ou baseada em qualquer outro tipo de micropartículas.

² Aqui entendemos informação no estrito sentido de uma entidade expressa em bit's ou em qualquer outra unidade de informação analógica.

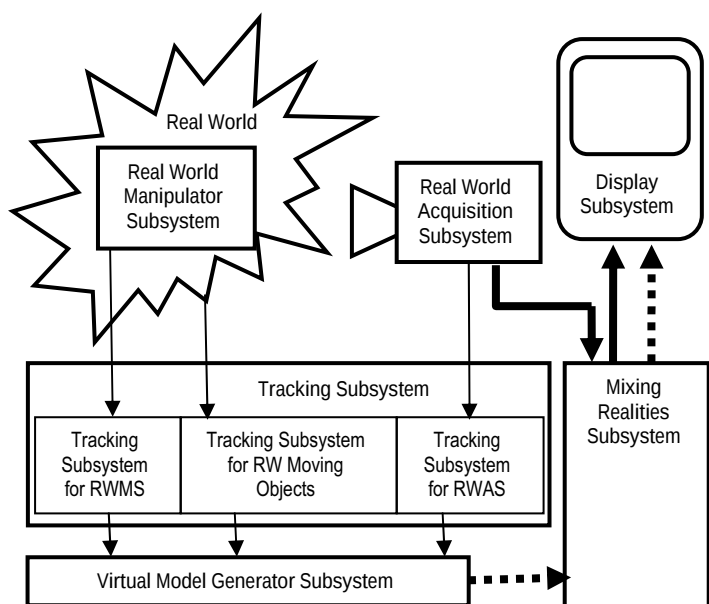


Figura 2: Arquitectura de um Sistema de RA Típico

5. Sempre que exista uma taxonomia geralmente aceite e aplicável, no sentido de oferecer critérios classificativos tecnológicos ou metodológicos, recorre a essa taxonomia para classificar os subsistemas.

3.2 Caracterização de um Sistema de RA

Como se pode ver pela sintaxe apresentada na Figura 3, os sistemas de RA são caracterizados de acordo com os seis subsistemas identificados na Figura 2 e por um grupo de características globais que se apresentam em seguida.

```
<DEF AR System> ::=
  <DEF Global Characteristics> ^p
  <DEF Real World Acquisition Subsystem> ^p
  <DEF Virtual Model Generator Subsystem> ^p
  <DEF Mixing Realities Subsystem> ^p
  <DEF Display Subsystem> ^p
  <DEF Real Manipulator Subsystem> ^p
  <DEF Tracking Subsystem> ^p
```

Figura 3: Caracterização Global de um Sistema de RA.

3.3 Características Globais de um Sistema de RA

As características não associáveis a um dos subsistemas são consideradas características globais do Sistema de RA e a sua sintaxe apresentada na Figura 4.

As seis características globais identificadas são:

1. Chave: número inteiro que identifica univocamente cada sistema de RA.
2. Título: nome dado ao Sistema de RA pelos seus autores, na sua ausência deve recorrer-se a um nome usado numa referência bibliográfica.
3. Áreas aplicacionais: áreas em que o sistema de RA encontra aplicação ou para as quais foi projectado.

4. Tipo de Sistema: Se é uma metodologia, uma tecnologia ou uma aplicação baseada ou não numa das plataformas de desenvolvimento de Sistemas de RA existentes.
5. Número de utilizadores simultâneos: quantidade de utilizadores que pode interagir com o sistema em cada momento.
6. Interação entre utilizadores: é ou não possível a interação entre utilizadores.

```
<DEF Global Characteristics> ::=
  <key> - <title> - <year> ^p
  <application fields> ; ^p
  <system type> ; ^p
  <number of simultaneous users> ; ^p
  <interaction between users> ; ^p
  <global link> ; ^p
  <global description> ; ^p
  <global references> . ^p

<key> ::= Id: Integer
<title> ::= Title: <plain text>
<year> ::= 19 <digit> <digit> | 20 <digit> <digit>
<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<application fields> ::= Application Fields: <af keyword> , <af keyword>
<af keyword> ::= <plain text>
<system type> ::= System Type: <st keyword>
<st keyword> ::= Methodology> | Technology> | Application
<number of simultaneous users> ::=
  Number of Simultaneous Users: <quantity>
<quantity> ::= 1 | 2 | Several
<interaction between users> ::=
  Interaction Between Users:
    <type of interaction between users>;
<type of interaction between users> ::= None | Collaborative
<global link> ::= Global Link: (<URL>)
<global description> ::= Global Description: (<plain Text>)
<global references> ::= Global References: (<plain Text>).
```

Figura 4: Características Globais de um Sistema de RA

3.4 Sistema de Captação do Mundo Real

O Sistema de Captação do Mundo Real pode ser caracterizado em termos dos sub-sistemas que permitem excitar os respectivos sentidos do ser humano com sinais provenientes do mundo real, tal como se apresenta na Figura 5. Por ser de especial relevância para o estudo então realizado, desenvolveu-se com maior detalhe a caracterização do subsistema de captação de Imagem que se apresenta em seguida.

Sistema de Captação de Imagem

Apesar de aplicável na sua generalidade à classificação de sistemas de captação de imagem, a taxonomia proposta em [Naimark, 1991] salienta a correlação entre as tecnologias de captação e de apresentação de imagem.

Na taxonomia proposta preferimos separar claramente os sistemas de captação dos sistemas de apresentação e deixar que, caso isso se verifique, a correlação entre

eles transpareça aquando da caracterização dos diferentes sistemas implementados ou da classificação das tecnologias usadas.

```
<DEF Real World Acquisition Subsystem> ::=
  <DEF Image Acquisition Subsystem> ^p
  <DEF Sound Acquisition Subsystem> ^p
  <DEF Smell Acquisition Subsystem> ^p
  <DEF Other Acquisition Subsystem> ^p

<DEF Sound Acquisition Subsystem> ::= Sound Acquisition Subsystem: <yn>
<DEF Smell Acquisition Subsystem> ::= Smell Acquisition Subsystem: <yn>
<DEF Other Acquisition Subsystem> ::= Other Acquisition Subsystem: <yn>

<yn> ::= Yes | No
```

Figura 5: Caracterização de um Sistema de Captação do Mundo Real

Para a classificação dos Sistemas de Captação de Imagem foram identificadas as seguintes cinco características relevantes:

1. Tipo de Captação: O conceito de Tipo de Captação coincide completamente com a definição de Visão Directa e Indirecta proposta em [Milgram et Al., 1994] e acima descrita.
2. Dinâmica: O SCI é considerado estático nos casos em que não se pode mover, o que implica que a interacção com o ambiente de RA se dá a outro nível que não o visual a fim de manter a coerência com a definição de [Azuma, 1997]. O SCI é considerado móvel em todos os restantes casos.
3. Dimensões da Imagem Recolhida: O conceito de Dimensões da Imagem Recolhida abarca as imagens monoscópicas, estereoscópicas e multiescópicas tal como definidas em [Naimark, 1991].
4. Calibração: Uma característica fundamental para a escolha do SCI a usar num sistema de RA é a sua dependência da necessidade de calibração prévia ou não. Neste sentido foram já desenvolvidas metodologias de utilização de SCI que não requerem a sua prévia calibração [Kutulakos et Al., 1998] [Seo et Al., 2000]. A grande maioria dos sistemas de RA documentados recorre a marcas de referência (Marcas Fiduciais), que vão permitir calcular a matriz de calibração da câmara, com recurso a métodos de Visão Computacional. Outro interessante caso documentado recorre a feixes de laser para calibrar a câmara [Nakajima et Al., 1998] [Braz, 2000]. Distinguem-se os sistemas que não necessitam de calibração, os que recorrem a marcas fiduciais e, numa terceira categoria, os que recorrem a sistemas activos de calibração, como sejam, por exemplo, feixes laser ou sistemas de ultrasons.
5. Tecnologia: O conceito de Tecnologia do SCI coincide com o tipo de dispositivo usado para captar a imagem do mundo real. Listam-se os diferentes dispositivos documentados na literatura [Azuma, 1997], [Milgram, 1994], [Braz, 2000], mas nada impede que <Outros> dispositivos sejam entretanto identificados e acrescentados à lista.

```
<DEF Image Acquisition Subsystem> ::=
  Image Acquisition Subsystem <IA system> ^p

<IA system> ::= <IA subsystem> | <IA subsystem> <IA system>

<IA subsystem> ::=
  <key> - <title> - <year> ; ^p
  <way of view> ;
  <dynamics> ;
  <captured image dimensions> ;
  <calibration> ;
  <technology> ; ^p
  <link> ; ^p
  <description> ; ^p
  <references> . ^p

<way of view> ::= Direct Vision | Indirect Vision

<dynamics> ::= Static | Mobil

<captured image dimensions> ::=
  Monoscopic Image |
  Stereoscopic Image |
  Multiscopic Image

<calibration> ::= Not Required | Fiducial Marks | Laser

<technology> ::=
  Human Eye | Video Camera | Microscope |
  Telescope | Magnetic Resonance

<link> ::= Link: <URL>

<description> ::= Description: ( <plain text> )

<references> ::= References: ( <plain text> )
```

Figura 6: Caracterização de um Sistema de Captação de Imagem

3.5 Sistema Gerador de Modelos Virtuais

Sendo tradicionalmente aceite para a classificação de ambientes virtuais, a taxonomia proposta em [Zeltzer, 1992] serviu como base para a classificação dos objectos virtuais presentes num ambiente de RA. Apresentam-se em seguida os parâmetros caracterizadores que poderão vir a permitir, através de, por exemplo, um simples método de ponderação de coeficientes, classificar os ambientes virtuais presentes em sistemas de RA com a citada taxonomia. Note-se, no entanto, que alguns outros aspectos relevantes para a caracterização dos modelos virtuais usados em sistemas de RA não se enquadram, pelo menos explicitamente, na classificação de Zeltzer, como seja o conhecimento sobre o mundo real.

A caracterização do Sistema Gerador de Modelos Virtuais é feita de acordo com os sentidos que permite excitar. A relevância dada à caracterização da componente gráfica justifica-se pelo âmbito do presente trabalho, mas nada impede a expansão dos restantes componentes.

Modelo Gráfico

A classificação do modelo gráfico faz-se de acordo com as seguintes seis características:

1. Grau de Realismo Gráfico: O modelo virtual pode ir do mais simples plano, usado em sistemas de etiquetagem do mundo real, até representações baseadas em imagens reais. Foram identificados sistemas com representações: 2D, | Fio de Arame a Cinzento, | Fio de Arame a Cor, | 3D Cinzento, | 3D Colorido, | Realístico, | Real,

```

<DEF Virtual Model Generator Subsystem>::=
  Virtual Model Generator Subsystem: ^p
  <VMG system> ^p

<VMG system> ::= <VMG subsystem> | <VMG subsystem> <VMG system>

<VMG subsystem> ::=
  <key> - <title> - <year> ; ^p
  <graphic model> ; ^p
  <other models> ; ^p
  <link> ; ^p
  <description> ; ^p
  <references> . ^p

<other models> ::= Other Models: ^p
  <hearing> ;
  <touch> ;
  <smell> ;
  <flavor>

<hearing> ::= Hearing: Yes | Hearing: No
<touch> ::= Touch: Yes | Touch: No
<smell> ::= Smell: Yes | Smell: No
<flavor> ::= Flavor: Yes | Flavor: No

```

Figura 7: Caracterização do Sistema Gerador de Modelos Virtuais

```

<graphic model> ::=
  GraphicModel: <yn> ^p
  <graphic quality> ;
  <extracted from real world> ;
  <knowledge about the real world> ;
  <dynamics> ;
  <interactivity> ;
  <technology>

<graphic quality> ::= 2D Gray Wireframe | 2D Color Wireframe | 2D Photo
Realistic | 3D Gray Wireframe | 3D Color Wireframe | Realistic | Real

<extracted from real world> ::= Extracted from Real World: True | Extracted
from Real World: False

<knowledge about real world> ::= No knowledge about real world |
Segmented | Segmented and Labeled

<dynamics> ::= Dynamics: True | Dynamics: False
<interactivity> ::= Interactive: True | Interactive: False
<technology> ::=
  Hardware Platform: (<plain text>) ; ^p
  Operating System: (<plain text>) ; ^p
  Graphic Language: (<plain text>) ; ^p

```

Figura 8: Caracterização do Modelo Gráfico.

2. Extraído do Mundo Real: O modelo virtual pode ter como base informação sobre objectos reais ou não. Em algumas aplicações de RA para a cirurgia guiada por imagem, o modelo virtual é construído tendo como base elementos do mundo real [Braz, 2000].
3. Conhecimento Sobre o Mundo Real: coincide completamente com a descrição apresentada em [Milgram et Al., 1994] para Extent of world knowledge. Foram identificadas três categorias de sistemas: Nenhum Conhecimento, | Segmentado, | Segmentado e Etiquetado.
4. Dinamismo: O Modelo Virtual é ou não dinâmico, isto é, modifica-se ou não ao longo do tempo, independentemente da interacção com o utilizador.
5. Interactividade: O modelo virtual reage ou não à interacção com o utilizador

6. Tecnologia: as características relacionadas com a tecnologia usada abarcam os conceitos de Plataforma de Hardware, Sistema Operativo e Linguagem Gráfica.

3.6 Sistema Misturador de Realidades

Para a caracterização do Sistema Misturador de Realidades (Figura 9) interessa-nos em especial a caracterização da mistura de Realidades Visuais.

```

<DEF Mixing Realities Subsystem> ::= Mixing Realities Subsystem ^p
  <MR system> ^p

<MR system> ::= <MR subsystem> | <MR subsystem> <MR system>

<MR subsystem> ::=
  <key> - <title> - <year> ; ^p
  <visual realities> ; ^p
  <other models> ; ^p
  <link> ; ^p
  <description> ; ^p
  <references> . ^p

```

Figure 9: Caracterização do Sistema de Misturador de Realidades

Realidades Visuais

[Azuma, 1997] identifica duas tecnologias de mistura de imagem: óptica e vídeo, e caracteriza-as em termos das vantagens e desvantagens de cada uma delas. Para a presente caracterização segue-se de perto a discussão apresentada por Azuma, alterando o termo vídeo para electrónico querendo com esta alteração realçar o carácter da tecnologia que processa os canais de imagem. De salientar que os dois tipos de tecnologias implicam a especificação de características radicalmente diferentes.

```

<visual realities> ::= Visual Realities Mixing Subsystem ^p
  <background image> ; ^p
  <foreground image> ; ^p
  <mixer technology> ; ^p

<mixer technology> ::=
  Electronic: Chromakeying | Electronic: Depth based | <optical mixer>

<optical mixer> ::= Optical Mixer: <characteristics>

<characteristics> ::=
  Reflected Light: <real number> % , <filtered wave length>

<filtered wave length> ::=
  Filtered Wave Length: (<real number> - <real number> 1012Hz)

<background image> ::= Background Image: <type of image>
<foreground image> ::= Foreground Image: <type of image>
<type of image> ::= Computer Generated Imagery | Video | Real World

```

Figura 10: Caracterização do Sistema Misturador de Realidades Visuais.

Assim sendo é necessário caracterizar o Sistema de mistura como sendo:

1. Óptico
2. Electrónico

E em seguida cada uma das duas tecnologias com as características inerentes a cada uma delas, no caso da

tecnologia electrónica, ainda segundo [Azuma, 1997] importa diferenciar se a mistura é feita por:

1. Chroma Keying ou
2. Baseada em Informação sobre Profundidade,

Caso o sistema misturador de imagem seja óptico descrevem-se as principais características como sejam a percentagem de luz proveniente do mundo real que o meio misturador translúcido reflecte e a presença ou não de filtragem de determinados comprimentos de onda [Azuma, 1997].

3.7 Sistema de Apresentação

Considera-se que o Sistema de Apresentação (Figura 11) deve incluir a possibilidade de reproduzir qualquer das realidades misturadas produzidas pelo Sistema Misturador de Realidades. Para o presente trabalho é relevante apenas o sistema de visualização.

```
<DEF Display Subsystem> ::= Display Subsystem: ^p <D system> ^p
<D system> ::= <D subsystem> | <D subsystem> <D system>
<D subsystem> ::=
  <key> - <name> - <year> ; ^p
  <visual display> ; ^p
  <other models> ; ^p
  <link> ; ^p
  <description> ; ^p
  <references> . ^p
```

Figura 11: Caracterização do Sistema de Apresentação.

3.8 Sistema de Visualização

A caracterização de um sistema de visualização (Figura 12) faz-se de acordo com os 5 conceitos de que em seguida se apresentam os respectivos valores semânticos.

```
<visual display> ::= Visual Display Subsystem ^p
  <display system technology> ; ^p
  <users view point> ; ^p
  <scale> ; ^p
  <color> ; ^p
  <resolution> ; ^p
  <refresh rate> ; ^p
<display system technology> ::= Optical see-through HMD | Video see-through HMD | Both Optical & Video see-through HMD | Projector to the Real World | Projector to a Special Device | Large Screen | Desktop Monitor
<users view point> ::= Egocentric | Exocentric
<scale> ::= Orthoscopic | Nonorthoscopic
<color> ::= Color: <type of color>
<type of color> ::= 1 bit | 8 bits | 16 bits | 24/32 bits
<resolution> ::= Resolution: <integer> X <integer>
<refresh rate> ::= Refresh Rate: <integer> Hz
```

Figura 12: Caracterização do Sistema de Apresentação Óptico.

1. Tecnologia: A tecnologia de visualização usada pode basear-se num dos quatro dispositivos listados: Optical see-through HMD, Video see-through HMD, Monitor, Projector.

2. Ponto de Vista do Utilizador: De acordo com [Milgram et Al., 1994] os dispositivos de visualização dividem-se nos que colocam o utilizador no centro do mundo, ou egocêntricos, de que são exemplos qualquer tipo de HMD e alguns tipos de sistemas baseados em projectores, e naqueles que colocam o utilizador fora do mundo observado, ou exocêntricos, de que um exemplo típico é o vulgar ecrã de computador.
3. Escala: De acordo com [Milgram et Al., 1994] os sistemas de visualização podem dividir-se entre os que permitem representar o mundo observado num factor de escala 1:1, ou ortoscópicos, e aqueles que não o permitem.
4. Cor: Quantidade de cores passíveis de reproduzir com o dispositivo de apresentação em número de bits necessários para as representar: 1 bit | 8 bits | 16 bits | 24/32 bits.
5. Frequência de refrescamento: Quantidade de quadros apresentados por segundo.

3.9 Sistema Real de Manipulação de Objectos

Considerando que a tecnologia envolvida é determinante para definir as características do sistema manipulador, e que estas são radicalmente diferentes para cada tipo de tecnologia, limitamo-nos a listar os tipos de tecnologias passíveis de utilizar para interagir hapticamente com o mundo real e virtual.

```
<DEF Real Manipulator Subsystem> ::=
  Real Manipulator Subsystem: ^p
  <RM system> ^p
<RM system> ::= <RM subsystem> | <RM subsystem> <RM system>
<RM subsystem> ::=
  <key> - <name> - <year> ; ^p
  <technology> ; ^p
  <link> ; ^p
  <description> ; ^p
  <references> . ^p
  <technology> ::= Virtual Reality Gloves | Robot | Robot Arm |
    Pointer | <other>
<other> ::= <plain text>
```

Figura 13: Caracterização do Sistema Real de Manipulação de Objectos

3.10 Sistema de Seguimento

De acordo com [Azuma, 1997] o maior dos obstáculos que, por si só, impede a construção de sistemas de RA eficazes, reduz-se aos requisitos quanto a sistemas de seguimento que reportem a localização do utilizador e dos objectos do ambiente real com elevada precisão e longo raio de acção. Aí, aponta-se como sendo o mais provável dos caminhos evolutivos o recurso a sistemas de seguimento complexos que recorram a diferentes tecnologias, por forma a extrair de cada uma as principais potencialidades e eliminar as desvantagens, caminho este que, de facto, de 1997 para cá se tem verificado [Livingston et Al., 1997] [Azuma et Al., 1999] [Auer et Al., 1999], a par, no entanto, de uma

clara preponderância no recurso a técnicas de visão computacional para o seguimento de objectos [Klinker et Al., 1999] [Dorfmueller, 1999] [Seibert, 1999].

A procura de uma solução para este problema tem sido uma das mais profícuas áreas de investigação da RA, e senão note-se a quantidade de artigos a ela dedicada. De tudo isto decorre que um dos subsistemas que de facto pode ser considerado como um dos mais relevantes num sistema de RA é também um dos mais difíceis de caracterizar. Desde logo porque a grande maioria recorre a sistemas de seguimento complexos, que incorporam diferentes tecnologias e metodologias, mas além disso, muito trabalho existente sobre dispositivos de seguimento tem sido desenvolvido pelo campo da biomecânica, onde os critérios classificativos divergem claramente dos critérios usados nos trabalhos sobre Realidade Virtual e Realidade Aumentada.

Para a caracterização dos sistemas de seguimento baseamo-nos principalmente em três trabalhos: [Rolland et Al, 2000], os Capítulos 4 e 5 de [Durlach, 1995], e [Mulder, 1994] um relatório técnico sobre dispositivos de seguimento de movimento humano da área da biomecânica, cujas últimas actualizações na WEB datam de 1998.

Os parâmetros usados para caracterização dos sistemas de seguimento são comuns aos três subsistemas potencialmente existentes num sistema de RA e identificados na Figura 2, pelo que são aqui apresentados apenas uma vez sob o título genérico de Sistemas de Seguimento, mas tendo em consideração que, quando da classificação de um sistema de RA teremos sempre de identificar o objecto seguido (<Tracked Object> na sintaxe da Figura 14), nomeadamente:

1. Sistema de Captação de Imagem ou SCI,
2. Objectos Reais que se movimentem na cena,
3. Sistema Real de Manipulação de Objectos.

As características identificadas como relevantes para a classificação de sistemas de seguimento são as seguintes:

1. Complexidade: Um Sistema de Seguimento dir-se-á Elementar quando envolver apenas uma das tecnologias citadas como elementares em [Durlach et Al., 1995]. Em contraponto considerar-se-á um sistema de seguimento Complexo quando envolver o recurso a mais de uma das tecnologias ou metodologias consideradas elementares. A classificação dos dispositivos elementares de seguimento segue de perto a proposta em [Durlach et Al., 1995]. Note-se que [Mulder, 1994] propõem uma classificação completamente diferente ao sistematizar tecnologias de seguimento do movimento humano, no entanto, ao nível das tecnologias elementares descritas, todos os dispositivos por ele citados se conseguem enquadrar nos tipos de tecnologias apresentados em [Durlach et Al., 1995]. Um único ponto merece no entanto referência, a relevância dada em [Mulder, 1994] aos

dispositivos do tipo *force ball* e Multibotões aos quais aqui não fazemos referência explícita optando pela sua inclusão nos dispositivos mecânicos com ligação à terra.

2. Tipo: Distinguem-se aqui três grandes categorias enunciadas em [Moeslund et Al., 2001]: *Passive Tracking*, *Tracking Based on Markers*, *Active Tracking*.

```

< DEF Tracking Subsystem > ::=
    Tracking Subsystem for < tracked object > ^p < T system > ^p
< T system > ::= < T subsystem > | < T subsystem > < T system >
< T subsystem > ::=
    < key > - < title > - < year > ; ^p
    < technology >
< tracked object > ::=
    Image Acquisition Subsystem | Real Objects | Real Manipulator
    Subsystem
< technology > ::=
    Passive Tracking: ^p < technical characteristics > |
    Tracking Based on Markers: ^p < technical characteristics > |
    Active Tracking: < active tracker type >; ^p < technical characteristics >
< active tracker type > ::= Goniometer | Earth Grounded | Magnetic | Acoustic
| Inertial | Eye Tracker | Structured Light System | Radar | Laser | <Other>
< technical characteristics > ::= < tracked data > ;
    < resolution > ;
    < sampling rate > ;
    < latency > ;
    < range > ;
    < workspace > ;
    < cost > ;
    < quantity of simultaneous measurements > ;
    < susceptibility to obscuration > ;
    < ease of calibration > ;
    < technical complexity > ;
    < convenience > ; ^p
    < link > ; ^p
    < description >; ^p
    < references > . ^p
< tracked data > ::= Motion | 2D Position | 3D Position | 2D Orientation |
    2D Position and Orientation | 3D Position and Orientation
< resolution > ::= < maximal positional error > | < maximal angular error > |
    < maximal positional error > < maximal angular error >
< maximal positional error > ::= Maximal Positional Error:
    ( < real > mm X - < real > mm Y - < real > mm Z )
< maximal angular error > ::= Maximal Angular Error:
    ( < angle > XY - < angle > YZ - < angle > ZX )
< sampling rate > ::= Sampling Rate: < real > Hz
< range > ::= Range: < real > m
< latency > ::= Latency: < real > ms
< workspace > ::= Room | Rooms | Modular | Outdoors
< quantity of simultaneous measurements > ::=
    Quantity of Simultaneous Measurements: < integer > |
    Quantity of Simultaneous Measurements: undetermined
< susceptibility to obscuration > ::= Susceptibility to Obscuration: < yn >
< technical complexity > ::= Technical Complexity: < degree >
< degree > ::= 1 | 2 | 3 | 4
< convenience > ::= Convenience: < degree >
< ease of calibration > ::= Ease of Calibration: < degree >
< cost > ::= Price: < integer > * 10<integer>€

```

Figura 14: Caracterização do Sistema de Seguimento

3. Tipo de seguidor activo: Seguindo [Durlach et Al., 1995] enunciam-se 9 tipos de seguidores activos: *Goniometer, Earth Grounded, Magnetic, Acoustic, Inertial, Eye Tracker, Structured Light System, Radar, Laser*.
4. Características Técnicas: independentemente do tipo de seguidor foram identificadas 12 características relevantes para a classificação de sistemas de seguimento. As características técnicas coincidem quase completamente com as propostas em [Durlach et Al., 1995]. Note-se que a definição das três últimas características : <Facilidade de calibração>, <Complexidade Técnica> ou <Conveniência> são de aferição bastante subjectiva. Optou-se no entanto pela sua manutenção a fim de permitir uma maior correspondência com as descrições de [Durlach et Al., 1995]. Em seguida apresenta-se o significado de cada uma destas características:
 - a. Dados Estáticos Rastreados: De acordo com a literatura, nomeadamente [Mulder, 1994], mas também com os diferentes sistemas de RA analisados, os dispositivos de seguimento podem captar apenas ou a posição ou a orientação dos objectos seguidos, caso em que para um sistema de RA se torna necessário combinar mais de um sistema, ou ainda permitir captar simultaneamente as posição e orientação.
 - b. Resolução: Entenda-se por Resolução o conjunto dos erros de posição e orientação máximos do sistema de seguimento.
 - c. Erro Posicional Máximo: O conceito de erro posicional máximo é definido como sendo a diferença em milímetros entre a posição de facto do objecto seguido e a posição reportada pelo dispositivo de seguimento em qualquer das três dimensões espaciais. Considerando que alguns dos dispositivos apresentam erros máximos de posição diferentes consoante se trate da profundidade e da largura ou altura, opta-se aqui por especificar o erro máximo para cada uma das dimensões.
 - d. Erro Angular Máximo: O conceito de erro máximo de orientação é definido como sendo a diferença em graus entre a orientação de facto do objecto seguido e a orientação reportada pelo dispositivo de seguimento em redor de qualquer dos três eixos de rotação de um sistema Cartesiano. Considerando que alguns dos dispositivos apresentam erros máximos diferentes consoante o plano em que é medido o angulo, opta-se aqui por especificar o erro máximo no plano perpendicular a cada um dos eixos de um sistema Cartesiano.
 - e. Taxa de Amostragem: Numero de amostras fornecidas por segundo.
 - f. Alcance: Distância máxima, em metros, a que o sistema de seguimento permite obter dados fidedignos.
 - g. Espaço de Trabalho: Caracteriza o Sistema de Seguimento em termos do tipo de área em que pode ser usado. Um sistema modular é aquele que está, à partida, preparado para ser ampliado de forma

modular à medida das necessidades, sem que no entanto permita um funcionamento em espaços abertos.

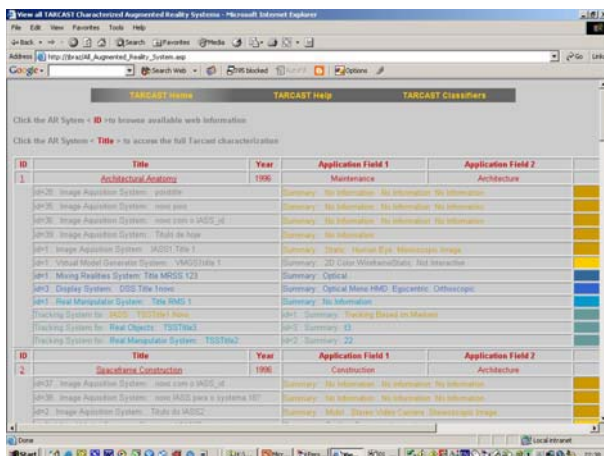
- h. Custo: Ordem de grandeza do preço do sistema de seguimento
- i. Susceptibilidade à Ocultação: sim ou não
- j. Quantidade de medições Simultâneas: Medição Única | Medições Múltiplas | Indeterminado.
- k. Facilidade de calibração: Considerando o elevado grau de subjectividade presente no julgamento sobre a complexidade técnica de um sistema de seguimento, optou-se por impor uma escolha reduzida, apenas quatro graus de complexidade, e forçar uma opção mais positiva ou mais negativa como único critério orientador oferecendo um número par de escolhas. Quanto maior mais fácil é a calibração.
- l. Complexidade Técnica: Considerando o elevado grau de subjectividade presente no julgamento sobre a complexidade técnica de um sistema de seguimento, optou-se por impor uma escolha reduzida, apenas quatro graus de complexidade, e forçar uma opção mais positiva ou mais negativa como único critério orientador oferecendo um número par de escolhas. Quanto maior mais complexo é o sistema.
- m. Conveniência: Considerando o elevado grau de subjectividade presente no julgamento sobre a conveniência de um sistema de seguimento, optou-se por impor uma escolha reduzida, apenas quatro graus de conveniência, e forçar uma opção mais positiva ou mais negativa como único critério orientador oferecendo um número par de escolhas. Quanto maior o valor mais conveniente é o sistema.

4 TARCAST: APLICAÇÃO DE SUPORTE À CARACTERIZAÇÃO

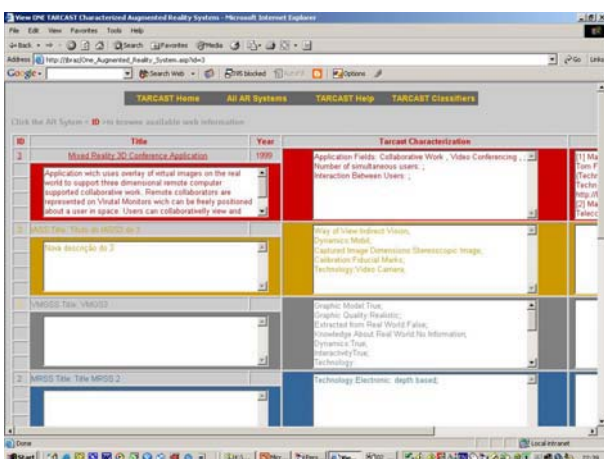
Para suporte ao processo de caracterização dos sistemas de RA com recurso à taxonomia descrita no ponto 3, foi desenvolvida uma aplicação, também denominada TARCAST (Figura 15), a que se pode aceder através de <http://ltodi.est.ips.pt/jbraz/Tarcast/index.htm>, sendo actualmente possível:

1. Ver uma resumida descrição da caracterização de todos os sistemas analisados em http://ltodi.est.ips.pt/jbraz/Tarcast/All_Augmented_Reality_System.asp
2. Ver a caracterização completa de cada sistema premindo o nome do sistema requerido na página indicada em 1).
3. Caracterizar novos sistemas de RA em <http://ltodi.est.ips.pt/jbraz/Tarcast/registration.htm>
4. Obter uma *password* e um nome de utilizador que permita a qualquer interessado caracterizar novos sistemas de RA e registá-los no TARCAST.
5. Alterar os dados relativos à caracterização de um dado sistema entrando com o nome de utilizador

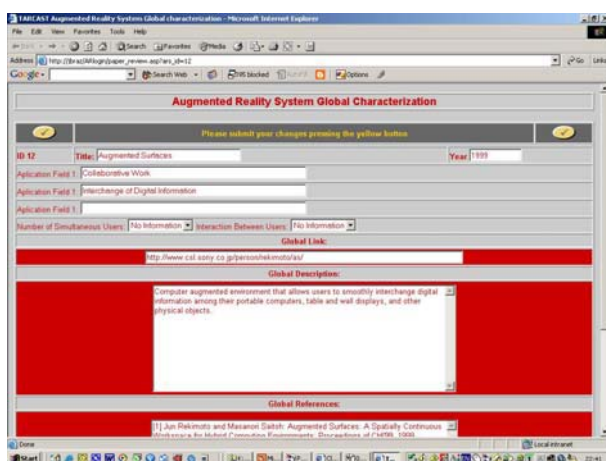
correspondente ao sistema a alterar e respectiva password.



a)



b)



c)

Figura 15: Interface da aplicação de suporte ao processo de caracterização. a) Listagem de todos os sistemas caracterizados. b) Listagem completa de um dos sistemas. c) Página inicial para caracterização de um sistema.

5 VANTAGENS

Algumas das potenciais vantagens da presente taxonomia são:

1. Pode vir a permitir obter uma classificação para os sistemas de RA de acordo com a taxonomia proposta por Paul Milgram para os Sistemas de Apresentação de Realidade Misturada de forma automática, bastando para tal especificar um sistema de ponderação para cada característica técnica.
2. Pode vir a permitir classificar os modelos virtuais de acordo com a taxonomia proposta por Zeltzer para os sistemas de realidade Virtual de forma automática, seguindo o mesmo princípio enunciado em 1).
3. Realça o facto de algumas das características necessárias para a classificação segundo a taxonomia de Milgram estarem distribuídas pelos diferentes constituintes de um sistema de RA, e desta forma:
4. Permite endereçar de forma mais clara os requisitos globais de um sistema de RA para o subsistema a que devem ser exigidos e permite endereçar os problemas com que se defronta a RA aos subsistemas mais intimamente relacionados com cada um deles.
5. Permite uma procura das características técnicas mais apropriadas ao desenvolvimento de futuros sistemas de RA com base nas tecnologias e metodologias actualmente usadas.

6 CONCLUSÕES

O desenvolvimento da taxonomia para Sistemas de RA possibilitou um estudo estruturado e comparativo de diferentes sistemas actualmente em funcionamento, mas, além disso, tornou claro que, num contexto em que a RA já está a sair dos laboratórios [Azuma et Al., 2001], facilita a escolha das tecnologias e metodologias que melhor se adaptem às necessidades de cada sistema de RA podendo vir a contribuir para transformar uma área ainda claramente de investigação e em que cada sistema é construído de raiz, numa área de desenvolvimento em que, com base em diferentes soluções para diferentes elementos constitutivos de um sistema de RA (no sentido da figura 2) seja possível desenvolver novos sistemas de forma modular, escolhendo “o melhor de cada um”.

A classificação de diferentes sistemas levou-nos a concluir que o processo de caracterização pode ser simplificado eliminando alguma da redundância ainda existente. Como exemplo disso, características como a ego/exocentricidade ou a ortoscopia parecem poder ser extraídas das tecnologias de visualização deixando assim de figurar de forma explícita, antes podendo ser “inferidas” de outras características classificativas.

7 TRABALHO FUTURO

De acordo com a última das conclusões acima enunciadas, é necessário eliminar a redundância ainda existente ao nível das características usadas para classificar cada sistema.

Tratando-se de uma taxonomia de sistemas com elevado nível de complexidade as futuras contribuições e evoluções em cada área deverão ser incorporadas, sob pena de a taxonomia se tornar obsoleta a curto prazo.

A possibilidade de inferir as classificação propostas em [Milgram et Al., 1994], para sistemas de visualização, e em [Zeltzer, 1992] para sistemas de realidade virtual, a partir da caracterização obtida com o TARCAST é um objectivo a atingir. Para o conseguir é no entanto necessário coligir uma quantidade significativa de sistemas classificados.

A possibilidade de alargar o tipo de consultas por procura e ordenação, se bem que possibilitado pela estrutura da base de dados subjacente ainda não foi implementada através da interface de utilização.

8 BIBLIOGRAFIA

- [Auer et Al., 1999] Thomas Auer, Axel Pinz: *The integration of optical and magnetic tracking for multi-user augmented reality*; Computer & Graphics, N°23, pp 805-808, Pergamon, 1999.
- [Azuma, 1997] Ronald T. Azuma: A Survey of Augmented Reality, in Presence: Teleoperators and Virtual Environments, Vol. 6, n°4, pp. 355-385, 1997.
<http://citeseer.nj.nec.com/azuma95survey.html>.
- [Azuma et Al., 1999] R. T. Azuma, J. W. Lee, B. Jiang, J. Park, S. You, U. Neumann: *Tracking in unprepared environments for augmented reality systems*, Computer & Graphics, N°23, pp 787-793, Pergamon, 1999.
- [Azuma et Al., 2001] R. Azuma, Y. Bailiot, R. Behringer, S. Feiner, S. Julier, B. MacIntyre: Recent Advances in Augmented Reality. IEEE Computer Graphics and Applications 21, 6 (Nov/Dec 2001), 34-47.
<http://www.cs.unc.edu/~azuma/cga2001.pdf>.
- [Billinghurst et Al., 1999] Mark Billinghurst, Hirokazu Kato: Collaborative Mixed Reality, in Proceedings of International Symposium on Mixed Reality (ISMR '99). Mixed Reality--Merging Real and Virtual Worlds, pp. 261-284, 1999.
<http://hitl.washington.edu/publications/r-98-36/>.
- [Braz, 2000] José Braz: *Realidade Aumentada em Engenharia Biomédica: Estado da Arte*, in Actas da 5ª Workshop em Engenharia Biomédica, Instituto Superior Técnico, Lisboa, 2000.
http://todi.est.ips.pt/jbraz/ficheiros/pubs/2000_5web_jbraz.pdf
- [Braz, 2005] José Braz, João Pereira, António Veloso: "VIDA – Interactive Viewer of Augmented (biomechanical) Data". in VIRTual (ISSN: 0873-1837)- special edition: "Advances in Computer Graphics in Portugal". Janeiro 2005.
- [Dorfmueller, 1999] Klaus Dorfmueller: *Robust tracking for augmented reality using retroreflective markers*, Computer & Graphics, N°23, pp 795-800, Pergamon, 1999.
- [Dubois et Al., 1999a] Emmanuel Dubois, Laurence Nigay, Jocelyne Troccaz, Olivier Chavanon, Lionel Carrat: Classification Space for Augmented Surgery, an Augmented Reality Case Study, in Proceedings of INTERACT'99, pp. 353-359, 1999.
<http://citeseer.nj.nec.com/dubois99classification.html>.
- [Durlach et Al., 1995] Nathaniel I. Durlach, Anne S. Mavor (Editors): Virtual Reality: Scientific and Technological Challenges, Report of the Committee on Virtual Reality Research and Development to the National Research Council, National Academy Press, pp. 161-204, 1995, ISBN: 0-309-05135-5.
- [Klinker et Al., 1999] G. Klinker, D. Stricker, D. Reinert: *Optically based direct manipulation for augmented reality*; Computer & Graphics, N°23, pp 827-830, Pergamon, 1999.
- [Kutulakos et Al., 1998] Kiriakos N. Kutulakos, James R. Vallino: Calibration Free Augmented Reality, in IEEE Transactions on Visualization and Computer Graphics, Vol 4, n°1, January March, 1998.
<http://www.cs.rochester.edu/u/kyros/site/research/augment/augment.html>
- [Livingston et Al., 1997] Mark A. Livingston, Andrei State: *Magnetic Tracker calibration for Improved Augmented Reality Registration*; Presence, Vol 6, N°5, pp 532-546, October 1997.
- [Milgram et Al., 1994] Paul Milgram, Fumio Kishino: A Taxonomy of Mixed Reality Visual Displays, in IEICE Transactions on Information Systems, Vol E77-D, N°12, December, 1994.
http://vered.rose.utoronto.ca/people/paul_dir/IEICE94/ieice.html
- [Moeslund et Al., 2001] Thomas B. Moeslund, Erik Granum: A Survey of Computer Vision-Based Human Motion Capture,. In Int. Journal of Computer Vision and Image Understanding, No. 3, Volume 81, 2001.
<http://citeseer.nj.nec.com/moeslund99computer.html>
- [Mulder, 1994] Axel Mulder: Human movement tracking technology, Hand Centered Studies of Human Movement project, technical report 94-1, 1994.
<http://www.cs.sfu.ca/~amulder/personal/vmi/HMTT.pub.html>.
- [Naimark, 1991] Michael Naimark: Elements of Realspace Imaging: a Proposed Taxonomy, in Proceedings of First Moscow International Workshop on Human-Computer Interaction, Moscow, 1991.
<http://www.naimark.net/writing/realspace.html>
- [Nakajima et Al., 1998] S. Nakajima, R. Kikinis, P. M. Black, H. Atsumi, M. E. Leventon, N. Hata, D. C. Metcalf, T. M. Moriarty, E. Alexander, F. A. Jolesz: Image-Guided Neurosurgery at Brigham and Women's Hospital.
http://splweb.bwh.harvard.edu:8000/pages/papers/shin/book_chap/manus.html
- [Rekimoto et Al., 1999] Jun Rekimoto and Masanori Saitoh: Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments; Proceedings of CHI'99, 1999.
<http://www.csl.sony.co.jp/person/rekimoto/papers/chi99.pdf>
- [Rolland et Al., 2000] J.P. Rolland, L. Davis, Y. Bailiot: *A Survey of Tracking Technology for Virtual Environments*, in Augmented Reality and Wearable Computers, Ch. 3, Ed. Barfield and Caudell, Mahwah, NJ., 2000.
http://www.ait.nrl.navy.mil/people/bailiot/publications/Tracking_survey/tracking-chapter.PDF
- [Seibert, 1999] Frank Seibert: *Augmented reality using uncalibrated optical tracking*; Computer & Graphics, N°23, pp 801-804, Pergamon, 1999.
- [Seo et Al., 2000] Yongduek Seo, Ki Sang Hong: Calibration-Free Augmented Reality in Perspective, in IEEE Transactions on Visualization and Computer Graphics, Vol. 6, N°4, October-December, 2000.
- [Vallino, 2001] Jim Vallino: Introduction to Augmented Reality; 2001.
<http://www.se.rit.edu/~jrv/research/ar/introduction.html>.
- [Zeltzer, 1992] David Zeltzer: Autonomy, Interaction and Presence; Presence: Teleoperators and Virtual Environments, Vol1, N°1, Winter 1992.

Avaliando a Veracidade das Histórias sobre Documentos

Daniel Gonçalves Joaquim A Jorge

Dep. Eng^a. Informática, IST

Av. Rovisco Pais, 1000 Lisboa

daniel.goncalves@inesc-id.pt, jaj@inesc-id.pt

Resumo

Um problema cada vez mais encontrado pelos utilizadores de computadores é localizar um determinado documento electrónico. As formas tradicionais para organizar os documentos são cada vez menos eficazes ao lidar com a quantidade de informação que até mesmo um utilizador comum tem ao seu dispor hoje em dia. Adicionalmente, são soluções artificiais, obrigando à classificação dos documentos em categorias arbitrárias no sistema de ficheiros, com todos os problemas que isso implica. É urgente o desenvolvimento de novos mecanismos de recuperação de documentos que reflectam a forma de pensar dos utilizadores quando se recordam dos seus documentos, recorrendo a informação autobiográfica com eles relacionada.

Na nossa investigação, verificámos que as histórias sobre os documentos são uma forma simples e natural dos utilizadores se lhes referirem. Um conjunto de entrevistas permitiu-nos conhecer qual a forma e conteúdos típicos dessas histórias. Através da avaliação de protótipos de baixa fidelidade, construídos com base nas características identificadas, identificámos a forma que uma interface para a recuperação de documentos baseada em narrativas pode tomar.

Restava, no entanto, responder a uma questão crucial: são as histórias suficientemente precisas na descrição dos documentos? Neste artigo descrevemos um estudo em que uma versão funcional de um protótipo de interface para capturar histórias sobre documentos foi usada para recolher trinta histórias sobre documentos reais. A informação nessas histórias foi então comparada com os documentos propriamente ditos. Verificámos que, no geral, as histórias descrevem os documentos de forma fidedigna (81% a 91% de toda a informação é correcta), validando as histórias como forma de recuperação de documentos. Confirmámos, também, que os utilizadores conseguem contar histórias usando a interface: as histórias contadas ao computador são semelhantes às contadas a humanos.

Palavras-Chave

Narrativas, recuperação de documentos, gestão de informação pessoal, interfaces pessoa-máquina

1. INTRODUÇÃO

Duas tarefas que nenhum utilizador de computadores consegue evitar são organizar e posteriormente encontrar os seus documentos electrónicos. Infelizmente, os mecanismos existentes para as realizar não sofreram alterações significativas nas últimas décadas. Baseiam-se, primordialmente, na classificação de *todos* os documentos criados numa hierarquia definida pelo utilizador, no sistema de ficheiros, e na atribuição de um nome ao ficheiro que representa o documento dentro dessa hierarquia. Isto causa inúmeros problemas, uma vez que não é invulgar um documento poder ser classificado em mais do que uma categoria, ou não parecer enquadrar-se em nenhuma das existentes. As decisões a que esse tipo de classificação obriga aumentam desnecessariamente a carga cognitiva a que os utilizadores estão sujeitos e dificulta a recuperação dos documentos num momento posterior.

Thomas Malone efectuou um dos primeiros estudos sobre as formas de organizar documentos [Malone83]. Neste, a

dificuldade dos utilizadores em classificar todos os documentos ficou bem patente. Era preferível para a maioria dos utilizadores organizar os documentos em pilhas e recorrer à sua memória visual e espacial do que ter que os classificar. Hoje em dia, o problema subsiste, agravado pelo aumento significativo do número de documentos com que os utilizadores lidam.

A causa da ineficiência encontrada ao usar sistemas de ficheiros hierárquicos é estes não reflectirem a forma mais natural dos utilizadores se referirem aos seus documentos. Quando um documento é procurado, não nos lembramos do um nome ou classificação arbitrariamente atribuídos, mas sim do documento em si, de porque o necessitamos nesse momento, o que continha, etc. Em suma, recorremos a informação autobiográfica quando nos recordamos de um documento. Isto foi confirmado por estudos das caixas de correio electrónico dos utilizadores, em que o mesmo problema de classificação existe ao guardar mensagens nas pastas hierarquicamente organizadas. Verificou-se, no entanto,

que muitos utilizadores recorriam a essas mensagens como forma de encontrar documentos! [Whittaker96] Isto deve-se ao facto das mensagens, ao contrário dos documentos no sistema de ficheiros, aparecerem inseridas num contexto que permite mais facilmente encontrá-las, associadas a informação adicional (data e hora, remetente, assunto, etc.) que está mais perto da forma natural de recordar dos utilizadores.

Numa tentativa de permitir aos utilizadores fazer uso dessa informação para recuperar os seus documentos, vários trabalhos foram desenvolvidos. Alguns seguiram uma abordagem mais limitada dando um papel preponderante ao tempo. É o caso do Lifestreams [Freeman96], em que todos os documentos são apresentados sequencialmente, de forma ordenada, sendo possível obter *substreams* de documentos que cumprem um certo critério filtrado a *stream* principal. Já o Timescape [Rekimoto99] apresenta o *desktop* do ambiente de trabalho como uma janela sobre um determinado instante de tempo, que pode ser movido quer para o passado quer para o futuro.

Mais abrangentes, algumas abordagens têm em conta um leque mais alargado de propriedades. O primeiro desses trabalhos foi o Semantic File System [Gifford91], em que os documentos estão organizados em directorios virtuais, resultado de pesquisas segundo critérios como o seu autor ou a data em que foi criado. Mais recentemente, sistemas semelhantes mas mais sofisticados foram desenvolvidos. É o caso dos Placeless Documents de Paul Dourish [Dourish00] e do PACO, de Ricardo Baeza-Yates [Baeza-Yates96], em que os documentos são organizados em colecções que os agrupam por algum critério.

Alguns sistemas tentam especificamente fazer uso de propriedades cujos valores não conseguem ser apreendidos directamente dos documentos propriamente ditos. É o caso do Stuff-I've-Seen [Dumais03] que integra informação sobre os documentos recolhida de várias aplicações que o utilizador corre. O sistema de Ariel Shamir [Shamir04] tem um comportamento semelhante. Os recentes sistemas de pesquisa no *desktop*, como o Google Desktop, também partilham desta característica.

Estes sistemas, embora promissores, não resolvem totalmente o problema de recuperar documentos. Recordar valores atribuídos a propriedades arbitrárias não é muito melhor do que recordar onde, no sistema de ficheiros, está um documento guardado. Falta uma interface que permita ao utilizador, de forma natural, referir toda a informação relevante. Propomos as histórias sobre os documentos como base para essa interface.

Os seres humanos são contadores de histórias por natureza. Todos contamos histórias da infância à velhice. Em particular, contamos histórias sobre os documentos que procuramos quando temos alguém ao nosso lado durante essa pesquisa. Uma interface capaz de fazer uso dessas histórias para recuperar os documentos seria, ao mesmo tempo, natural para os utilizadores, e faria uso da

informação realmente associada pelos utilizadores aos seus documentos.

Na tentativa de criar interfaces baseadas em histórias para recuperação de documentos, conseguimos, recorrendo a entrevistas, recolher sessenta histórias. A sua análise permitiu-nos conhecer a forma e conteúdos que podemos esperar. Um conjunto de *guidelines* para a criação das interfaces foi, também, desenvolvido [Gonçalves04]. Com base nessas *guidelines*, e com a ajuda da avaliação de protótipos de baixa fidelidade, foi possível identificar um formato promissor para a interface propriamente dita [Gonçalves04a].

No entanto, duas questões cruciais para a validação das histórias como base de uma interface para recuperação de documentos ainda estavam por responder. Em primeiro lugar, era necessário confirmar que um utilizador consegue contar histórias a um computador da mesma forma que o faz a um interlocutor humano. Em segundo lugar, era preciso saber até que ponto as histórias contadas são fidedignas. Relembrar correcta e precisamente alguma informação pode não ser fácil, principalmente quando tentando descrever documentos manipulados há já algum tempo. Se a informação contida nas histórias estiver muito desfasada da realidade, não poderá ser usada para encontrar documentos.

Para dar resposta a estas questões, implementámos a interface que se mostrou mais adequada nos protótipos de baixa fidelidade. Trinta novas histórias sobre documentos foram recolhidas, tendo sido comparadas com a informação real associada aos documentos descritos. Isto permitiu-nos avaliar até que ponto a informação estava correcta. Adicionalmente, as histórias foram comparadas com as previamente recolhidas com o auxílio de um entrevistador humano, de modo a verificar se as histórias contadas a uma interface no computador eram ou não semelhantes às restantes. Verificámos que é esse o caso. Constatámos, também, que entre 81% e 91% de toda a informação contida nas histórias está correcta.

Na secção seguinte descreveremos, sucintamente, a interface implementada. Em seguida, referiremos a metodologia usada no estudo descrito neste artigo, após o qual os principais resultados serão enumerados. Seguir-se-á uma discussão dos mesmos. Concluiremos então, apresentando possíveis caminhos para o trabalho futuro.

2. O PROTÓTIPO

A interface usada no estudo descrito neste artigo foi concebida com base nos resultados de testes com protótipos de baixa fidelidade [Gonçalves04a]. Esses protótipos foram criados com base nas linhas-mestras inferidas a partir da análise de histórias contadas a ouvintes humanos [Gonçalves04]. Comparando-as com as recolhidas usando os protótipos, foi possível identificar qual deles melhor permitia aos utilizadores contarem as suas histórias. A satisfação dos utilizadores foi também medida com a ajuda de um questionário.

Uma vez escolhida a forma que a interface deveria tomar esta foi implementada usando a linguagem de

programação Python, escolhida tanto pelo seu poder e flexibilidade como pela rapidez de construção de um protótipo que permite. A aplicação resultante foi baptizada com o nome de Quill.

2.1 A Interface

A Figura 1 mostra o aspecto final da interface Quill na altura da realização do estudo. A janela está dividida em três áreas. A que mais se destaca, no canto superior esquerdo, é a área da história propriamente dita. Nela, a história que descreve os documentos procurados vai sendo incrementalmente construída. Cada elemento possível vai sendo sucessivamente apresentado ao utilizador sob a forma de uma frase incompleta. Após a introdução da informação relevante pelo utilizador, essa frase será terminada. A parte da frase que contém a informação específica sobre o documento funcionará como um elo hipertextual permitindo a posterior alteração dessa informação. Os vários elementos possíveis foram inferidos a partir da análise de conteúdos efectuada às histórias contadas a humanos num estudo anterior.

A informação é introduzida com a ajuda de diálogos especializados que aparecem no lado direito da janela. A informação a pedida em cada um desses diálogos reflecte directamente o que os conteúdos encontrados nas histórias contadas a entrevistadores humanos. Os diálogos são flexíveis. Por exemplo, se o utilizador mencionar que um documento é do tipo Microsoft Word, então no

diálogo referente ao Conteúdo do documento (em que o aspecto visual do mesmo pode ser especificado) apenas serão dadas opções referentes a possíveis aspectos de documentos desse tipo. Se, caso contrário, se tratasse de uma folha de cálculo, seriam esses os aspectos mostrados.

A história é construída como um todo coerente, legível pelos utilizadores. Assim, quando nova informação é introduzida e as frases completadas, estas podem sofrer pequenas alterações de modo a reflectir essa informação. Pretende-se manter o texto o mais correcto possível (dentro de limites razoáveis) para que a história como um todo continue a ser apercebida como tal pelos utilizadores. Dito isto, tentámos que as alterações fossem tão pequenas quanto possível evitando desconcertar e confundir os utilizadores.

A ordem na qual os vários elementos da história são sugeridos pelo programa é a inferida a partir das histórias contadas a humanos, que considerámos ser a mais natural. Apesar disso, o utilizador tem ao seu dispor um conjunto de botões que lhe permitem alterar o fluxo da história. Estes estão localizados imediatamente abaixo da área da história. O mais à esquerda, “*I Want Another*” (“Quero Outro”) permite aos utilizadores escolherem um qualquer elemento à sua escolha para referir em seguida. Para tal, apresenta uma lista ao utilizador com os elementos possíveis. O botão seguinte, “*It Didn’t Happen*” (“Não Aconteceu”), pode ser pressionado pelos utilizadores

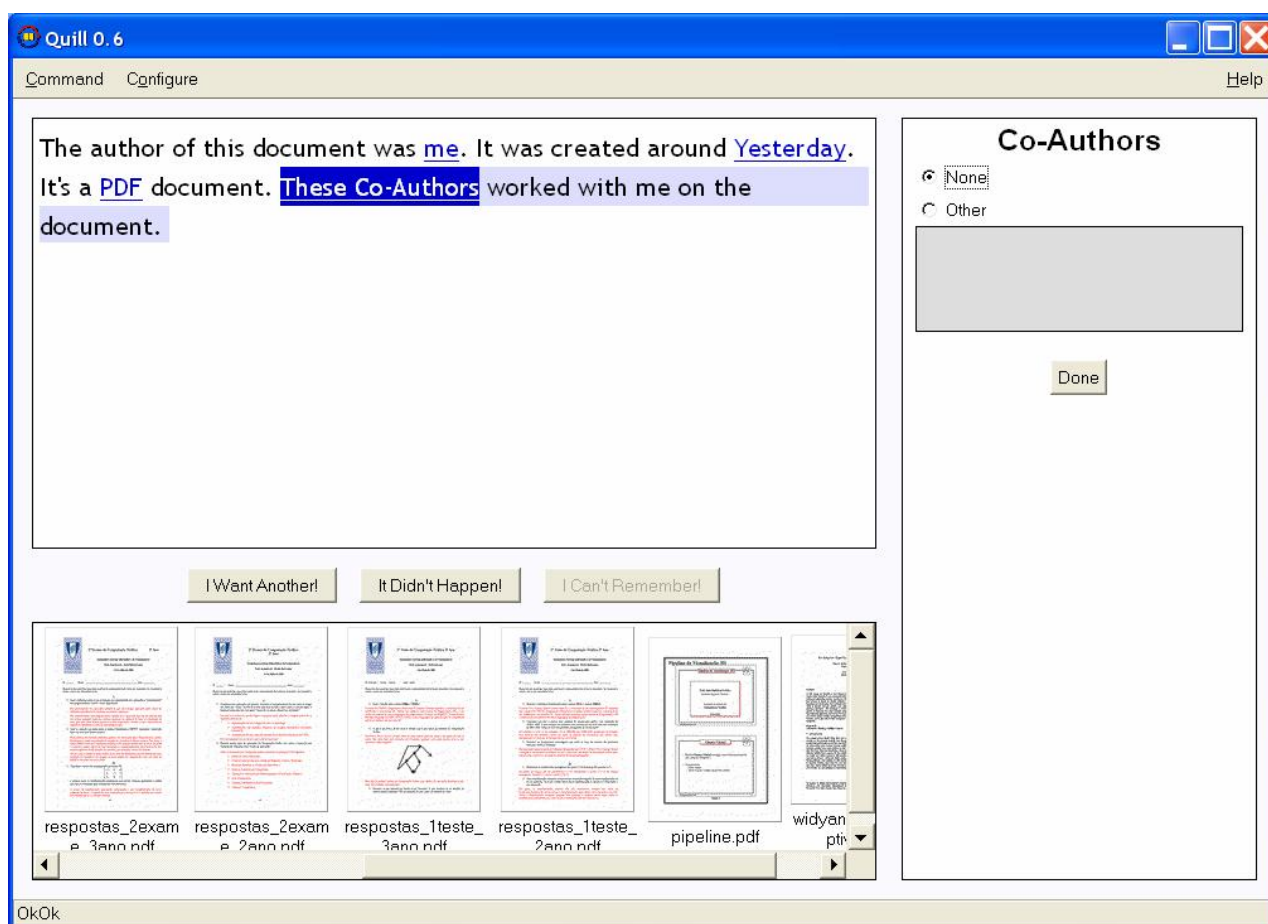


Figura 1 – Aspecto geral da interface

quando o elemento sobre o qual vão falar num dado momento se reporta a algo que não teve lugar. Por exemplo, se questionados quanto ao envio do documento por correio electrónico para outro utilizador, este botão pode ser pressionado caso isso não tenha tido lugar. Finalmente, encontramos o botão “*I Can’t Remember*” (“Não me Lembro”). Este poderá ser usado quando os utilizadores são inquiridos sobre um elemento que não recordam bem. A existência destes dois últimos botões permite ao utilizador estabelecer a diferença entre a informação que não sabe e aquela que sabe não ter acontecido. Isto acontecia recorrentemente nas histórias contadas a humanos pelo que se tentou preservar essa informação na interface.

A terceira área na janela da aplicação é a área de sugestão de documentos. À medida que a história se vai tornando mais completa, o Quill procura continuamente os documentos que mais se adequam à descrição nela contida. O seu nome é mostrado junto a um pequeno *thumbnail* da primeira página do documento (ou o seu aspecto geral, se for uma imagem, por exemplo). Efectivamente, outro dos factores mais vezes referidos nas histórias era o aspecto visual dos documentos, pelo que esperamos, apresentando este *thumbnail*, tomar partido da memória visual dos utilizadores e ajudá-los a reconhecer o documento que procuram de forma natural, sem perturbar indevidamente o processo de contar a história distraíndo-os da mesma.

2.2 A Infraestrutura

Subjacente à interface que acabámos de apresentar está uma infraestrutura que permite ao sistema compreender as histórias contadas pelos utilizadores e encontrar os documentos procurados. Cedo foi verificado que a informação contida nas histórias se reporta a um contexto alargado e não apenas ao documento em si. É informação autobiográfica, sendo necessário para a compreender conhecer o utilizador e o mundo que o rodeia. Recorrer apenas a palavras-chave ou outro tipo de meta-informação preenchida pelos utilizadores não é suficiente, e seria ineficaz dada a relutância dos mesmos em fornecê-la. Decidimos, assim, recolher automaticamente a maior quantidade de informação possível sobre os utilizadores e as suas acções. Para tal, foi criado um sistema que monitoriza continuamente o que se passa no computador e guarda essa informação.

O sistema de monitorização, ainda em desenvolvimento, começa por recolher toda a informação já existente no computador na primeira vez em que é corrido, mantendo-a, a partir daí, actualizada. No momento em que o estudo foi realizado, apenas era recolhida a informação sobre os ficheiros do utilizador. As directorias de sistema são automaticamente filtradas, e um mecanismo de *blacklist/whitelist* permite o refinar dos critérios usados para a realidade de cada utilizador. Toda a informação relevante é extraída dos ficheiros, como o seu nome e data de criação e modificação. Para ficheiros de determinados tipos outra informação disponível é extraída. Por exemplo, dos ficheiros MP3 ou OGG é

extraída a meta-informação contida no seu cabeçalho ID3. Dos ficheiros textuais são extraídas palavras-chave, recorrendo ao algoritmo *tfidf* [Salton88], após terem sido divididos em *tokens* e sujeitos a um algoritmo de *stemming* para obter apenas as raízes das palavras tornando a aplicação mais robusta em termos de tempos verbais, género, número, etc. O algoritmo usado foi o algoritmo de Porter [Porter80], mas tanto o *stemming* como a divisão em *tokens* está feita de modo a que outros algoritmos possam ser aplicados sem grande esforço (bastando alterar o ficheiro de configuração), para lidar com documentos noutras línguas, por exemplo.

Toda a informação recolhida pelo sistema de monitorização é armazenada numa base de conhecimento. Desde o início que se tornou patente que para compreender as histórias seria necessário ter algum conhecimento de senso comum. Até uma coisa tão simples como referir que um documento foi escrito “por volta do Ano Novo” obriga a conhecimento adicional para ser interpretada: saber que o Ano Novo ocorre a 1 de Janeiro de todos os anos, e que provavelmente como o utilizador não referiu nenhum ano, está a falar do último. Assim, foi decidido usar uma base de conhecimento em que a informação estará devidamente estruturada de modo a permitir as inferências que serão necessárias para a correcta interpretação das histórias. Com a ajuda de ontologias, será possível enriquecer a base de conhecimento com toda a informação necessária para tal. Após um estudo dos formalismos usados para construir bases de conhecimento [Gonçalves04b], decidimos usar o RDF, parte da iniciativa da *Semantic Web*. Para além de ser um standard, espera-se que, com o passar do tempo, informação semântica representada nesse formalismo passe a permear a WWW. O acesso imediato a essa informação pode ser de uma ajuda inestimável para compreender o que está a interessar ou preocupar o utilizador a cada momento. Verificámos que o RDF tem um poder expressivo suficientemente grande para a nossa aplicação e que, caso se venha a mostrar necessário aumentá-lo, basta recorrer a outro nível de complexidade de representação sobre RDF (RDFS ou OWL) sem invalidar tudo o que já foi criado anteriormente.

Em termos de implementação, foi criada uma biblioteca em Python, a que chamámos Scroll, que esconde os detalhes de implementação referentes ao RDF e permite a preocupação apenas com questões de mais alto nível. Isto inclui a realização de inferência de nó e de caminho, e também a representação de conhecimento de forma semelhante a uma rede semântica, cujos *case-frames* estão definidos num *schema* designado por iQUILL, com uma expressividade perto da lógica de primeira ordem (exceptuando a quantificação existencial e negação). Outro *schema*, QUILL, define todos conceitos necessários para o armazenamento da informação recolhida sobre os documentos, páginas web, etc.

Sempre que é contada uma nova história, são criadas regras de inferência referentes à informação introduzida. As inferências correspondentes são então efectuadas no

Scroll, e aos documentos resultantes atribuído um valor numérico. Da combinação dos valores resultantes das diversas regras de inferência presentes na base de conhecimento num dado momento resulta uma pontuação global para os documentos. Os que estiverem melhor pontuados serão sugeridos ao utilizador na área de sugestão de documentos da interface.

3. METODOLOGIA

O objectivo do estudo era avaliar o grau de confiança que se pode ter na informação contida nas histórias contadas a um protótipo, e comparar essas histórias com as contadas a humanos de modo a verificar se partilham das mesmas propriedades. Para o realizar, foi necessário aceder aos computadores dos utilizadores. Só assim poderiam as histórias contadas ser comparadas com os documentos propriamente ditos, de forma a verificar quanta da informação nelas contida é fidedigna. Consequentemente, foi necessário entrevistar os utilizadores na sua casa ou no seu local de trabalho. Isto colocou algumas limitações ao número de utilizadores entrevistados. Preocupações de privacidade resultantes da necessidade de aceder a documentos reais foram, também, uma barreira. No total, foram entrevistados dez utilizadores, seis dos quais no seu local de trabalho. Houve um esforço consciente de que a amostra fosse a mais diversificada possível. Foram entrevistados utilizadores com profissões como consultor na área de informática ou advogado, entre outras. As suas idades estão compreendidas entre os 26 e os 56 anos, sendo seis homens e quatro mulheres.

Cada entrevista começou com uma pequena introdução em que foi explicado aos entrevistados o objectivo do estudo e como a entrevista iria decorrer. O protótipo foi então instalado nas suas máquinas, após lhes ser garantido que este estaria contido num único directório bem definido, pelo que seria possível apagá-lo sem deixar rasto uma vez terminada a entrevista. Enquanto o programa recolhia informação sobre os documentos do utilizador, o entrevistador dava um pequeno tutorial sobre o modo de funcionamento da interface, e preenchia um pequeno formulário com informação demográfica sobre o utilizador. Foi garantida a confidencialidade total aos entrevistados. Uma vez terminado este período introdutório, o processo de recolha de informação por parte do protótipo era interrompido, caso não tivesse ainda terminado. Isto deu origem a uma base de conhecimento incompleta, mas em nada comprometeu o estudo, uma vez que a validação das histórias foi feita manualmente. Permitir que o processo de recolha de informação chegasse ao fim seria demasiado moroso, dadas as restrições temporais colocadas para as entrevistas. Tentou-se que estas não ultrapassassem a uma hora de duração. Evitou-se assim interferir no trabalho dos entrevistados mais do que o estritamente necessário.

Aos utilizadores foi então pedido que contassem três histórias sobre três documentos diferentes: um documento Recente, criado por eles até há duas semanas atrás, um documento Antigo, criado entre um a dois anos atrás, e um documento de Outro Autor. Esses três tipos de

documentos tinham já sido considerados em estudos anteriores, e mantê-los permitiu a comparação directa das histórias entre os estudos. Para evitar que existisse alguma deturpação dos resultados devido à crescente familiaridade dos utilizadores com a interface ao longo da entrevista, as histórias sobre os documentos de diferentes tipos foram pedidas em ordens distintas de utilizador para utilizador (Tabela 1).

	1º lugar	2º lugar	3º lugar
Recente	5	3	2
Antigo	3	7	0
Outro	2	0	8

Tabela 1 – Posição das histórias na entrevista

O tempo gasto com cada história foi registado. Após o término de cada uma, foi pedido aos utilizadores que encontrassem manualmente o documento procurado (caso o Quill não tivesse sido capaz de o fazer). A informação com ele relacionada foi então anotada junto da informação referida na história, tendo sido avaliado se esta era correcta ou não. Estes dados foram armazenados para uma posterior análise, mais cuidada.

3.1 Avaliando a Correção da Informação

Confirmar a correcção dos vários elementos que constituem as histórias não é trivial. Nem todos os elementos são passíveis do mesmo grau de verificação. Por exemplo, se o nome do ficheiro onde o documento está guardado pode ser verificado inequivocamente, o mesmo não pode ser dito sobre as trocas desse documento com outros utilizadores. Isso implicaria, em última análise, verificar todas as mensagens de correio electrónico do utilizador, todos os discos amovíveis, etc. e mesmo assim seria impossível ter a certeza absoluta quanto à ocorrência ou não de uma certa troca. Para os elementos cuja verificação mais inequívoca foi impossível, foi estabelecido um diálogo com os utilizadores em que estes tinham que defender a informação dada. Várias perguntas foram feitas pelo entrevistador, com base na informação concreta sobre o documento que foi possível encontrar. Para que o elemento fosse considerado como válido a explicação dada devia ser consistente com essa informação e convencer o entrevistador. Este tentou sê-lo exaustivo quanto possível nas questões colocadas.

Mesmo assim, e para garantir a correcção dos resultados apresentados, foram distinguidos dois níveis diferentes de validade dos elementos. Por um lado, encontramos aqueles que cuja veracidade foi inequivocamente verificada (comparando o nome de ficheiro do documento com o dado na história pelos utilizadores, por exemplo). Por outro, temos os que foram considerados como correctos após questionar os utilizadores. A análise dos resultados será efectuada tendo sempre presente esta distinção.

Para alguns dos elementos mais problemáticos, estratégias específicas de validação foram usadas. As mais importantes estão referidas em seguida. Uma lista mais completa pode ser encontrada no relatório técnico que descreve o estudo [Gonçalves05]:

- **Razão:** nalguns casos era possível verificar a razão de ser do documento a partir do seu conteúdo. Caso contrário, os utilizadores foram questionados.
- **Outros Documentos:** sempre que possível, pedimos para consultar também esses documentos (frequentemente guardados na mesma pasta), confrontando os utilizadores com eles, fazendo-os verificar se se os tinham recordado correctamente. Em casos em que os documentos relacionados eram citados no documento principal, a verificação foi imediata e inequívoca.
- **Vida Pessoal:** se a informação estivesse nas agendas dos utilizadores, ou fosse conhecida pessoalmente pelo entrevistador era considerada como correcta..
- **Acontecimentos no Mundo:** estes acontecimentos foram verificados a partir da memória dos utilizadores e do próprio entrevistador, recorrendo a fontes noticiosas *online* para confirmar que ocorreram na altura referida pelo utilizador.
- **Trocas:** tentámos perceber se a informação estava correcta a partir do conteúdo do documento e aparente fim a que se destinava.
- **Tarefas:** se as tarefas mencionadas se reflectissem no conteúdo do documento (inserir imagens ou preparar gráficos, por exemplo), seriam consideradas como válidas
- **Acontecimentos** (ocorridos durante a interacção com o documento): a única forma de confirmar se esta informação estava correcta ou não foi dialogar com os utilizadores

4. RESULTADOS

Nesta secção iremos descrever os principais resultados do estudo. Começaremos por verificar se as histórias contadas usando a interface são ou não similares às contadas a entrevistadores humanos. Em seguida avaliaremos a veracidade da informação contida nas histórias, tanto em termos globais com de cada um dos elementos que as constituem.

4.1 Contando as Histórias...

Nenhum utilizador necessitou de explicações adicionais quanto à utilização da interface após o tutorial inicial. Com algumas excepções, todos os elementos da interface foram correctamente compreendidos e usados. Em seguida apresentamos uma das histórias contadas por um dos utilizadores, copiada directamente da interface (após alterações para garantir o anonimato do entrevistado):

O autor deste documento fui eu. Foi criado por volta de há um mês. A razão

pela qual o criei foi para um relatório de trabalho. Quando trabalhei no documento estava no local de trabalho. Foi escrito apenas por mim. O documento é sobre Data Warehouse. este documento lembra-me outro chamado "adenda" (areas) porque foi manuseado na mesma altura e foi escrito pela mesma razão e era sobre o mesmo assunto e está guardado na mesma pasta (ou lá perto...). Dei-o a [colega1] e [colega2] usando email e LAN (pastas partilhadas, etc.). É um documento do tipo Microsoft Word. O documento contém as palavras ou expressão "Adenda Relatório Tardes DW Projectos" e tem o aspecto de um texto a uma coluna. O documento está no Computador no Local de Trabalho. Para o escrever, tive que recorrer a outros documentos. Teve várias versões. O nome do ficheiro era algo como "tardes DW". Nessa altura não aconteceu nada de especial. Na altura em que o criei, na minha vida pessoal (amigos, família, trabalho) não aconteceu nada de especial. Quando estava a trabalhar no documento, nada de especial aconteceu.

A interface podia ser configurada para funcionar em Inglês ou Português, de acordo com as preferências dos utilizadores. A historia apresentada pertence a um dos dois utilizadores que o fez em Português. O Português usado não é perfeito, mas é suficientemente bom para que a história seja satisfatoriamente legível.

Foi observada uma curva de aprendizagem da interface, reflectida no tempo levado pelos utilizadores a contar as histórias (Figura 2). Em média, o tempo levado para contar a terceira história era apenas 60% do gasto com a primeira, independentemente dos tipos de documentos descritos (de 427 a 269 segundos).

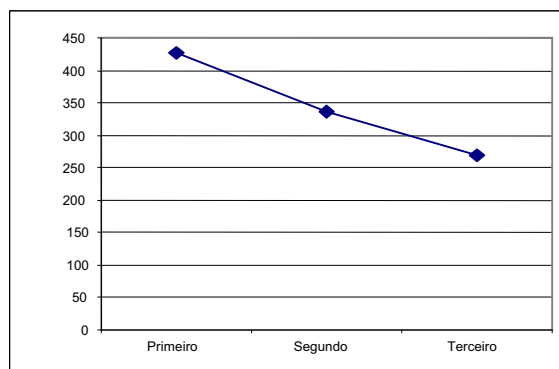


Figura 2 – Tempo gasto a contar as histórias (em seg.)

4.2 Comparando as Histórias

É importante verificar até que ponto os utilizadores continuam a sentir que estão a contar histórias e não apenas a dar valores a propriedades usando uma interface

no computador. Para tal, as histórias contadas usando o Quill foram comparadas às contadas a um entrevistador humano. Se partilhassem da mesma estrutura e conteúdos, estaria efectuada a nossa validação. Mais concretamente, usámos como base de comparação as histórias encontradas durante a fase de avaliação de protótipos de baixa fidelidade, comprovadamente semelhantes às contadas a um entrevistador [Gonçalves04]. Comparámos tanto a estrutura quer o conteúdo das histórias.

4.2.1 Estrutura das Histórias

As histórias agora recolhidas têm comprimentos semelhantes aos das anteriormente analisadas: cerca de 14 elementos (Tabela 2). A razão entre os comprimentos das histórias actuais e das anteriores são de 98%, 101,5% e 100,7%, para os documentos Recentes, Antigos e Outros, respectivamente. Testes de t-student confirmaram que esses valores são iguais, com 95% de confiança.

	Actual		Anterior	
	Média	Desv. Pad	Média	Desv. Pad
Recente	14	1.05	14.3	2.06
Antigo	13.5	1.08	13.3	1.25
Outro	13.4	1.43	13.3	2.06

Tabela 2 – Comprimento das Histórias (em elementos)

Avaliámos a ordem na qual os diversos elementos ocorrem nas histórias tendo em conta que a ordem em que eram sugeridos aos utilizadores era idêntica à inferida a partir das histórias contadas a humanos. Assim, qualquer desvio dessa ordem indicaria que esta não era considerada natural neste contexto. Foi contabilizado o número de vezes que o utilizador pediu um elemento diferente do sugerido, usando o botão para esse fim disponível na interface.

Apenas um utilizador pediu um elemento diferente do sugerido, uma vez em cada uma das suas histórias, resultando numa média global de 0.1 utilizações por história. Podemos, assim, concluir que a ordem na qual os elementos eram sugeridos aos utilizadores era considerada natural por estes.

Em suma, a estrutura das histórias contadas usando o protótipo é semelhante à das histórias contadas a humanos.

4.2.2 Conteúdo das Histórias

Uma vez que as histórias possuem estruturas semelhantes, é espectável que os seus conteúdos também o sejam. Uma verificação mais cuidada deste facto será descrita em seguida.

Um dos aspectos a ter em conta é a importância dos vários elementos nas histórias. Para verificar se esta se mantinha inalterada, verificámos as frequências com que esses elementos aparecem nas histórias recolhidas em ambos os estudos. O gráfico na Figura 3 mostra os valores dessas frequências.

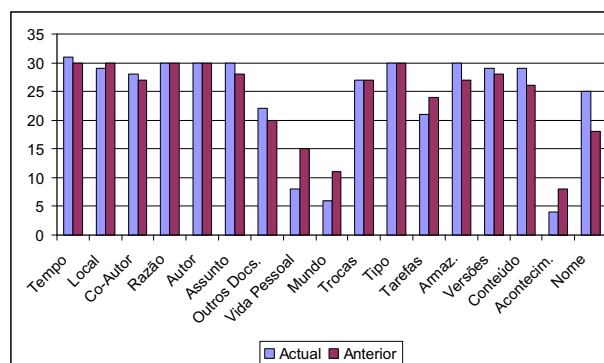


Figura 3 – Comparação das frequências dos elementos

É imediatamente aparente que, com poucas excepções, as frequências são bastante semelhantes. Diferenças relevantes podem ser observadas apenas para quatro elementos. O Nome dos documentos é mencionado 39% mais frequentemente nas histórias contadas usando o protótipo. Isto pode ser explicado por duas causas diferentes. Em primeiro lugar, os utilizadores estavam, na sua maioria, a trabalhar com o computador imediatamente antes da entrevista, o que os pode ter tornado mais atentos às convenções de nomes que costumam usar, refrescando a sua memória. Em segundo lugar, o simples facto de estarem a lidar um com computador pode ter influenciado a sua propensão para mencionar informação relacionada com este, como é o caso dos nomes dos ficheiros. Seja como for, a diferença encontrada é positiva, uma vez que mais informação é sempre útil para encontrar um documento.

Já os elementos sobre a Vida Pessoal, Acontecimentos no Mundo e Acontecimentos ocorrem menos frequentemente (47, 45 e 50 por cento, respectivamente). Estes são os três elementos que têm mostrado ser menos confiáveis e difíceis de recordar ao longo de todos os estudos efectuados. A maioria dos utilizadores pura e simplesmente não os associa aos documentos que manipula. Foi encontrada uma grande variabilidade individual no que diz respeito a estes elementos. Alguns utilizadores nunca os usam, enquanto que outros recorrem a eles esporadicamente. Dada a baixa frequência absoluta com que ocorrem, não é possível tirar conclusões fiáveis a partir das diferenças observadas.

Passando à importância relativa dos vários elementos, independentemente do valor absoluto de ocorrências encontrado, comparar simplesmente a ordem na qual os elementos ocorrem não seria correcto, uma vez que pequenas variações em valor absoluto podem resultar em alterações nessa ordem. Em vez disso, os elementos foram divididos em duas classes: os que são mencionados em quase todas as histórias e os que apenas o são raramente. O valor limite que os separa é terem sido mencionados em 70% das histórias. Em redor a esse valor não há elementos, tendo todos frequências claramente maiores ou menores. Esta divisão foi feita para as distribuições de elementos nos dois conjuntos de histórias. Foi, assim, possível encontrar os elementos cuja importância relativa se alterou (i.e., mudaram de classe).

Para os documentos Recentes, não encontramos quaisquer diferenças. Para os documentos Antigos e Outros, e no geral, apenas o elemento Nome mudou de importância. Como já foi anteriormente mencionado, este elemento foi referido mais frequentemente nas histórias contadas usando a interface. Passou assim de elemento raro a comum.

Para concluir, as histórias contadas ao computador usando o protótipo mantiveram-se, no geral, idênticas às contadas a interlocutores humanos.

4.3 Validade das Histórias

Iremos agora estudar a validade das histórias, verificando até que ponto podemos confiar na informação nelas contida. Atentaremos, em primeiro lugar, à correcção geral das histórias, e focaremos em seguida a de cada um dos seus elementos.

4.3.1 Validade Geral

O gráfico na Figura 4 mostra as percentagens de elementos válidos encontrados nas histórias de cada um dos diferentes tipos. Como já foi mencionado acima, considerámos duas classes diferentes de elementos: os cuja validade foi verificada inequivocamente e os que fora considerados correctos sem que tenha sido possível prová-lo cabalmente.

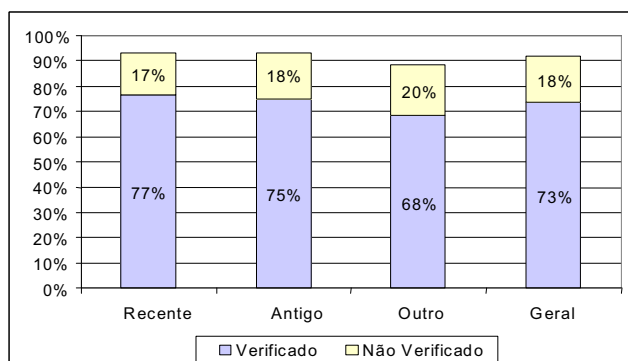


Figura 4 – Veracidade das Histórias

Não parecem existir diferenças relevantes entre as histórias para os documentos Recentes e Antigos, revelando que os utilizadores se conseguem lembrar igualmente bem da informação para documentos desses tipos. As histórias sobre documentos de Outros autores, por outro lado, parecem ligeiramente menos precisas do que as dos documentos do próprio utilizador. Testes de t-student confirmam isto, com um nível de confiança de 95%.

Já no respeitante à validade das histórias no caso geral, verificámos que, embora não perfeita, é razoavelmente boa. Mais de 91% dos seus elementos constituintes são verdadeiros. Mesmo considerando apenas os inequivocamente verificados, o valor é de cerca de 70%. Em média, entre 73% e 92% do que os utilizadores mencionam nas histórias corresponde à verdade.

Olhando mais atentamente para os dados, podemos verificar que a percentagem de elementos não verificados (17, 18 e 20 por cento para os três tipos de documentos, respectivamente) é devida, principalmente, a três

elementos: Vida Pessoal, Acontecimentos no Mundo e Acontecimentos. Esses três elementos explicam 59, 41 e 47 por cento de todos os elementos não verificados para os três tipos de documentos. Isto deve-se ao método de verificação usado: dado que a maior parte dos utilizadores se limitou a dizer que “nada aconteceu” ou que “não se lembram” desses elementos, não tínhamos forma de o verificar. O elevado número de vezes em que isto ocorreu reflecte o facto desses elementos não estarem a ser recordados em relação aos documentos. O utilizador não os mencionar é tão válido como se realmente nada tiver acontecido. Adicionalmente, mesmo que algo tenha de facto ocorrido não estamos na presença de informação incorrecta tanto como estamos na ausência de informação. Posto isto, e recalculando as percentagens de excluindo esses elementos, obtemos os resultados mostrados no gráfico da Figura 5. Podemos aí constatar que entre 81% e 91% das histórias pode ser considerada correcta. Isto corresponde apenas a de 1 a 3 elementos errados por história, mais do que suficiente para as usar como base para a recuperação de documentos.

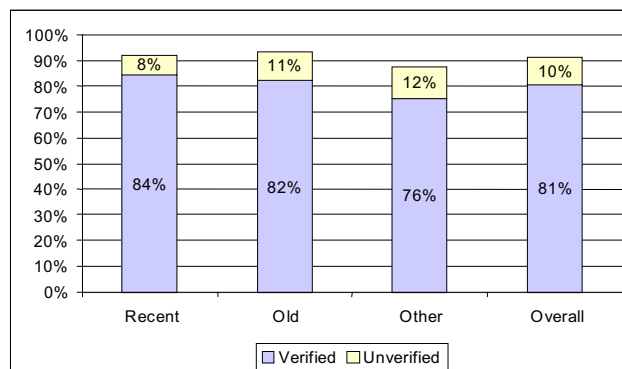


Figura 5 – Veracidade das Histórias (corrigida)

4.3.2 Correcção dos Elementos

É importante saber quais os elementos mais frequentemente errados, para melhor lidar com as suas eventuais imprecisões. O gráfico na Figura 6 representa a veracidade dos vários elementos separadamente, para todos os tipos de documentos.

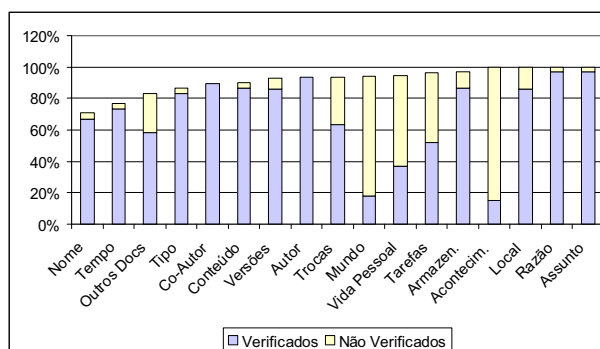


Figura 6 – Veracidade dos Elementos

Alguns elementos foram notoriamente difíceis de verificar. É o caso dos três menos precisos, Vida Pessoal, Acontecimentos no Mundo e Acontecimentos, já discutidos na secção anterior. Também nesta situação

estão as Trocas e Tarefas. Na maior parte dos casos estávamos confiantes na sua correcção, mas foi impossível confirmá-lo.

O Nome parece ser o elemento mais infrequentemente lembrado com precisão. Nalguns casos o utilizador estava errado, mas na maior deles os utilizadores tinham alguma ideia do nome correcto (“janeiro2005” em vez de “2005janeiro”, Utilizador 7), o nome real tinha sido abreviado, ou parte do nome sugerido pelos utilizadores pertencia ao nome real (“estudo produtividade bolseiros” em vez de “produtividade bolseiros – calculos”, Utilizador 4).

Quanto ao Tempo, na maior parte dos casos os elementos errados estavam perto da data correcta, mas mesmo assim fora dos intervalos de tolerância considerados. Talvez estes necessitem de ser ajustados à luz destes resultados.

O terceiro elemento menos preciso são os Outros Documentos. Frequentemente, os utilizadores lembravam-se correctamente de alguma informação, mas outra estava errada. Por exemplo, o Utilizador 9 mencionou correctamente que o documento versava sobre mesmo assunto que o procurado, mas enganou-se no nome referido. O Utilizador 4 referiu os autores correctos mas enganou-se, também, no nome. De um modo geral, os mesmos problemas encontrados na descrição do documento principal ocorreram também na descrição dos documentos relacionados.

Confusões quanto ao Tipo deveram-se, principalmente, a problemas a distinguir entre tipos semelhantes: texto simples vs Microsoft Word, Powerpoint em vez de imagem (Utilizador 1), ou PDF em vez de Word (Utilizador 7). Os documentos colocados *online* também trouxeram problemas. Por exemplo, o Utilizador 6 mostrou-se confundido sobre se uma base de dados em Microsoft Access que tinha disponibilizado no seu *web site* deveria ser considerada uma base de dados ou um documento web.

Todos os elementos restantes tiveram precisões acima dos 90%. Não foi identificado nenhum outro padrão de erro como os que acabámos de descrever.

Finalmente, no respeitante a eventuais diferenças de veracidade em função do tipo de documento, o único aspecto a referir é que o Autor é recordado correctamente muito menos frequentemente para os documentos de Outros Autores, do que para os do utilizador (80% a 100%), resultado pouco surpreendente.

5. DISCUSSÃO

No que diz respeito à comparação entre as histórias contadas a humanos e as contadas usando a interface, *não foram encontradas diferenças significativas*. A estrutura das histórias mantém-se idêntica, tal como o fazem os seus conteúdos. A única excepção é o elemento Nome, que aparece mais frequentemente usando o computador. Isto confirma que é possível manter a sensação de “contar uma história” usando o computador, sem o qual a construção de interfaces baseadas em narrativas para a localização de documentos seria impossível.

Quanto à veracidade das histórias, verificámos que, *de um modo geral, podemos acreditar nos utilizadores*. Foram envidados esforços para verificar para além de qualquer dúvida razoável a validade dos vários elementos que constituem as histórias. Conseguimos fazê-lo para 81% dos elementos e constatámos que entre 73% e 91% da informação nas histórias está correcta. Se alguns ajustamentos forem feitos tendo em conta a natureza de alguns dos elementos, este intervalo sobe para de entre 81% e 91% (1 a 3 elementos por história). Isto implica que a informação nas histórias não pode determinar taxativamente quais os documentos sugeridos aos utilizadores. Em vez disso, um mecanismo que tenham em conta de forma limitada a contribuição dos vários elementos é necessário, minorando assim os efeitos adversos de alguma informação errónea. Vimos ainda que nalguns casos é possível saber que espécie de imprecisões esperar e agir em conformidade (melhores margens de tolerância para o Tempo, por exemplo).

Os elementos menos precisos foram identificados: Nome, Tempo, Outros Documentos e Tipo. Igualmente, verificámos que nos documentos de outros autores, o elemento Autor é recordado com bastante menor precisão do que no caso dos restantes.

Outro resultado importante é que as histórias sobre os documentos do utilizador, quer Recentes quer Antigos, partilham as mesmas propriedades. Alguns resultados de estudos anteriores apontavam já nesta direcção. Este é, no entanto, um resultado surpreendente, uma vez que se poderia esperar que a informação sobre os documentos mais antigos fosse recordada com muito menor precisão. No entanto, tal não foi observado. Em termos do desenho de interfaces baseadas em narrativas, isto indica que estas podem evitar fazer a distinção entre estes dois tipos de documentos, tratando de modo distinto apenas os documentos do utilizador e os de outros autores.

Sobre a interface usada, podemos afirmar que esta é fácil de aprender e usar. Um pequeno tutorial informal foi suficiente para a sua utilização, e após apenas três histórias terem sido contadas, o tempo gasto para o fazer decresce para 60% do tempo usado com a primeira interacção. Isto mostra que as escolhas feitas com base na avaliação de protótipos de baixa fidelidade estavam correctas.

6. CONCLUSÕES

Encontrar documentos nos sistemas disponíveis hoje em dia é uma tarefa morosa e difícil. Isto deve-se não só aos números crescentes de documentos com que temos que lidar, como à diversificação dos tipos destes. As formas tradicionais de organização e localização não avançaram de modo a fazer frente a esta nova realidade.

As narrativas sobre documentos parecem uma boa base para a criação de interfaces que permitam aos utilizadores fornecer ao computador toda a informação autobiográfica de que se recordam quando procuram um documento, de forma natural e eficiente. Para validar a sua aplicabilidade, era necessário responder a duas questões:

podem histórias semelhantes às contadas a outros humanos ser contadas ao computador? É a informação contida nas histórias suficiente precisa para ser usada como base na localização de documentos?

Os resultados do estudo descrito neste artigo permitem-nos responder afirmativamente a ambas questões. As histórias contadas ao computador são semelhantes às contadas a humanos, e a informação é, de um modo geral (cerca de 90% dos casos), correcta. A adequação das narrativas para a localização de documentos foi, pois, confirmada.

No futuro, e tendo em conta as lições aprendidas neste estudo, o protótipo será melhorado de modo a mais se adequar às reais necessidades dos utilizadores. Iremos integrar o sistema de monitorização com a interface, recolhendo mais informação que nos permitirá melhor compreender as histórias contadas. Actualmente, recolhe já informação referente ao correio electrónico do utilizador, as páginas web a que este acede e as aplicações corridas.

Assim que a interface tiver evoluído o suficiente, efectuaremos testes mais alargados com os utilizadores. Nesses testes, ao longo de maiores períodos de tempo, conseguiremos ter uma melhor ideia da curva de aprendizagem da interface, e responder a uma terceira e premente questão: contêm as histórias informação suficiente para distinguir entre documentos semelhantes mas diferentes?

7. AGRADECIMENTOS

Os autores agradecem a todos os utilizadores que participaram no estudo descrito neste artigo. O trabalho aqui apresentado foi financiado pelo projecto FCT POSI/EIA/59022/2004.

8. BIBLIOGRAFIA

- [Baeza-Yates96] Baeza-Yates, R., Jones, T. and Rawlins, G. A New Data Model: Persistent Attribute-Centric Objects, Technical Report, University of Chile, 1996
- [Dourish00] Dourish, P. *et al.* Extending Document Management Systems with User-Specific Active Properties. *ACM Transactions on Information Syst.*, 18(2), pp 140-170, ACM Press 2000.
- [Dumais03] Dumais, S. T. *et al.* Stuff I've Seen: A system for personal information retrieval and re-use. In *Proceedings of SIGIR 2003*.
- [Freeman96] Freeman, E. and Gelernter, D. Lifestreams: A Storage Model for Personal Data, *ACM SIGMOD Record*, 25(1), pp 80-86, ACM Press 1996
- [Gifford91] Gifford, D., Jouvelot, P., Sheldon, M. and O'Toole, J. Semantic File Systems. *13th ACM Symposium on Principles of Programming Languages*, October 1991.
- [Gonçalves04] Gonçalves, D. and Jorge, J., "Tell Me a Story": Issues on the Design of Document Retrieval Systems. In *Proceedings DSV-IS'04*, Lecture Notes on Computer Science, Springer-Verlag, July 2004, Hamburg, Germany.
- [Gonçalves04a] Gonçalves, D. and Jorge, J. Telling Stories to Computers. In *Proceedings CHI2004*, ACM Press, 27-29 April 2004, Vienna, Austria.
- [Gonçalves04b] Gonçalves, D. and Jorge, J. Why RDF? Considerations on a Language for the Representation of Document-Describing Stories. *Technical Report, IST/UTL*. June 2004, (disponível online em http://immi.inesc.pt/~djvg/phd/files/why_rdf.pdf)
- [Gonçalves05] Gonçalves, D. Real Stories about Real Documents: Evaluating the Trustworthiness of Document-Describing Stories. *Technical Report, IST/UTL*. March 2005, (disponível online em http://immi.inesc.pt/~djvg/phd/files/real_stories.zip)
- [Malone83] Malone, T. How do People Organize their Desks? Implications for the Design of Office Information Systems, *ACM Transactions on Office Information Systems*, 1(1), pp 99-112, ACM Press 1983.
- [Porter80] Porter, M. F. An algorithm for suffix stripping. *Program* 14, 130-137, 1980.
- [Rekimoto99] Rekimoto, J. Time-machine computing: a time-centric approach for the information environment. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 45-54, ACM Press, 1999.
- [Salton88] Salton, G. Automatic text processing, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1988.
- [Shamir04] Shamir, A.. A View on Views. In *Proceedings SmartGraphics04*, Banff Center, Canada, May 2004.
- [Whittaker96] Whittaker, S., Sidner, C. Email overload exploring personal information management of email. In *Conference proceedings on Human factors in computing systems*, pages 276-283, ACM Press, 1996.

Plataforma de Suporte à Geração de Cenas Animadas com Agentes Inteligentes

M. Beatriz Carmo, A. Paula Cláudio, João D.
Cunha, Helder Coelho,
Miguel Silvestre

FCUL-DI, Campo Grande 1749-016 Lisboa

{bc,apc,jdc,hcoelho}@di.fc.ul.pt,
msilvestre@gmail.com

Maria Pinto-Albuquerque

DCTI, Instituto Sup. de Ciências do Trabalho e da
Empresa

Av. das Forças Armadas Edifício ISCTE 1649-026
Lisboa

maria.albuquerque@iscte.pt

Resumo

Neste artigo apresenta-se uma plataforma de suporte à geração de cenas animadas com agentes inteligentes. É proposta uma arquitectura que separa o processamento gráfico do processamento inteligente. A plataforma que a concretiza incorpora várias aplicações que são articuladas por dois módulos independentes que comunicam entre si: a camada de processamento gráfico e a camada de processamento inteligente.

Palavras Chave: *humanos virtuais, ambientes virtuais, animação comportamental.*

1. INTRODUÇÃO

A construção de ambientes virtuais onde se movimentam humanos virtuais inteligentes é cada vez mais comum. Estes ambientes têm aplicação em áreas tais como entretenimento, educação, treino em ambientes industriais, reconstituição de ambientes históricos, aplicações médicas, tratamento de fobias.

A representação de humanos virtuais, quer em ambientes virtuais, quer em ambientes reais, como acontece na Realidade Aumentada, é um tema de investigação onde se combinam contribuições de várias disciplinas, nomeadamente, da Computação Gráfica, da Inteligência Artificial e das Ciências Sociais. O trabalho descrito neste documento enquadra-se no projecto MAGO2, Modelação de Agentes em Organizações. Este projecto, desenvolvido no LabMAG, Laboratório de Modelação de Agentes [http:LabMAG], integra várias tarefas relacionadas com a estrutura e dinâmica de sociedades de agentes. A tarefa do MAGO2 que corresponde ao trabalho descrito propõe-se concretizar uma biblioteca de representações gráficas para agentes inteligentes, que possa vir a suportar estudos sobre o comportamento de agentes individualmente ou integrados em sociedade.

Têm sido apresentadas diversas concretizações de plataformas para a geração de cenas animadas com agentes inteligentes. Na secção 2 são referidas algumas destas soluções. Na secção 3 apresenta-se o trabalho desenvolvido e na secção 4 conclui-se descrevendo o trabalho em curso e o trabalho futuro.

2. TRABALHO RELACIONADO

A geração de ambientes virtuais com agentes virtuais autónomos implica a integração de ferramentas gráficas com módulos de processamento inteligente. Referem-se

em seguida trabalhos que explicitamente se debruçam sobre a integração destas duas componentes.

O trabalho de Tu et al. [Tu94] sobre a simulação artificial de peixes é um caso particular da animação de agentes inteligentes. Neste caso a modelação de cada agente incorpora três componentes: o sistema motor, o sistema de percepção e o sistema de comportamento. O sistema motor baseia-se em modelos físicos para gerar o movimento dos peixes. O sistema de percepção capta a informação sobre o meio envolvente através de um sensor de visão e de um sensor da temperatura da água. O sistema de comportamento faz a mediação entre os sistemas sensorial e motor. A modelação do comportamento tem em atenção a definição dos hábitos do peixe, do seu estado mental e a informação obtida através dos sensores. Os hábitos do peixe são definidos interactivamente pelo utilizador através do ajustamento de um conjunto de parâmetros. Funge et al. [Funge99] estendem este trabalho, incluindo no modelo de comportamento, um modelo cognitivo que permite aos agentes virtuais tomarem decisões baseando-se em conhecimento adquirido sobre o mundo. O modelo de Funge utiliza o formalismo de cálculo de situações, utilizado em Inteligência Artificial, para a descrição de alterações no mundo envolvente.

O sistema desenvolvido por Perlin et al. [Perlin96], Improv, permite a criação de actores animados com comportamentos próprios e que interagem entre si e com o utilizador. Este sistema integra duas componentes fundamentais. A primeira, *animation engine*, permite a criação de animações, incluindo ferramentas para gerar movimentos realistas. Além do recurso a técnicas de *inverse kinematics* e de deformação de superfícies, usa funções de ruído para combinar diferentes movimentos

numa só acção [Perlin95]. Os actores, quer sejam humanos, animais ou seres imaginários, são constituídos por segmentos ligados por articulações. A segunda componente, *behavior engine*, controla as regras de comportamento dos actores. Os diferentes tipos de comportamento dos actores são fornecidos numa linguagem de *script* própria. A selecção do comportamento de um actor baseia-se em regras de decisão e no peso atribuído pelo autor a cada comportamento. O sistema Improv recorre à biblioteca gráfica SGI GL e é implementado como um conjunto de programas distribuídos num ambiente Unix comunicando entre si através de *sockets*, protocolos *multicast* e *pipes*.

Devillers et al. [Devillers02] construíram uma plataforma para a animação de agentes inteligentes que designaram por GASP (*General Animation and Simulation Platform*). Esta plataforma inclui vários módulos que contemplam a modelação de diferentes aspectos dos agentes, tais como, controlo do movimento, visão, comportamento. A execução pode ser distribuída por vários processadores recorrendo ao PVM (*Parallel Virtual Machine*) para realizar a comunicação entre eles [Donikian98].

Outra proposta de arquitectura para a animação de humanos virtuais com raciocínio cognitivo é apresentada em [Torres02]. Esta arquitectura inclui uma componente gráfica, uma componente de raciocínio e uma interface entre estas duas componentes [Torres03]. A representação gráfica dos humanos é definida por um modelo hierárquico que representa o conjunto de articulações do corpo humano e um conjunto de funções para gerar movimentos nessas articulações. São usados ficheiros XML para definir a representação e a movimentação dos agentes. A componente de raciocínio é implementada através da linguagem AgentSpeak(XL) [Bordini02]. Esta linguagem é uma extensão de uma linguagem de programação de agentes AgentSpeak(L) proposta em [Rao96] para implementar agentes de acordo com o modelo BDI (*Beliefs, Desires, Intentions*). A interface faz a comunicação entre o “corpo articulado” e a mente. A interface recebe do ambiente gráfico informação sobre o estado do ambiente e comunica-a à componente de raciocínio. Esta, por sua vez, decide qual o comportamento adequado para o agente e fornece à interface esta informação. A componente de raciocínio é independente da componente gráfica e comunica com esta através de *sockets*.

Daniel Thalmann et al. [Thalmann04] propõem também uma arquitectura baseada na separação entre a simulação física e a simulação comportamental. Neste caso são identificadas duas componentes, uma para cada um dos dois tipos de simulação: o *Agents' Common Environment* (ACE) e o *Intelligent Virtual Agent* (IVA). O ACE contém um conjunto de comandos para controlar a simulação do mundo físico como, por exemplo, comandos para criar e animar objectos 3D e humanos virtuais capazes de exprimir emoções [Kallmann00].

Engloba ainda animações básicas, como por exemplo vários tipos de locomoção de humanos virtuais. As linguagens usadas para este nível são o C++ e a linguagem de *scripting* Python. Esta última é usada para a criação de comandos de animação dos agentes a incluir no cenário. Cada agente corresponde a uma *thread* de modo a poder interagir de modo independente com o ambiente 3D que o rodeia. Os agentes comunicam entre si e sincronizam-se com os restantes elementos do cenário, usando uma área comum de memória que é gerida por uma entidade designada por *Agents Controller*. Esta área comum é especialmente útil quando os agentes estabelecem entre si comunicação verbal. A animação em tempo-real dos humanos virtuais está a cargo da aplicação AgentLib [Boulic97], integrada no ACE. A animação produzida explora várias técnicas: sequências de *keyframes*, *inverse kinematics* e captura de movimentos (*motion capture*). O IVA baseia-se na arquitectura BDI [Caicedo00]. Esta componente foi desenvolvida em Java [Monzani01]. Cada entidade da simulação que seja dotada de inteligência corresponde a uma *thread* no IVA. Cada uma destas *threads* comunica, através de uma ligação TCP/IP, com a *thread* que lhe corresponde no ACE. Contudo, tal como os humanos não comunicam entre si usando ligações directas entre os seus cérebros, as *threads* do IVA não estabelecem qualquer tipo de comunicação entre si.

Vosinakis et al. [Vosinakis05] desenvolveram uma ferramenta, *SimHuman*, para a construção de ambientes virtuais com agentes autónomos. O *SimHuman* é composto por uma biblioteca escrita em C++ e recorre à biblioteca gráfica OpenGL. Uma aplicação pode incorporar chamadas a esta biblioteca para fazer o *rendering* e animação de um ambiente virtual interactivo 3D com um número arbitrário de objectos e humanos virtuais, quer com comportamentos próprios, quer controlados pelo utilizador. O *SimHuman* inclui ainda dois utilitários: um que ajuda na criação e animação de agentes (*AgentDesigner*) e outro que facilita a colocação de objectos numa cena e a visualização de sequências de acções dos agentes (*Scene Designer*). O movimento e acção dos agentes tem subjacente a utilização de modelos baseados na Física (*Physically Based Modeling*), *Inverse Kinematics*, detecção de colisões e visão dos agentes. O modelo geométrico dos agentes pode ser definido utilizando uma aplicação comercial, *Curiouslabs Poser*. Por outro lado, a geometria dos objectos que compõem a cena pode ser definida em qualquer *software* de modelação que exporte ficheiros no formato VRML97.

Nos trabalhos referidos é nítida a separação entre a componente gráfica e a componente de inteligência artificial. Esta foi também a abordagem adoptada na arquitectura que propomos para a construção de ambientes virtuais habitados por seres dotados de comportamento inteligente.

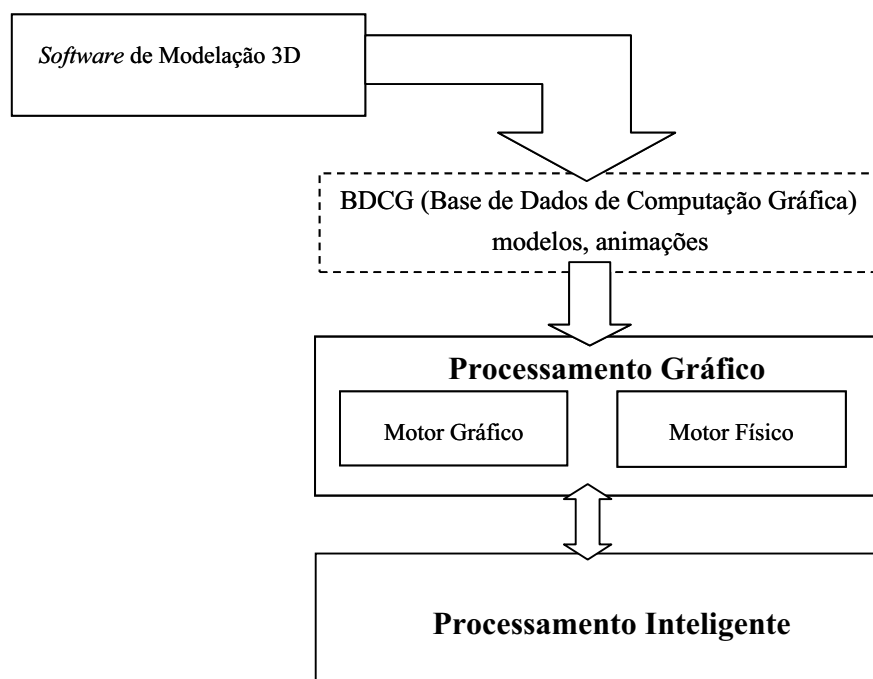


Fig. 1 – Arquitectura de suporte à geração de cenas animadas com agentes inteligentes

3. ARQUITECTURA ADOPTADA E PLATAFORMA DESENVOLVIDA

A arquitectura que propomos para suportar a geração de cenas animadas com agentes inteligentes tem duas componentes fundamentais: uma componente gráfica e uma componente de inteligência artificial. O parágrafo seguinte explica em traços gerais o que levou à concepção da arquitectura proposta e que a figura 1 ilustra.

Relativamente à componente gráfica, um dos elementos fundamentais é um *software* de modelação 3D para criar os elementos da cena e gerar as animações pretendidas. Com o objectivo de escolher uma abordagem que permita a geração de imagens em tempo-real, optou-se por juntar um motor gráfico. Como o motor gráfico nem sempre resolve alguns aspectos da simulação do comportamento físico dos objectos da cena como, por exemplo, as colisões ou a aplicação de forças, considerou-se necessária a inclusão de um motor físico.

Uma vez que o motor gráfico importa o modelo criado pelo *software* de modelação através da leitura de um ficheiro, esta arquitectura é independente do *software* de modelação, o que consideramos ser uma vantagem considerável. Além do programa de modelação, do motor gráfico e do motor físico, esta arquitectura fica aberta à inclusão de outras ferramentas.

Quanto à componente de processamento inteligente optou-se pela ligação a uma bancada de agentes. Depois de definida a arquitectura houve que tomar decisões sobre o *software* a utilizar nas duas componentes. O processo de desenvolvimento de ferramentas gráficas e de inteligência artificial é normalmente moroso e exigente do ponto de vista de recursos humanos. Para reduzir o tempo de criação de uma plataforma de trabalho optou-se por incorporar aplicações já existentes e de

domínio público. Nas subsecções seguintes justificamos as escolhas efectuadas e descrevemos os detalhes mais relevantes da concretização da plataforma.

3.1 Componente Gráfica

Como ponto de partida e com o objectivo de escolher de forma ponderada as ferramentas a adoptar, iniciou-se uma pesquisa sobre as ferramentas de modelação existentes. Compararam-se as diferentes ferramentas com base no seguinte conjunto de características: capacidades de modelação e de animação disponibilizadas, facilidades de exportação e de importação de ficheiros, quantidade de ajuda acessível através da internet e, por último, contextos efectivos de utilização da ferramenta [Silvestre04].

O Blender [<http://Blender>] foi o *software* de modelação 3D escolhido. Trata-se de um *software* gratuito, com uma grande comunidade de utilizadores e que tem estado em contínuo desenvolvimento. Apesar de se ter considerado a interface menos amigável do que a de outras ferramentas, concluiu-se que possui fortes capacidades de modelação gráfica e permite importar ficheiros de vários tipos.

Relativamente ao motor gráfico foi adoptado o OGRE [<http://Ogre>]. Trata-se de um motor genérico, isto é, não foi desenvolvido para um tipo particular de aplicação. O OGRE é um motor bastante flexível, modular, eficiente, que conta com uma grande comunidade de desenvolvimento e de utilizadores. As imagens geradas pelo OGRE são de qualidade superior às de outros motores gráficos analisados. Uma vez que o OGRE não inclui um sistema de simulação de fenómenos físicos ou de colisões, incluiu-se no processamento gráfico um motor físico que permite simular estes fenómenos.

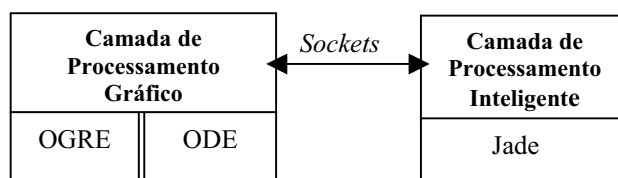


Fig. 2 Ligação entre a componente gráfica e a componente inteligente

O ODE [httpODE] foi o motor físico escolhido uma vez que é muito usado, bastante referenciado, com uma grande comunidade de utilizadores e com bons materiais de apoio, incluindo vários exemplos sobre como usar o ODE em conjunto com o OGRE.

Para construir uma aplicação usando estas componentes criam-se primeiro os modelos gráficos dos agentes e as suas animações básicas, como andar, correr, entre outras, utilizando o *software* de modelação 3D. Em seguida, estes modelos são exportados para um formato conhecido do motor gráfico, criando-se assim uma Base de Dados de modelos animados ou inanimados, como é o caso dos elementos estáticos, por exemplo, cadeiras, mesas, objectos de arte.

Para garantir a ligação entre as aplicações OGRE e ODE desenvolveu-se um conjunto de classes C++ designadas por camada de processamento gráfico [Silvestre05a]. Esta camada contém igualmente classes que suportam a comunicação com a componente de inteligência artificial.

3.2 Componente de Inteligência Artificial

A fim de simular os comportamentos inteligentes dos agentes adoptámos uma bancada de agentes já anteriormente utilizada no contexto de outros trabalhos desenvolvidos no LabMag: JADE (*Java Agent Development Framework*).

O JADE é uma plataforma de agentes que facilita a tarefa de criação de novos agentes, bem como toda a comunicação feita entre agentes. É uma plataforma distribuída, sendo transparente para o utilizador a máquina onde a *thread* de cada agente é executada. Deste modo podemos ter vários agentes a serem executados em máquinas distintas.

Desenvolveu-se uma camada de *software* sobre o JADE, designada por camada de processamento inteligente, de forma a estabelecer a ligação entre o JADE e a camada de processamento gráfico.

O JADE e a camada de processamento inteligente estão implementados em JAVA, enquanto o OGRE, o ODE e a camada de processamento gráfico estão implementados em C++. A solução adoptada para fazer a ligação entre as duas camadas passou pelo recurso a *sockets*. A utilização de *sockets* permite distribuir por diferentes computadores o processamento relativo a cada uma destas camadas.

Apesar de ser possível ligar cada agente JADE directamente à camada de processamento gráfico, havendo assim tantas ligações quantos os agentes existentes, optámos por criar apenas uma conexão simples entre a camada de processamento gráfico e a

camada de processamento inteligente, usando apenas uma ligação TCP/IP (Fig.2). A ideia principal é ter um agente na bancada JADE responsável pela comunicação e coordenação entre os agentes JADE e a camada de processamento gráfico.

A adopção de *sockets* para fazer a ligação entre as camadas de processamento gráfico e de processamento inteligente tornou necessário criar um protocolo de comunicação que fosse entendido por ambas as camadas. As duas camadas comunicam através de uma sequência de pedidos e respostas que respeitam o protocolo de comunicação definido. A camada de processamento gráfico efectua um pedido à camada de processamento inteligente, esta recebe o pedido, executa-o e envia a resposta à camada de processamento gráfico.

4. CONCLUSÕES E TRABALHO FUTURO

Têm sido propostas diferentes soluções para a criação de plataformas para a geração de agentes autónomos em ambientes virtuais, como referimos atrás. Nas soluções referidas grande parte da componente gráfica é construída de raiz e, na maior parte dos casos, a plataforma desenvolvida não está disponível para outros utilizadores.

O desenvolvimento de uma solução própria com base em bibliotecas gráficas elementares, apesar de ser muito mais flexível, é muito mais demorado e exigente em termos de recursos humanos. Optou-se assim por criar uma plataforma modular baseada na integração de aplicações já com algum grau de especialização. A escolha das aplicações teve em atenção o custo, a documentação acessível e a existência de grupos de utilizadores activos.

A arquitectura está aberta à adopção de um *software* de modelação distinto do escolhido o que permitirá utilizar outras ferramentas de modelação. Tendo em conta que algumas ferramentas comerciais oferecem um período de utilização gratuita ou uma versão incompleta do produto, esta hipótese é perfeitamente viável e não acarreta custos financeiros.

Até ao momento foi criado um núcleo restrito de operações que permitiu a execução de uma demonstração para exemplificar a viabilidade da arquitectura. Nesta demonstração a movimentação do agente é feita interactivamente pelo utilizador através do teclado ou do rato [Silvestre05b].

O interesse do trabalho descrito reside na sua potencial utilidade nos ambientes de investigação (LabMag) e de docência (disciplinas dos cursos do departamento de Informática da FCUL) em que se insere. Destacamos a

sua utilização na exploração de conceitos no contexto da animação comportamental (*behavioral animation*). Como exemplo podemos referir a implementação do conceito de *smart object* [Thalmann04], um objecto que encapsula uma espécie de guia de utilização que deve ser seguido pelos agentes durante a interacção com o referido objecto.

Esta plataforma pode servir também de suporte experimental para projectos a realizar no contexto das disciplinas de Inteligência Artificial ministradas no departamento e, eventualmente, para trabalhos de investigação de maior envergadura.

Além do enriquecimento das camadas do processamento gráfico e de inteligência artificial, a incorporação de um sistema de visão é um ponto chave para a continuação do trabalho. Através de um sistema de visão, um agente pode reconhecer o ambiente e os agentes que o rodeiam permitindo a interacção entre os vários intervenientes da cena. Outro aspecto a considerar é a realização de testes sobre a adequação da plataforma para a concepção de uma biblioteca de expressões faciais capazes de transmitir emoções e estados de espírito dos agentes inteligentes.

5. REFERÊNCIAS

- [Bordini02] Bordini, R.H., Bazzan, A.L.C., Jannone, R.O., Basso, D.M., Vicari, R.M., and Lesser, V.R., AgentSpeak(XL): Efficient intention selection in BDI agents via decision-theoretic task scheduling, in Castelfranchi, C., and Johnson, W.L., eds., Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2002), Bologna, Italy, pp 1294-1302, New York, NY: ACM Press, 2002
- [Boullic97] Boullic, R., Beicheraz, P., Emering, L., Thalmann, D., Integration of Motion Control Techniques for Virtual Human and Avatar Real-Time Animation, Proceedings of VRST'97, pp. 111-118, ACM Press, 1997
- [Caicedo00] Caicedo, A., Thalmann, D., Virtual Humanoids: Let Them be Autonomous without Losing Control, Proc. 3IA 2000, Limoges, France, 2000
- [Devillers02] Devillers, F., Donikian, S., Lamarche, F., Taille, J.-F., A programming environment for behavioral animation, The Journal of Visualization and Computer Animation, 13:263-274, 2002
- [Donikian98] Donikian, S., Chauffaut, A., Duval, T.Z., Kulpa, R., GASP: from Modular Programming to Distributed Execution, Computer Animation' 98, Philadelphia, USA, 1998
- [Funge99] Funge, J., Tu, X., Terzopoulos, D., Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters, SIGGRAPH 99, Los Angeles, CA, 1999
- [httpBlender] <http://www.blender3d.org/cms/Home.2.0.html>
- [httpJADE] <http://jade.tilab.com/>
- [httpLabMAg] <http://labmag.di.fc.ul.pt/>
- [httpODE] <http://ode.org>
- [httpOGRE] <http://www.ogre3d.org/>
- [Kallmann00] Kallmann, M., Monzani, J.-S., Caicedo, A., Thalmann, D., ACE: A Platform for the Real Time Simulation of Virtual Human Agents, EGAS'2000, 11th Eurographics Workshop on Animation and Simulation, Interlaken, Switzerland, 2000
- [Maciel02] Maciel, A., Nedel, L.P., Freitas, C.M., Anatomy-Based Joint Models for Virtual Humans Skeletons. In: Computer Animation 2002, 2002, Lausanne. Computer Animation, v.1 p.110 – 116, 2002
- [Monzani01] Monzani, J.-S., Caicedo, A., Thalmann, D., Integrating Behavioural Animation Techniques, Eurographics 2001, A. Chalmers and T.-M. Rhyne (eds), Computer Graphics Forum, vol 20, nº 3, 2001
- [Perlin95] Perlin, K., Real Time Responsive Animation with Personality, IEEE Transactions on Visualization and Computer Graphics, pp 5-15, vol. 1, nº1, 1995,
- [Perlin96] Perlin, K., Goldberg, A., Improv: A System for Scripting Interactive Actors in Virtual Worlds, Proceedings of SIGGRAPH'96, pp205-216, 1996
- [Rao96] Rao, A.S., AgentSpeak(L): BDI Agents speak out in a logical computable language, Proceedings MAAMAW-96, Springer Verlag, LNAI 1038, pp 42-55, 1996
- [Silvestre04] Silvestre, M., Pinto-Albuquerque, M., Carmo, M.B., Cláudio, A.P., Cunha, J.D., Coelho, H., Arquitectura de Suporte à Geração de Cenas Animadas com Agentes Inteligentes, relatório técnico do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa, DI-FCUL TR-04-7, Julho 2004
- [Silvestre05a] Silvestre, M., Pinto-Albuquerque, M., Carmo, M.B., Cláudio, A.P., Cunha, J.D., Coelho, H., Concretização de uma Arquitectura de Suporte à Geração de Cenas Animadas com Agentes Inteligentes, relatório técnico do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa, DI-FCUL TR-05-6, Março 2005
- [Silvestre05b] Silvestre, M., Pinto-Albuquerque, M., Carmo, M.B., Cláudio, A.P., Cunha, J.D., Coelho, H., Platform for the Generation of Virtual Environments Inhabited by Intelligent Virtual Humans, *student poster* ACM ITiCSE'05, Monte da Caparica, Junho 2005
- [Thalmann04] Handbook of Virtual Humans, edited by N. Magnenat-Thalmann, D. Thalmann, John Wiley & Sons, 2004
- [Torres02] Torres, J.A.R., Maciel, A., Nedel, L.P., Uma Arquitetura para Animação de Humanos Virtuais

com Raciocínio Cognitivo In: SVR 2002 – Symposium on Virtual Reality, 2002, Fortaleza - CE Proceedings, p.341-352. Fortaleza: SBC, 2002

[Torres03] Torres, J.A., Nedel, L.P., Bordini, H.R., Using the BDI Architecture to Produce Autonomous Characters in Virtual Worlds, Interactive Virtual Agents (IVA 2003), Irsee, Germany, 2003

[Tu94] Tu, X., Terzopoulos, D., Artificial fishes: Physics, locomotion, perception, behavior, Proc. ACM SIGGRAPH'94 Conference, Orlando, FL, 1994, in Computer Graphics Proceedings, Annual Conference Series, 43-50, 1994

[Vosinakis05] Vosinakis, S. and Panayiotopoulos, T., A Tool for Constructing 3D Environments with Virtual Agents, Multimedia Tools and Applications, 25, 253-279, 2005

Modelação Expedita de Ambientes Virtuais Urbanos para Utilização em Dispositivos Móveis

Maximino Bessa António Coelho João Paulo Moura José Bulas Cruz
Centro de Estudos Tecnológicos, do Ambiente e da Vida (CETAV)
Universidade de Trás-os-Montes e Alto Douro
Quinta de Prados, Vila Real
{maxbessa, acoelho, jpmoura, jcruz}@utad.pt

Fernando Nunes Ferreira
FEUP
Rua Dr. Roberto Frias, Porto
fnf@fe.up.pt

Alan Chalmers
University of Bristol
alan.chalmers@cs.bris.ac.uk

António Augusto de Sousa
FEUP / INESC Porto
Rua Dr. Roberto Frias, Porto
aas@fe.up.pt

Resumo

A evolução da tecnologia dos dispositivos de computação móveis levou ao aparecimento de um conjunto inovador de serviços e aplicações, enquanto o aumento da largura de banda das redes de comunicação sem fios permite a transmissão de uma maior quantidade de informação. Por outro lado, o aumento da precisão do GPS e a sua integração em plataformas móveis, torna estas adequadas para a prestação de serviços e aplicações baseadas no contexto do utilizador, quer em termos da sua localização geoespacial como também da tarefa que se encontre a efectuar.

Neste artigo é descrito o projecto 3D4LBMS, que tem por objectivo principal o desenvolvimento de processos expeditos para a criação de modelos tridimensionais extensos de ambientes virtuais urbanos, e a sua distribuição em dispositivos móveis através de redes sem fios, sob a forma de aplicações tridimensionais interactivas. Estas aplicações interagem com o utilizador de forma contextualizada com a sua localização geoespacial.

Palavras-chave

Modelação 3D, Realidade Virtual, Sistemas L, SIG, Percepção Visual, Guias móveis.

1. INTRODUÇÃO

A problemática da orientação e navegação em ambientes desconhecidos é uma questão que tem acoissado a humanidade, levando ao surgimento da cartografia e das ciências geográficas há vários séculos atrás. Os mapas 2D existentes são uma mais valia para tarefas de localização de um determinado local num ambiente desconhecido, graças à simplicidade do seu formato e à sua normalização. No entanto um mapa por si só não garante uma navegação eficaz uma vez que sendo uma representação abstracta do nosso mundo, requer um determinado conhecimento específico para a sua correcta interpretação.

Desde o seu nascimento, o ser humano tem uma percepção tridimensional do mundo envolvente, pelo que os mapas, sendo um suporte bidimensional, representam apenas de uma forma abstracta essa realidade. Por outro lado os mapas 3D têm a potencialidade de representar o nosso mundo de uma forma muito mais realista e tudo leva a crer que venham a ser adoptados por cada vez mais

utilizadores, não só por serem mais atractivos, mas também por serem bastante mais intuitivos [Schilling03].

Os desenvolvimentos tecnológicos da ultima década, os novos dispositivos móveis de que dispomos, as novas redes globais de telecomunicações que permitem cada vez um maior debito de dados, bem como a maior acuidade dos sistemas GPS, permitem-nos pensar em soluções móveis, tais como os guias electrónicos. Este novo tipo de tecnologia, potencia soluções de orientação de elevada precisão bem como o fornecimento de informação contextualizada com a localização do utilizador.

O projecto 3D4LBMS, que se aborda neste artigo, tem como objectivo principal a criação de modelos tridimensionais de ambientes urbanos e a sua disponibilização sob a forma de serviços móveis baseados na localização (LBMS – *Location Based Mobile Services*). No âmbito deste projecto, existem dois componentes principais: O primeiro denominado bloco de produção possibilita a construção dos ambientes virtuais urbanos de forma expedita, acedendo ao maior número possível de fontes de dados, enquanto o segundo componente, denominado

bloco de distribuição, potencia a distribuição destes ambientes virtuais ao utilizador (turistas, arquitectos, técnicos municipais, elementos das forças de segurança e de serviços de emergência).

Na secção seguinte são apresentados alguns trabalhos relacionados com o projecto 3D4LBMS. Na secção 3 é apresentado o projecto 3D4LBMS incluindo a sua arquitectura e funcionalidade básica. Na secção 4 descreve-se o processo de modelação expedito e na secção 5 o acesso interoperável às fontes de dados. Na secção 6 descreve-se a especificação de modelação XL3D e na secção 7 analisa-se a percepção visual para o desenvolvimento de metodologias de optimização dos modelos. Por último, na secção 8 faz-se a discussão dos resultados obtidos e finalmente na secção 9 são sintetizadas algumas conclusões e perspectivados alguns trabalhos futuros do projecto 3D4LBMS.

2. TRABALHOS RELACIONADOS

Os Sistemas de informação geográfica (SIG) podem fornecer um conjunto de dados de fácil análise acerca das nossas cidades. Métodos automáticos para modelação de grandes áreas urbanas já foram demonstrados por trabalhos como por exemplo [Pimentel01] e [Schilling03]. Utilizando técnicas de modelação procedimental, geram modelos geométricos extensos de edifícios e vegetação, no entanto, dado que a informação geográfica não inclui informação geométrica acerca dos aspectos visuais das entidades dos modelos gerados, não são geralmente muito detalhados ou realistas.

Um dos principais problemas para a automatização do processo de modelação é a amplificação de dados, ou seja, a capacidade de gerar novos dados a partir de um conjunto inicial reduzido, por incorporação de conhecimento. Gramáticas formais, como por exemplo os sistemas L, têm sido utilizadas em variadas aplicações no âmbito da computação gráfica, com maior relevância na modelação de fenómenos naturais e organismos [Deussen98], [Lane02] e [Prusinkiewicz01]. Adicionalmente, têm surgido alguns trabalhos aplicando os sistemas L à modelação expedita de ambientes urbanos extensos. Em [Parish01], são utilizados Sistemas L estendidos para derivar ruas e lotes de edifícios, e sistemas L paramétricos e estocásticos para gerar a geometria dos edifícios. Uma abordagem diferente é apresentada por Wonka [Wonka03], onde uma *split grammar* foi desenvolvida, com suporte num conjunto de regras baseadas no conceito de forma. Embora restringida à modelação automática de estruturas arquitectónicas, esta abordagem simplifica o processamento das coordenadas ao integrar o conceito de forma nas regras de produção. Outras abordagens como [Polys03] utilizam simplesmente uma folha de estilo XSLT de forma a converterem um documento XML existente para o formato X3D, sendo por isso uma solução menos flexível.

Questões como a modularidade e a reutilização são críticas para que o processo de especificação da modelação seja célere e eficiente. O projecto CONTIGRA [Dachsel02] é claramente um bom exemplo de como a utiliza-

ção de XML para a especificação de aplicações gráficas 3D torna possível a integração e reutilização de componentes no processo do desenvolvimento de aplicações em X3D.

Apesar de condicionamentos como a largura de banda ou a reduzida dimensão dos ecrãs, têm sido desenvolvidas algumas aplicações comerciais que nos fornecem um conjunto de funcionalidades bastantes úteis para auxílio à navegação [CoPilote] [NavMan]. No entanto são protótipos de projectos como o TellMaris [Schilling03] e o SmartKom [Wahlster02] que nos fazem vislumbrar o futuro dos guias móveis onde mapas digitais 3D são utilizados em conjunto com uma série de novas interfaces, de forma a proporcionar ao utilizador uma experiência mais natural e intuitiva.

No projecto TellMaris é fornecido um mapa tridimensional da cidade de Tonsberg (na Noruega), que é utilizado por turistas para os ajudar a encontrar os pontos de interesse da cidade, tais como atracções turísticas, restaurantes, etc. O protótipo TellMarisGuide é baseado no *plugin* do Pocket Cortona [PocketCortona], que foi especialmente concebido para plataformas móveis. Este protótipo combina mapas tridimensionais com mapas bidimensionais, sendo possível o seu intercâmbio através dos comandos disponibilizados pela sua interface gráfica.

Noutro projecto denominado Lol@ (local location assistant) [Anegg02] foi desenvolvido um protótipo de um serviço baseado na localização para UMTS (Universal Mobile Telecommunications System). Este protótipo foi projectado para fornecer instruções aos turistas que visitam a cidade de Viena de Áustria acerca dos pontos de interesse existentes. Adicionalmente, são fornecidas instruções para que o turista se desloque entre dois determinados locais, proporcionando ao longo do percurso informação multimédia acerca das atracções turísticas.

Já no projecto SmartKom está a ser desenvolvida uma *framework* centrada no aspecto da interacção multimodal, utilizando gestos, a fala, e a mímica. Está também a ser desenvolvido um guia móvel que oferece serviços de navegação e informação utilizando GSM/UMTS para comunicação, e GPS para localização. A apresentação da informação é efectuada através de mapas, linguagem natural e de um agente de apresentação antropomórfico.

3. O PROJECTO 3D4LBMS

O projecto 3D4LBMS surge com o objectivo principal do desenvolvimento de um conjunto de processos expeditos para a criação de modelos tridimensionais extensos de ambientes virtuais urbanos, e a sua disponibilização como serviços móveis baseados na localização, através de redes de comunicação sem fios.

Este novo tipo de serviços pode ser amplamente adoptado não só para guias electrónicos para pedestres mas também em soluções desde o turismo virtual, a manutenção de infra-estruturas, ou planeamento urbano, entre outros.

A modelação de ambientes virtuais extensos, requer normalmente uma grande intervenção humana. Por outro lado, existe uma grande quantidade de informação sobre

os ambientes reais a modelar que provém de diversas fontes de informação, tais como SIG, bases de dados relacionais, documentos ou imagens. Toda esta informação tem grande potencial para a modelação tridimensional de ambientes virtuais, através de processos automáticos.

Um dos pontos-chave do sistema 3D4LBMS é a sua capacidade de prototipagem rápida de ambientes virtuais extensos e a possibilidade do seu refinamento progressivo de forma a atingir um maior nível de realismo.

Estes ambientes virtuais permitirão a diferentes tipos de utilizador, interagir com diferentes classes de informação através de apresentações visuais diversas, consoante o seu perfil.

3.1 Arquitectura

Uma das preocupações aquando da concepção do sistema 3D4LBMS, é que este fosse baseado numa arquitectura distribuída, com base na tecnologia dos serviços web, e que potenciase a interoperabilidade entre os diversos módulos que o compõem, ou com aplicações externas.

O sistema foi assim desenvolvido segundo dois blocos principais, tal como observado na figura 1: um bloco para a produção de modelos tridimensionais de ambientes urbanos, e um segundo bloco para a sua disponibilização de serviços em dispositivos móveis, através de redes de comunicação sem fios.

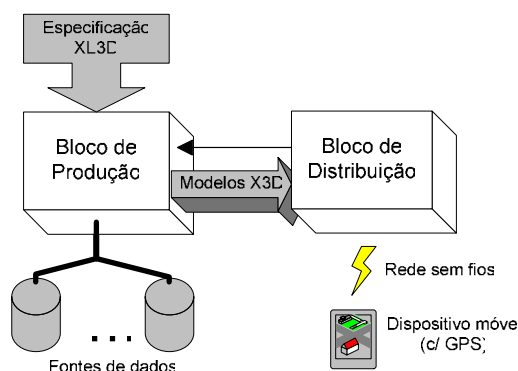


figura 1 – Arquitectura do sistema 3D4LBMS

O bloco de produção recebe uma especificação XL3D [Coelho03a], descrevendo os processos de modelação, bem como as fontes de dados a utilizar para determinados modelos. Após invocação pelo bloco de distribuição, o bloco de produção gera automaticamente os modelos 3D especificados, em formato X3D.

O bloco de distribuição recebe os modelos tridimensionais de ambientes urbanos e adapta-os para sua disponibilização como serviços móveis baseados na localização, através de redes sem fios.

3.2 Bloco de produção

O bloco de produção tem como núcleo principal o modelador XL3D [Coelho05]. Este modelador automático reage em função de uma especificação de modelação, gerando modelos de cenas urbanas no formato X3D, que são

armazenados numa base de dados de modelos. Esta base de dados é posteriormente acedida pelo bloco de distribuição para compor a informação a ser enviada para os dispositivos móveis, de acordo com a sua localização e perfil dos utilizadores.

Como podemos observar na figura 2, o utilizador cria uma especificação de modelação utilizando um editor de documentos XL3D. Este documento é depois utilizado para invocar o modelador XL3D, que encontrando-se implementado como serviço Web, gera automaticamente os modelos urbanos.

Este processo poderá ser realizado sem a intervenção do utilizador quando, por exemplo, um servidor de LBMS recebe um pedido para que seja gerado um novo modelo. Neste caso o servidor de LBMS gera automaticamente um documento com uma especificação XL3D e invoca o sistema de modelação automático, que por sua vez irá gerar o modelo tridimensional pretendido. Esta flexibilidade potencia a integração do sistema de modelação num conjunto alargado de aplicações distribuídas.

Os dados a serem utilizados nos processos de modelação podem ser importados a partir de directórios de ficheiros, bases de dados ou a partir de serviços Web. No entanto todas estas fontes de dados devem ter um denominador comum, que é o formato XML. Utilizando folhas de estilo XSLT transforma-se toda a informação em cadeias modulares paramétricas, que são posteriormente concatenadas de forma a compor a cadeia inicial de dados a ser utilizada no processo de modelação.

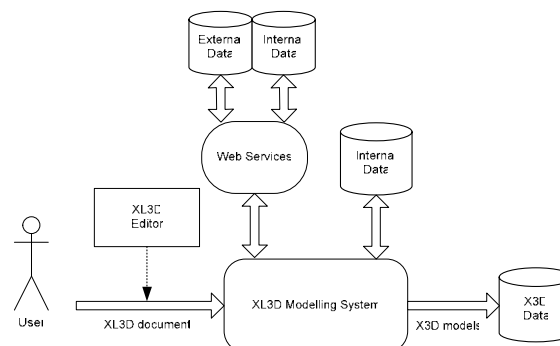


figura 2 -Módulos do processo de produção.

Cada processo de modelação é definido e controlado a partir de um conjunto de regras de produção utilizando sistemas L. A cadeia inicial é então transformada através de um processo iterativo até que uma solução satisfatória seja obtida. A cadeia final é interpretada para gerar uma cena X3D que representa o modelo tridimensional especificado pelo utilizador.

O modelador XL3D foi desenvolvido como um serviço Web, baseado na tecnologia .Net [DotNet]. Após a recepção da especificação de modelação é criado um conjunto de objectos que permite representar correctamente todas as regras de produção, protótipos e fontes de dados, incrementando o nível de eficiência e velocidade da geração de soluções de modelação.

3.3 Bloco de Distribuição

O desenvolvimento do bloco de distribuição baseou-se numa arquitectura cliente-servidor, tal com se observa na figura 3. Este componente encontra-se actualmente em desenvolvimento.

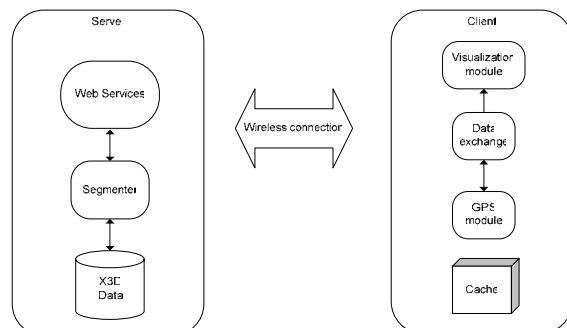


figura 3 – Bloco de distribuição

O dispositivo cliente (PDA, SmartPhone, etc) acede ao servidor, onde é realizada uma autenticação do serviço, e após a qual é-lhe enviada a sua localização (coordenadas obtidas por GPS). Por sua vez, o servidor responde com o modelo tridimensional da zona do modelo da cidade envolvente à localização do utilizador.

Para que a informação enviada não seja nem precária nem excessiva, está a ser desenvolvido um segmentador que irá ter em conta não só a localização do cliente mas também outros factores importantes como: geografia e topologia do modelo, a resolução do ecrã do dispositivo, a largura de banda disponível, ou o perfil do utilizador.

4. PROCESSO DE MODELAÇÃO EXPEDITO

O processo de modelação expedita do modelador XL3D [Coelho03b] baseia-se nos sistemas L (acrónimo de sistemas Lindenmayer) [Lindenmayer68] paramétricos, sensíveis ao contexto e estocásticos.

O conceito fundamental dos sistemas L é o de reescrita, ou seja, uma técnica que possibilita a definição de objectos complexos através da sucessiva substituição de partes de um objecto inicial utilizando um conjunto de regras de reescrita ou produção. Outro dos conceitos fundamentais no contexto dos sistemas L, é o de módulo, designando uma unidade construtiva discreta que é repetida à medida que o sistema se desenvolve. Exemplo concreto deste conceito é o dos ramos e das folhas no caso da modelação de uma árvore. No caso dos sistemas paramétricos cada módulo pode ter um conjunto de parâmetros associados.

No caso mais simples, os sistemas L de contexto livre, cada regra de produção substitui um módulo predecessor, por zero, um, ou mais módulos sucessores. No caso dos sistemas sensíveis ao contexto, a aplicabilidade de uma determinada regra de produção não depende apenas do módulo predecessor, mas também da sua vizinhança, ou seja, dos módulos posicionados à sua esquerda ou à sua direita na cadeia modular.

Um sistema L é definido a partir de um axioma, ou seja a cadeia de módulos inicial, e um conjunto de regras de produção que definem como o sistema irá evoluir ao longo da sequência de desenvolvimento. As regras de produção são aplicadas em paralelo, sendo reescritos todos os módulos simultaneamente em cada passo de derivação. A sequência de estruturas obtida em passos de derivação consecutivos a partir de uma estrutura inicial predefinida (axioma) denomina-se sequência de desenvolvimento.

Um sistema L desenvolve um determinado axioma, produzindo uma cadeia modular final que representa um determinado organismo ou entidade. Ao nível da Computação Gráfica, esta cadeia final terá que ser interpretação de forma a gerar a sua representação gráfica.

Tendo em conta a complexidade do processo de modelação, é desejado que exista a capacidade de dedução de novos dados. Estes podem ser deduzidos a partir dos sistemas L paramétricos, já que conferem uma grande capacidade de amplificação de dados às regras de produção definidas pelo utilizador. Por outro lado, os sistemas L estocásticos permitem a incorporação de aleatoriedade nos processos de modelação, de forma a simular a diversidade dos ambientes urbanos.

O processo de modelação funciona com o seguinte algoritmo:

1. Todas as cadeias modulares contidas nas fontes de dados do projecto são geradas através da transformação dos dados originais através de uma folha de estilo XSLT.
2. Cada modelo do projecto XL3D é modelado para um determinado documento X3D.
3. Cada um destes modelos é obtido por agrupamento de todas as entidades que o compõem.
4. O modelo de cada entidade é obtido por instanciação de procedimentos de modelação ou então inserindo modelos X3D. Adicionalmente, se uma entidade possui uma subentidade então o modelo correspondente também é agrupado ao modelo gerado.
5. A instanciação de um processo de modelação é conseguida através da construção de um axioma inicial obtido a partir de uma ou mais cadeias modulares contidas nas fontes de dados.
6. O processo de modelação é instanciado com um conjunto de parâmetros, um grupo de protótipos e o axioma inicial de forma a gerar o sistema L que levará ao processo de modelação da entidade.
7. A sequência de desenvolvimento do sistema L produz a cadeia final que representa o modelo da entidade, após um determinado número máximo de iterações, ou quando mais nenhuma regra de produção for aplicável.
8. Finalmente a interpretação da cadeia final resultante do passo anterior, pelo conjunto de protó-

tipos especificado, gera o modelo final no formato X3D.

Este algoritmo possibilita assim a geração automática de modelos tridimensionais de ambientes urbanos com base apenas na especificação de modelos num documento XL3D.

5. ACESSO INTEROPERÁVEL ÀS FONTES DE DADOS

Os dados utilizados para a modelação expedita de ambientes urbanos extensos estão disponíveis a partir de várias fontes de dados em diversos formatos e provenientes de diferentes plataformas. A integração de todos estes dados é uma tarefa complexa, originando problemas de interoperabilidade.

Sendo a informação geográfica provavelmente o tipo de informação com maior relevância para a modelação de ambientes urbanos, a sua integração num projecto de modelação motiva o recurso a padrões geoespaciais como garante da sua interoperabilidade. O consorcio Open Geospatial [OGC] é uma organização que se dedica ao problema da interoperabilidade na área dos sistemas de informação geográfica, e lançou algumas especificações baseadas em XML, tais como o GML (*Geography Markup Language*) [GML] e o WFS (*Web Feature Service*) [WFS]. Estas especificações têm vindo a ser adoptados pelos produtores de software, e a sua integração nos processos de modelação é de grande importância para o desenvolvimento de soluções automáticas.

Na área da computação gráfica o problema da interoperabilidade foi abordado pelo consórcio Web3D [Web3D] através do desenvolvimento do formato X3D [X3D]. O X3D é uma evolução do VRML (*Virtual Reality Modeling Language*), permitindo a especificação de ambientes virtuais e aplicações 3D em XML.

O modelador XL3D possibilita a utilização de fontes de dados que referenciem qualquer tipo de documento XML, potenciando assim a utilização de padrões, tais como os geoespaciais ou de computação gráfica, entre uma enorme diversidade de possibilidades. Estes documentos podem ser acessíveis num ficheiro, contido numa determinada directoria numa máquina local ou em rede, ou por invocação de um serviço Web. Neste último caso é possível aceder assim a fontes de dados dinâmicas, que retornam dados actualizados a cada invocação do serviço.

6. A ESPECIFICAÇÃO DE MODELAÇÃO XL3D

Todo o processo de modelação é especificado de uma forma declarativa através de um documento. Dada a especificidade dos processos de modelação e o problema da interoperabilidade foi necessário desenvolver um XML-schema próprio denominado XL3D [Coelho03a].

A estrutura básica de um documento XL3D é hierárquica e o nó principal é o projecto XL3D (figura 4). Um projecto XL3D é composto por um cabeçalho com informação acerca do autor do projecto, um conjunto de modelos descrevendo a estrutura hierárquica das entidades que

descrevem o ambiente urbano, procedimentos de modelação que incorporam conhecimento relevante para o processo de modelação, protótipos que interpretam os sistemas L, e fontes de dados.

Cada modelo é composto por entidades, reflectindo o carácter complexo e estruturado de um ambiente urbano, de acordo com a percepção subjectiva do utilizador. Cada entidade (figura 5) instancia procedimentos de modelação para gerar modelos tridimensionais a partir de uma cadeia de módulos que são obtidos a partir de várias fontes de informação.

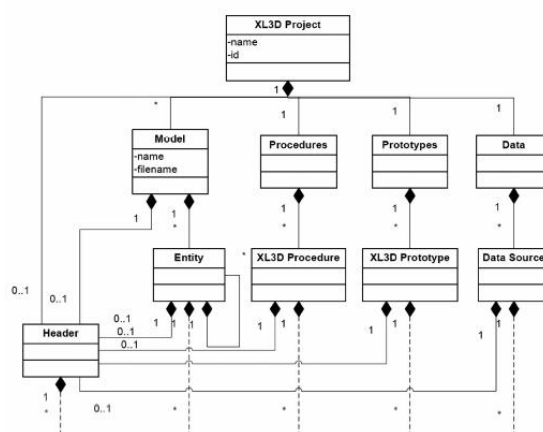


figura 4 – Projecto XL3D

Cada entidade pode ser composta por outras entidades cujo modelo pode ser colocado em qualquer posição específica do nó entidade e pode ser submetido a uma transformação de coordenadas.

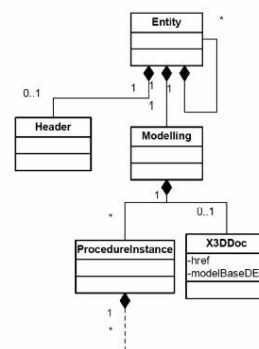


figura 5 – Entidade

Cada entidade instancia um conjunto de processos de modelação, que produz o modelo tridimensional a partir das cadeias de módulos obtidas com base nas diversas fontes de dados especificadas.

Os procedimentos de modelação (figura 6) são o componente nuclear do processo de modelação, baseando-se em sistemas L para modelar partes específicas de cada entidade. Os procedimentos de modelação são definidos como componentes reutilizáveis que devem ser instancia-

dos para a modelação das diversas entidades, ou estendidos para a criação de novos procedimentos de modelação.

O tipo de sistemas L utilizado é paramétrico, sensível ao contexto e estocástico. Com base na cadeia modular inicial, o sistema L produz outra cadeia modular representando a estrutura conceptual do modelo tridimensional pretendido. Esta cadeia modular é então interpretada de forma a criar um grafo de cena em X3D, através da instanciação de uma série de protótipos.

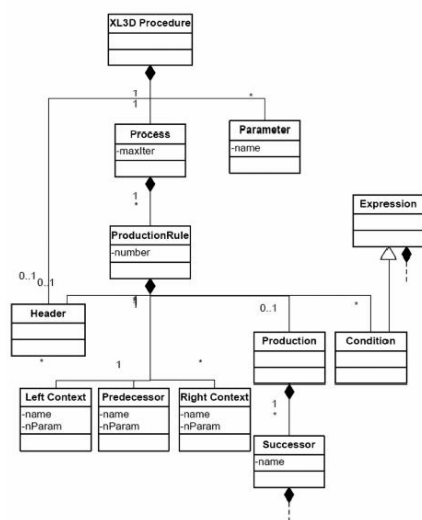


figura 6 - Procedimentos de modelação

As fontes de dados contêm referências para dados guardados em documentos XML, que se encontram em directórios locais ou são obtidas por invocação de um serviço web. Esta informação é então transformada, do seu formato original para uma cadeia de módulos, através da aplicação de uma folha de estilo XSLT.

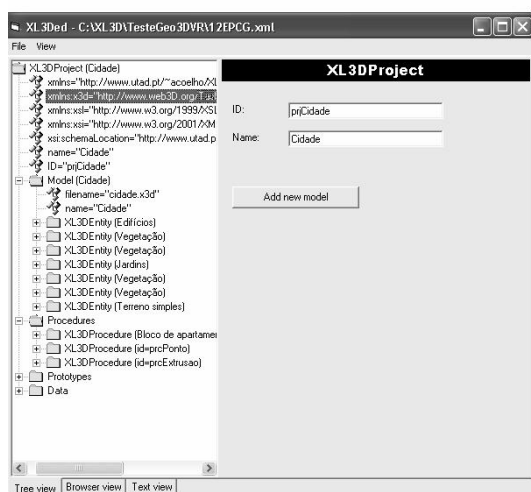


figura 7 – Aplicação para concepção das especificações de modelação XL3D

Para que a criação da especificação de modelação XL3D não se tornasse numa tarefa demasiadamente morosa e complexa foi desenvolvida uma ferramenta gráfica (Editor XL3D, figura 7) que assiste a criação de documentos XL3D.

7. A PERCEPÇÃO COMO METODOLOGIA PARA OPTIMIZAÇÃO DO MODELO

Os modelos tridimensionais extensos não são os mais adequados para visualização em dispositivos móveis. Existem várias restrições a ultrapassar, tais como o baixo poder de cálculo (em comparação com PC), a reduzida dimensão dos ecrãs, a largura de banda das comunicações sem fios, etc.

Está a ser desenvolvida uma metodologia de forma a minimizar alguns dos problemas referidos. Se soubermos exactamente para onde as pessoas estão a olhar a cada instante, é possível recriar esses objectos em alta resolução mantendo os restantes em baixa resolução, por um custo computacional muito menor, e sem que o utilizador se aperceba dessa perda de qualidade.

A forma como reconhecemos e identificamos o ambiente que nos rodeia depende de quem somos e da tarefa que estamos a executar em determinado momento, nesse ambiente. A atenção visual é o processo pelo qual o ser humano selecciona um pedaço da informação visual disponível para a localização, identificação e percepção dos objectos presentes no ambiente que o rodeia. Este processo permite ao nosso sistema visual processar os estímulos visuais que recebemos, com base na mudança da atenção em cada imagem, prestando assim mais atenção às regiões mais salientes, e menos atenção a regiões menos salientes. Apesar do sistema visual humano ser bastante preciso, não é perfeito, e assim quando a nossa atenção não está focada em determinados itens duma determinada cena, estes podem passar completamente despercebidos. Este fenómeno tem por nome *Inattention Blindness* [Mack98]. Nos últimos anos, a utilização do conhecimento do sistema visual humano tem vindo a ser cada vez mais usado para melhorar a qualidade das imagens criadas em diversos trabalhos [Cater03] [Greenberg97] [MacNamara00] [Myszkowski01] [Wahlster02].

Um recente estudo [Bessa04] foi efectuado para descobrir quais seriam os pontos de interesse utilizados por qualquer pessoa, na identificação do local e da direcção exacta de onde uma determinada fotografia tinha sido tirada. Este conhecimento poderá ser utilizado de forma a prever quais seriam os objectos em que o utilizador focará a sua atenção.

Os resultados apontam para que a selecção dos objectos a utilizar sejam determinados tendo em conta a sua proximidade à pessoa e o seu tamanho. Existem no entanto alguns tipos de elementos característicos que normalmente são utilizados para obter uma boa localização, tais como o mobiliário urbano, a publicidade ou determinados pormenores dos edifícios, como as varandas.

Estes resultados são de enorme relevância num projecto como o 3D4LBMS, para a optimização da utilização dos recursos dos dispositivos móveis e das redes de comunicação.

8. DISCUSSÃO

A modelação de ambientes virtuais urbanos extensos é uma tarefa árdua e complexa, e o modelador XL3D permite automatizar este processo, através da definição do processo de modelação de forma declarativa, numa única especificação XL3D. Esta abordagem tem grandes vantagens relativamente aos modeladores convencionais, potenciando a sua integração em sistemas distribuídos.

A escalabilidade é um factor importante para a criação de modelos extensos e complexos de um ambiente virtual urbano. A especificação XL3D é por natureza escalável, pois novas entidades representando novos objectos podem ser utilizadas para refinar o modelo. Adicionalmente, os procedimentos podem ser estendidos de forma a produzirem objectos mais detalhados e realistas, por adição de regras de produção suplementares ao processo de modelação.

A reutilização é também tida em conta em toda a filosofia do processo de modelação, já que o mesmo conjunto de regras pode ser utilizado com um novo grupo de dados, como por exemplo os respeitantes a outra cidade. Sendo os procedimentos de modelação paramétricos podem também ser utilizados, sem qualquer modificação, para modelar diferentes entidades com semelhantes processos de desenvolvimento.

Já que a maioria dos ambientes urbanos partilham uma estrutura similar, uma única especificação de modelação pode ser utilizada em diferentes casos sem qualquer modificação, exceptuando obviamente as fontes de dados.

A fidelidade visual dos modelos é crucial em algumas aplicações, e por isso os processos de modelação podem ser definidos para gerar modelos altamente detalhados, tudo dependendo da quantidade de dados utilizada no processo de modelação.

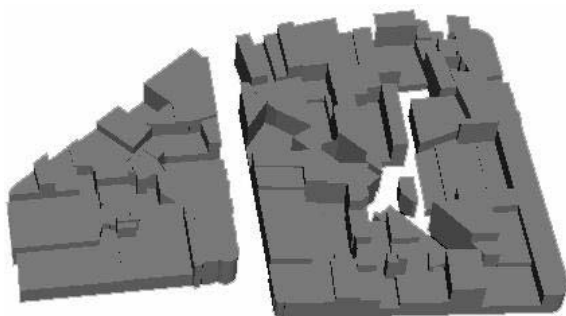


figura 8 – Vista de uma cena gerada automaticamente, onde apenas são aplicadas pequenas extrusões aos contornos dos edifícios.

Devido às características intrínsecas do processo de modelação baseado em sistemas L, o modelador XL3D é capaz de produzir cenas muito detalhadas. No entanto,

para se obter modelos mais realistas é necessário possuir elevada quantidade e qualidade de dados.

A performance dos sistemas LBMS é condicionada pela memória interna dos dispositivos móveis e a largura de banda das redes de comunicação sem fios. Para que seja possível disponibilizar modelos tridimensionais urbanos é necessário algum desenvolvimento de forma a transmitir apenas a informação relevante ao utilizador, com base na sua localização em relevância dos objectos. Este último depende da percepção do ambiente que rodeia cada utilizador.

O processo de modelação é um processo incremental onde a partir de um pequeno conjunto de regras é possível obter uma rápida representação dos dados (figura 8). No entanto, para obter um modelo mais detalhado são necessárias mais regras de produção de forma a incorporar detalhes como as fachadas dos edifícios, as portas as janelas ou as varandas (figura 9), assim como outros elementos como estradas, vegetação ou mobiliário urbano (figura 10).

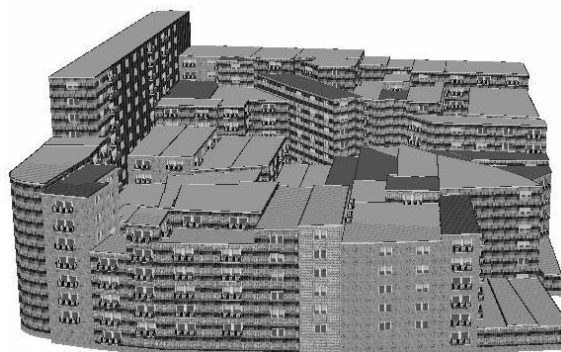


figura 9 - Vista de um quarteirão de uma cena gerada automaticamente.



figura 10 – Vista de pormenor de uma cena gerada automaticamente.

Todos os aspectos mencionados anteriormente como a escalabilidade, reutilização e fidelidade visual fazem do modelador XL3D uma ferramenta versátil que permite a modelação de ambientes 3D extensos de forma simples, eficaz e, sobretudo, expedita.

9. CONCLUSÕES E TRABALHO FUTURO

O projecto 3D4LBMS foi criado com o intuito de desenvolver a tecnologia necessária para criar ambientes virtuais tridimensionais e disponibilizá-los em aplicações LBMS inovadoras. Este projecto tem claramente dois objectivos distintos: a modelação expedita dos modelos tridimensionais, e a sua distribuição para dispositivos móveis

O sistema de modelação apresentado neste artigo está directamente relacionado com a resolução do problema da modelação de ambientes virtuais urbanos extensos. Implementado como um serviço Web, permite uma fácil integração numa arquitectura distribuída permitindo que os modelos sejam distribuídos aos dispositivos móveis utilizando redes sem fios.

A automatização oferecida pelo sistema de modelação XL3D foi conseguida através da interoperabilidade no acesso a diversos tipos de dados, e a utilização de processos de modelação baseados em sistemas L. Os sistemas L possuem um elevado poder de amplificação de dados, fazendo emergir novas propriedades não presentes nos dados iniciais. Estas características fornecem ao sistema de modelação XL3D a aptidão de gerar modelos detalhados, a partir de um conjunto de dados abstractos, como os provenientes de SIG.

Todo o processo de modelação é especificado de forma declarativa, através de documentos XL3D. O sistema de modelação pode ser utilizado com vantagens claras para criar soluções iniciais de forma expedita. O sistema pode também ser empregue para a geração de modelos com grande fidelidade visual e detalhe, de ambientes onde exista uma grande quantidade de informação (geográfica ou não) disponível. Pode, no entanto também ser utilizado em ambientes onde não existe muita informação, gerando modelos tridimensionais com menor nível de fidelidade visual, mas mantendo o nível de detalhe.

Se soubermos exactamente quais os pontos-chave usados pelos utilizadores dos guias móveis quando estão a tentar localizar determinado sítio, podemos utilizar essa informação para a da definição dos processos de modelação e das regras de produção, dando mais detalhe e realismo onde mais interessa, utilizando metodologias baseadas na percepção visual. Por forma a melhorar os modelos de ambientes virtuais urbanos gerados, irão ser desenvolvidos mais testes no terreno, onde se irá realizar uma comparação entre os modelos virtuais construídos sem ter em conta a percepção visual, e outros onde esta foi utilizada para a sua optimização. Estes resultados irão permitir concluir se este tipo de abordagem pode trazer mais qualidade aumentando o desempenho global do modelo.

Como trabalho futuro aponta-se a necessidade de melhorar as ferramentas existentes para edição de documentos XL3D, através do desenvolvimento de uma interface mais intuitiva para a definição das regras de produção. O sistema de modelação XL3D é uma ferramenta complexa que realiza várias iterações até que o resultado final seja obtido. Como um ambiente urbano é normalmente exten-

so, o tempo de processamento requerido para que o modelo seja gerado é considerável a implementação do modelador numa arquitectura paralela através de um cluster de PC's poderá possibilitar uma resposta mais rápida a outras aplicações em sistemas distribuídos.

10. AGRADECIMENTOS

Este trabalho foi financiado parcialmente pelo POSI, a União Europeia e o FEDER através do projecto POSI/CHS/48220/2002 denominado "3D4LBMS – Modelação Tridimensional de Ambientes Virtuais Urbanos para Serviços Móveis Baseados na Localização".

Um agradecimento muito especial para toda a equipa do projecto 3D4LBMS, pelo empenhamento e dedicação que têm devotado ao projecto.

11. REFERÊNCIAS

- [Anegg02] Anegg, H.; Kunczler, H.; Michlmayr, E.; Pospischil, G. and Umlauf, M.: LoL@: designing a location based UMTS application. *Elektrotechnik und Informationstechnik*, 119(2):48–51, 2002.
- [Bessa04] Bessa, M.; Coelho A. and Chalmers, A.: Alternate Feature Location for Rapid Navigation using a 3D Map on a Mobile Device. In proceedings MUM 2004, College Park, Maryland, Pages: 5 – 9, ACM 2004.
- [Cater03] Cater, K.; Chalmers, A. and WARD, G.: Detail to attention: Exploiting visual tasks for selective rendering. In Proceedings of the 14th Eurographics workshop on Rendering, Leuven, Belgium, pp 270 – 280, 2003.
- [Coelho03a] Coelho A., Sousa A., Ferreira F.: 3D Modelling of Large Urban Scenes from Diverse Sources of Information. Proceedings of the 7th ICCG/IFIP International Conference on Electronic Publishing, pp. 278-287, 2003.
- [Coelho03b] Coelho, A.; Sousa A.; Ferreira F.: Modelação expedita de cenas urbanas. Proceedings of 12 encontro Português de Computação Gráfica, ISEP, Porto, 2003.
- [Coelho05] Coelho A.; Sousa A.; Ferreira F.: Modelling Urban Scenes for LBMS, Proceedings of Web3D 2005 Symposium, University of Wales, Bangor, UK, 2005.
- [CoPilote] CoPilote, <http://www.alk.eu.com/>
- [Dachselt02] Dachselt R.; Hinz, M.; Meiner K.: CONTIGRA: An XML-Based Architecture for Component-Oriented 3D Applications, In Proceedings of Web3D '02, pp. 155-163, 2002.
- [Deussen98] Deussen, O.; Hanrahan P.; B. Lintermann, Rümeh, Pharr, M. and PruRUSINKIEWICZ P.: Realistic modeling and rendering of plant ecosystems. *Computer Graphics 32*, (Annual Conference Series), 275–286, 1998.
- [DotNet] .Net, <http://www.microsoft.com/net/>
- [Greenberg97] Greenberg, D.; Torrance, K.; Shirley, P.; Arvo, J.; Ferwerda, J.; Pattanaik, S.; Lafortune, A.;

- Walter, B.; Foo, S. and Trumbore, B.: A framework for realistic image synthesis. In Proceedings of SIGGRAPH 1997(Special Session), pages 477{494. ACM, 1997.
- [Lane02] Lane, B. and Prusinkiewicz, P.: Generating spatial distributions for multilevel models of plant communities. In Proceedings of Graphics Interface, 69–80, 2002.
- [Mack98] Mack, A. and Rock, I.: Inattentional Blindness. Massachusetts Institute of Technology Press, 1998.
- [Mcnamara00] Mcnamara, A.; Chalmers A.; Troschianko T. and Gilchrist, I.: Comparing real and synthetic scenes using human judgements of lightness. In 12th Eurographics Workshop on Rendering, pages 207{219, 2000.
- [Myszkowski 01] Myszkowski, K.; Tawara, T.; Akamine, H. and Seidel H.: Perception-guided global illumination solution for animation rendering. In Proceedings of SIGGRAPH 2001, pages 221 {230. ACM, 2001.
- [NavMan] NavMan, <http://www.navman-europe.com/>
- [OGC] OGC. The open geospatial consortium, inc. <http://www.opengeospatial.org/>.
- [Parish01] Parish Y, Muller P.: Procedural Modeling of Cities. Proceedings of SIGGRAPH 2001, pp. 301-308, Los Angeles, 2001.
- [Pimentel01] Pimentel J., Baptista N., Goes L., Dionísio J.: Construção e gestão da complexidade de cenários urbanos 3D em ambientes virtuais imersivos. Actas do 10º Encontro Português de Computação Gráfica, pp. 165-174, 2001.
- [PocketCortona] Pocket Cortona
<http://www.parallelgraphics.com/products/cortona/>
- [Polys03] Polys N. F.: Stylesheet transformations for interactive visualization: towards a web3d chemistry curricula. In Web3D '03 Proceeding of the eighth international conference on 3D Web technology, ACM Press, 85–ff.
- [Prusinkiewicz90] Prusinkiewicz P., Lindenmayer A.: The Algorithmic Beauty of Plants, Springer, 1990.
- [Prusinkiewicz01] Prusinkiewicz P., Mündermann L., Karwowski R., and Lane B.: The use of positional information in the modeling of plants. In ACM Computer Graphics (Proceedings of SIGGRAPH 2001), 289–300, 2001.
- [Ramasubramanian99] Ramasubramanian M.; Pattanaik, and Greenberg D.: A perceptually based physical error metric for realistic image synthesis. In Proceedings of SIGGRAPH 1999, pages 73 {82. ACM, 1999.
- [Schilling03] Schilling, Arne and Coors, Volker (2003): 3D Maps on Mobile Devices. Workshop "Design kartenbasierter mobiler Dienste", 08.09.2003, Stuttgart, Germany.
- [Taylor92] Taylor, C.: "Fleshing out" Artificial Life II. In C. G. Langton, C. Taylor, J. D. Farmer and S. Rasmussen, editors, Artificial Life II, pp. 25-38, Addison Wesley, Redwood City, 1992.
- [Wahlster02] Wahlster W.: Smartkom: Fusion and fission of speech, gestures, and facial expressions. In Proceedings of the 1st International Workshop on Man-Machine Symbiotic Systems, pages 213–225, Kyoto, Japan, 2002.
- [Web3D] The Web3D Consortium.
<http://www.web3d.org/>
- [WFS] Web Feature Service
<http://www.opengis.org/techno/RFC13.pdf> [X3D] Web3D Consortium: "X3D: The Virtual Reality Modelling. Language – International Standard ISO/IEC 14772:200x".
<http://www.web3d.org/TaskGroups/x3d/specification/>
- [Wonka03] Wonka P.; Wimmer M.; Sillion F. and Ribarsky, W.: Instant architecture. ACM Trans. Graph. 22, 3, 669–677, 2003.
- [XML] XML - Extensible markup Language. <<http://www.w3.org/XML/>>
- [XML-Schema] XML-Schema.
<<http://www.w3.org/XML/Schema>>
- [XSLT] XSLT- Extended Stylesheet Language Tranformations (XSLT).
<<http://www.w3.org/TR/xslt11/>>

Visualização de Dados Arqueológicos: Um Exemplo de Utilização do *Visualization Toolkit*

Paulo Bernardes
Unidade de Arqueologia, U. Minho
Braga
pbernardes@uaum.uminho.pt

Joaquim Madeira
DET / IEETA – U. Aveiro
Aveiro
jmadeira@det.ua.pt

Sumário

A Arqueologia tem como principal objectivo reconstruir todo o processo que conduziu à formação do registo arqueológico. Ou seja, pretende-se compreender como foram erguidos edifícios e estruturas, e como foram destruídos e cobertos por sucessivas camadas de terra. Entender os mecanismos de formação de um arqueosítio é fundamental para a sua interpretação. Modelos 3D de reconstituição de edifícios são uma importante ferramenta para assistir o arqueólogo nesta tarefa.

Com este trabalho pretende-se mostrar que a aplicação de técnicas actuais de Visualização a dados arqueológicos é uma mais-valia importante na prossecução desses objectivos. Para isso desenvolveu-se um primeiro protótipo de aplicação, com base no VTK, que importa dados arqueológicos e modelos reconstruídos, e permite a sua visualização interactiva. É possível realizar um conjunto de operações sobre os modelos visualizados, com o objectivo de proporcionar aos arqueólogos diferentes tipos de leitura de um arqueosítio.

Palavras-Chave

Arqueologia, Visualização, Modelos Virtuais.

1. INTRODUÇÃO

A Arqueologia tem como principal objectivo reconstruir todo o processo que conduziu à formação do registo arqueológico [Martinez92]. Para a correcta interpretação de qualquer arqueosítio (espaço onde decorreu ou está a decorrer uma intervenção arqueológica) é fundamental entender os seus mecanismos de formação, i.e., compreender como foram erguidos edifícios e estruturas, e como foram destruídos e cobertos por sucessivas camadas de terra. A utilização de modelos 3D e a aplicação de técnicas apropriadas da Visualização de Dados proporcionam aos arqueólogos a realização de diferentes tipos de leitura de um arqueosítio, contribuindo para o enriquecimento da sua interpretação.

O objectivo principal deste artigo é apresentar um primeiro protótipo de uma aplicação para visualizar dados arqueológicos e modelos reconstruídos, com base no *Visualization Toolkit* (VTK). É possível realizar sobre os modelos visualizados quer as operações mais comuns (p.ex., *zoom*, *pan* e *rotate*), quer seleccionar, remover e repor alguns elementos que os constituem.

Antes porém, para entender a relação entre a Arqueologia, a Computação Gráfica e a Visualização de Dados, faz-se uma breve abordagem histórica à aplicação destas à Arqueologia, desde os anos 70 até à actualidade. Ainda nessa secção, apresentam-se algumas correntes de pensamento para o futuro da Arqueologia Virtual.

Depois, faz-se uma apresentação sucinta de alguns Sistemas de Visualização 3D mais divulgados. Esta tem como objectivo principal mostrar as suas diferentes valências, e reforçar a motivação para a escolha do VTK como suporte do protótipo desenvolvido. Para além do VTK, os sistemas apresentados são o *Advanced Visual System – AVS*, o *IRIS Explorer* e o *VisiQuest*.

Posteriormente, apresentam-se os principais objectivos que se pretendem atingir com este trabalho e descreve-se a arquitectura global do protótipo, com destaque para o módulo de visualização.

A aplicabilidade do protótipo desenvolvido é ilustrada para o caso de uma *domus* romana de Braga. Os resultados ilustram o conjunto de operações que um utilizador pode realizar para manipular todo um modelo ou para, após selecção, deslocar algum(s) elemento(s) para analisar com mais detalhe certos pormenores.

Por fim, são elaboradas algumas conclusões e indicadas algumas ideias para trabalho futuro.

2. COMPUTAÇÃO GRÁFICA, VISUALIZAÇÃO E ARQUEOLOGIA

A Arqueologia serve-se de múltiplas ajudas, provenientes de outras disciplinas, com o propósito quer de assegurar a validade dos seus dados, quer de suportar o processo interpretativo de um arqueosítio. Por isso mesmo, a utilização da Computação Gráfica e de técnicas de

Visualização de Dados em Arqueologia não é recente e tem vindo, gradualmente, a aumentar. Este incremento é, também, um reconhecimento por parte dos arqueólogos que a Computação Gráfica e as técnicas de Visualização são uma efectiva mais-valia na validação e divulgação dos resultados da sua investigação.

2.1 Os Primeiros Tempos

Desde sempre, a Arqueologia teve necessidade de registar tudo aquilo que escavava, da forma mais rigorosa possível. Esta necessidade potenciou o desenvolvimento do desenho arqueológico, que se pode considerar o primeiro método de visualização de dados arqueológicos. Com o aparecimento dos sistemas de desenho assistido por computador, os desenhos arqueológicos, que se encontravam apenas em papel, passaram a poder estar disponíveis em suporte digital, facilitando assim a sua análise e gestão.

A utilização de técnicas da cartografia digital e de códigos de cor permitiu a implementação de sistemas interactivos para auxiliar o arqueólogo na análise da distribuição espacial dos vários achados de uma escavação, como é o caso do trabalho apresentado em [Badler78]. Mais tarde houve algumas tentativas para encorajar a comunidade de arqueólogos a utilizar o *Graphical Kernel System (GKS)* nos seus trabalhos, como se pode verificar em [Brodlie87]. Na mesma altura começaram os primeiros ensaios para a representação 3D do terreno onde decorria a escavação arqueológica, como é o caso dos trabalhos de [Harris87], e surgiram projectos que procuravam expandir o papel da Computação Gráfica na prospecção arqueológica [Boismier87].

Já na década de 90 intensificou-se a utilização das reconstruções 3D em sistemas de ambiente virtual que pretendiam divulgar sítios arqueológicos de interesse generalizado, como é o caso de [Teixeira97].

2.2 Estado da Arte

Se, no passado, a aplicação da Computação Gráfica e da Visualização à Arqueologia se caracterizava mais por ser uma ferramenta de análise complementar e de divulgação dos dados arqueológicos, nos dias de hoje caracteriza-se por ter uma intervenção maior nas fases de escavação e interpretação, e por utilizar interfaces com o utilizador mais intuitivas. Os projectos *ARCHAVE* e *3D-MURALE* são exemplos disso.

2.2.1 ARCHAVE

O *ARCHAVE* (*Archaeology in Virtual Environments*) é, fundamentalmente, um sistema que utiliza um ambiente virtual enquanto interface para a investigação e análise arqueológica. O objectivo principal deste projecto é apoiar o trabalho do arqueólogo numa fase posterior à escavação [Acedo00].

O sistema *ARCHAVE* permite visualizar um modelo das estruturas arquitectónicas encontradas, bem como as diferentes áreas de escavação. Cada área possui ainda informação acerca das diferentes campanhas de escava-

ção. Todos os dados da escavação são armazenados numa base de dados.

A análise espacial dos dados da escavação é feita em ambiente virtual. Este ambiente virtual permite ao arqueólogo estudar o arqueosítio, navegando através das diferentes etapas de escavação ou de fases históricas importantes de um edifício (no caso, o grande Templo de Petra, na Jordânia). O acesso à base de dados pode ser feito através de um sistema de reconhecimento de voz ou de comandos baseados em gestos.

Este sistema representa uma mais-valia pelo facto de possibilitar aos arqueólogos uma melhor compreensão do contexto de escavação, e relacionar uma grande quantidade de artefactos com as características do sítio e da arquitectura [Vote01].

Porém, uma desvantagem deste sistema é o elevado custo do equipamento usado na visualização e interacção com os dados arqueológicos (CAVE e/ou Mesa Virtual).

2.2.2 3D-MURALE

O projecto *3D Measurement and Virtual Reconstruction of Ancient Lost Worlds of Europe (3D MURALE)* está a desenvolver um sistema multimédia de baixo custo, capaz de registar, reconstruir, codificar e visualizar edifícios, partes de edifícios, estátuas, partes de estátuas, cerâmica, estratigrafia, geometria e textura do terreno, e texturas de materiais. Este sistema está associado a uma base de dados que permite armazenar a informação de uma forma adequada e disponibilizá-la de acordo com as necessidades dos utilizadores [Cosmas01].

Os registos da informação arqueológica são feitos utilizando, fundamentalmente, dois suportes: máquina fotográfica digital e câmara de vídeo digital. A informação de fotogrametria é utilizada para registar edifícios, partes de edifícios e estátuas. A reconstituição 3D da informação arqueológica é realizada recorrendo a várias ferramentas, que incluem algum *software* comercial como o *AutoCAD* e o *Maya*. Tanto a textura dos materiais como a textura do terreno são produzidas utilizando um gerador de texturas.

O sistema de visualização do *3D MURALE* está direccionado para dois tipos de utilizadores: o arqueólogo e o visitante comum. Ao arqueólogo é fornecida uma ferramenta que permite visualizar os achados arqueológicos com grande rigor, em particular a estratigrafia e o espólio encontrado. O visitante comum não necessita de tanto rigor na visualização dos dados arqueológicos, mas, em contrapartida, necessita de compreender o sítio arqueológico. Assim, o *3D MURALE* permite ao visitante comum inferir a aparência de um sítio (na circunstância, o sítio arqueológico de Sagalassos, na Turquia) ao longo do tempo. Os dados são também tratados para permitir quer a visualização através da *Web*, quer num sistema de visualização *stand-alone* para museus [Cosmas03].

2.3 O Futuro

Um dos grandes objectivos da chamada Arqueologia Virtual (conceito introduzido por [Reilly89]) era a utilização de reconstruções 3D para representar modelos de edifícios antigos e de artefactos. O progresso da Computação Gráfica permitiu que os modelos criados se tornassem cada vez mais realistas e visualmente agradáveis. Infelizmente, este aumento de realismo, que é alcançado, quer através da visualização fotorealista quer através da visualização não-fotorealista, está associado a um menor grau de interacção com os modelos.

Uma nova corrente de pensamento, na Arqueologia Virtual, pretende contrariar esta tendência e aumentar o grau de interacção com os modelos para ampliar a percepção do realismo [Roussou03]. Assim, os mesmos modelos poderão ser utilizados por diferentes públicos-alvo (especialistas e estudantes, mas também o público comum). Para os especialistas e estudantes os modelos deverão suportar, em tempo-real, várias operações sobre as suas características como, por exemplo, a alteração da posição e do tamanho dos objectos, a modificação das propriedades dos materiais e a variação do modelo de iluminação. Por outro lado, o público comum poderá efectuar operações de simplificação de cada modelo para facilitar a sua compreensão.

Para além disso, um modelo não deve servir apenas para descrever o que foi encontrado num sítio arqueológico, mas também para construir todo o processo explicativo das diferentes etapas da intervenção arqueológica, desde a aquisição de dados à compreensão do sítio [Barcelo03].

Por outro lado, segundo [Barcelo05], um caminho que também se deve explorar é a utilização de Inteligência Artificial, mais concretamente de redes neuronais, para desenvolver sistemas que possam assistir o arqueólogo durante a escavação. Esse auxílio passará pela realização automática de trabalhos como a escavação ou o levantamento e registo de dados. De certa forma, este projecto conceptual de desenvolver um robot para intervir em escavações arqueológicas vem ao encontro da ideia expressa em [Forte00] de que o futuro da Arqueologia Virtual depende de uma série de factores, dos quais se pode destacar a evolução tecnológica do *hardware* e *software*.

3. SISTEMAS DE VISUALIZAÇÃO 3D

A associação de um comportamento dinâmico a modelos 3D de sítios arqueológicos é, essencialmente, um problema de visualização de dados. Assim, as características de uma aplicação para a visualização de dados arqueológicos dependem das funcionalidades do sistema de visualização 3D escolhido como base.

De seguida apresentam-se as características principais de alguns sistemas comerciais de visualização 3D mais divulgados: *AVS*, *IRIS Explorer* e *VisiQuest*. Na secção seguinte será apresentado o *VTK*, que é um sistema de visualização 3D *public-domain*.

3.1 Advanced Visual System – AVS

O *AVS* é um sistema de visualização 3D extremamente versátil e completo, que pode ser facilmente usado e que fornece métodos de visualização (p.ex., optimização de malhas poligonais, representação de campos vectoriais, representação de iso-linhas ou iso-superfícies, etc.) extremamente eficazes para representar dados de elevada complexidade [AVS05].

As áreas de aplicação do *AVS* vão desde a Medicina à Geologia, sendo inclusivamente utilizado na visualização de dados financeiros. O *AVS* possui módulos especializados para importar dados em formato *raster* e formato vectorial, para importar e manipular dados de sistemas de informação geográfica e para interagir com bases de dados diversas. Permite também a exportação dos dados 3D para *VRML* e a criação de filmes em formato *MPEG* e *AVI*. Este sistema de visualização 3D é suportado por todas as plataformas (*PC*, *UNIX* e *MAC*).

3.2 IRIS Explorer

O *IRIS Explorer* é um ambiente de programação visual estruturado em módulos, que usa bibliotecas do *Open Inventor*, *ImageVision* e *OpenGL*, associadas à biblioteca numérica do *NAG*, para visualizar, animar e manipular dados 3D [IRIS05].

Tendo em conta a sua modularidade, o ambiente de programação visual do *IRIS Explorer* permite a fácil criação de aplicações adaptadas às necessidades do utilizador, seleccionando os módulos adequados e ligando-os entre si. Para alterar e actualizar uma aplicação o procedimento é o mesmo. Assim, é possível analisar interactivamente colecções de dados complexas e visualizar os resultados para análise, apresentação e publicação. Após a versão 5.2, o *IRIS Explorer* passou a ser suportado pelos sistemas operativos *PC*, *UNIX* e *MAC*. O *IRIS Explorer* é aplicado em áreas como a Medicina e a Geologia, entre outras.

3.3 VisiQuest

O sistema *VisiQuest* é fundamentalmente um *software* de análise de imagem e dados que possui um ambiente de programação visual e está estruturado em módulos. Esta característica permite desenvolver rapidamente protótipos adequados às necessidades do utilizador [VisiQuest05].

Este *software* possui várias funcionalidades que vão desde a manipulação de dados à visualização interactiva (dados 2D e 3D) e as suas áreas de aplicação abarcam desde a Medicina à classificação de imagens de satélite. O *VisiQuest* está disponível para a maioria dos sistemas operativos (incluindo *MAC OSX*).

4. VISUALIZATION TOOLKIT – VTK

O *VTK* é um sistema orientado para objectos, projectado para desenvolver aplicações que tenham fortes componentes de Computação Gráfica, Visualização e Processamento de Imagem. A implementação de interfaces com

o utilizador pode ser realizada em *C++*, *Tcl*, *Python* e *Java* [VTK05a].

O *VTK* suporta um conjunto muito vasto de estruturas para representar dados 3D e também o fácil intercâmbio de dados com outras aplicações. Além disso, o *VTK* possui um grande número de módulos para processamento de imagem e visualização de dados.

O *VTK* é um *toolkit* gratuito, aberto e disponível para várias plataformas. A principal desvantagem do *VTK* é o facto de não possuir um ambiente de programação visual.

Apesar disso, relativamente aos sistemas de visualização anteriores, o *VTK* parece ser o sistema de visualização mais útil e adequado quando se necessita de visualizar dados 2D e 3D, e também de ter interacção com os dados durante o processo de visualização.

4.1 Áreas de Aplicação

Podemos considerar que as áreas de aplicação do *VTK* são todas aquelas onde é necessário resolver problemas de visualização. Assim, áreas como a Medicina, Biologia, Acústica e Cartografia são particularmente visadas pelo *VTK* devido à especificidade dos seus dados [VTK05b].

Exemplos desta aplicação são [VTK05b]:

- O projecto *CustusX*, que usa dados provenientes da Tomografia Axial Computorizada e da Ressonância Magnética;
- O projecto da Universidade de Stanford, na Califórnia, para criar uma biblioteca de modelos de seres vivos para aplicar no ensino;
- O projecto *Center for New Music and Audio Technologies* da Universidade de Berkeley, na Califórnia, para visualização de campos acústicos.

4.2 Breve Descrição das Funcionalidades do VTK

O que torna o *VTK* tão poderoso é o facto de disponibilizar um conjunto extenso de algoritmos úteis para a Visualização de Dados e para o Processamento de Imagem [VTK05c].

O *pipeline* de visualização do *VTK* é, de facto, muito simples e começa sempre com a leitura de dados ou geração de modelos. Estes dados ou modelos são então filtrados e, de seguida, é feito o *rendering* do resultado. Por fim, o utilizador pode interagir com os dados filtrados. Os filtros tanto podem ser aplicados a imagens *raster*, como a dados 3D.

Alguns exemplos de filtros (i.e., operações) aplicáveis a imagens são [Avila04]:

- O redimensionamento de uma imagem;
- A criação de uma imagem simétrica da original;
- A aplicação de operações lógicas sobre imagens (*AND*, *OR*, *XOR*, *NAND*, *NOR*, *NOT*);
- A aplicação de uma operação de *thresholding*;

- A conversão de uma imagem para valores *RGB*.

Para operar sobre dados 3D existem, também, uma grande variedade de filtros, tais como:

- A geração de normais a uma superfície;
- A triangulação de polígonos;
- A redução do número de triângulos de uma malha poligonal;
- A eliminação de pontos coincidentes e de primitivas degeneradas;
- A implementação da triangulação 3D, usando o algoritmo de *Delaunay*;
- A geração de iso-superfícies e de iso-linhas.

O *VTK* é composto fundamentalmente por dois modelos: o modelo gráfico e o modelo de visualização. O modelo gráfico é responsável pela “*imagem final*” que o utilizador vê no seu dispositivo de saída (por exemplo monitor ou mesa virtual).

Assim, este modelo é composto pelos seguintes objectos fundamentais:

- O *Renderer* – encarregado de representar uma cena. Coordena o *rendering* das luzes, câmaras e actores;
- As Luzes – responsáveis por controlar a iluminação dos actores numa cena;
- As Câmaras – que controlam a posição do observador;
- Os Actores – objectos que são representados pelo *Renderer* numa cena;
- A Geometria – associa uma geometria (i.e., uma forma) específica a um actor;
- As Propriedades dos Materiais – que permitem associar a cada actor materiais com diferentes características;
- As Matrizes de Transformação – responsáveis pela especificação da posição e orientação dos actores, câmaras e luzes;
- O *Render Window* – responsável pela gestão de uma janela no dispositivo de saída.

O modelo de visualização é composto por dados poligonais, imagens e volumes, malhas de elementos finitos, grelhas estruturadas e grelhas rectilíneas. A estes dados, o *VTK* pode aplicar filtros que produzem novos dados. Note-se que os dados de visualização têm associada não apenas informação geométrica, mas também informação topológica.

Para além de se poder guardar todos os dados num formato próprio, existe ainda uma interface para importar e/ou exportar dados *raster* e 3D. Para complementar a tarefa de visualização, o *VTK* possui ainda um conjunto de métodos de anotação, como é o caso do texto 2D e 3D e das escalas.

Também para o processamento de imagem o *VTK* possui um conjunto de filtros muito completo, perfeitamente integrados com o processo de visualização. Alguns exemplos destes filtros são:

- A geração do histograma de uma imagem;
- O cálculo do gradiente de uma imagem;
- A extracção de contornos de imagens.

5. PROTÓTIPO DESENVOLVIDO COM BASE NO *VTK*

Tal como já foi referido na Secção 2, a aplicação da Computação Gráfica e de técnicas da Visualização à Arqueologia deve ir no sentido de conferir interactividade aos modelos de reconstrução de sítios arqueológicos.

As características funcionais do *VTK* mostram que este sistema de visualização pode ser uma mais-valia importante para atingir os propósitos referidos anteriormente. Assim, o protótipo desenvolvido é um primeiro ensaio de visualização dinâmica de modelos arqueológicos virtuais usando o *VTK* (versão 4.4).

5.1 Objectivos

Nesta primeira fase são utilizados os modelos (estáticos) 3D desenvolvidos em [Bernardes02]. Tirando partido das características do modelo gráfico do *VTK*, o protótipo desenvolvido permite a manipulação dos modelos e a alteração interactiva de algumas das suas características. Desta forma, a investigação arqueológica e a divulgação dos seus resultados ganham outra dimensão.

Assim, os objectivos principais do protótipo, que está a ser desenvolvido, podem ser enunciados da seguinte forma:

- Visualização dos modelos de reconstrução arquitectónica de estruturas encontradas em arqueossítios;
- Alteração de propriedades materiais desses modelos virtuais;
- Manipulação de planos de corte e visualização de secções que podem conferir outro tipo de leitura aos modelos 3D;
- Visualização, diferenciada, dos elementos efectivamente encontrados durante o processo de escavação, e dos elementos que resultam da interpretação arqueológica.

Para além destes objectivos, é também propósito deste trabalho desenvolver uma aplicação que tenha uma interface bastante intuitiva. Desta forma, os utilizadores (fundamentalmente arqueólogos ou estudantes de Arqueologia) terão um tempo de aprendizagem reduzido, podendo rapidamente interagir com os modelos virtuais.

5.2 Arquitectura da Aplicação

Tendo em conta os objectivos propostos para esta aplicação, a sua arquitectura foi estruturada em três componentes nucleares distintas (figura 1):

- Interface com o Utilizador;

- Repositório de dados arqueológicos;
- Visualização de dados arqueológicos.

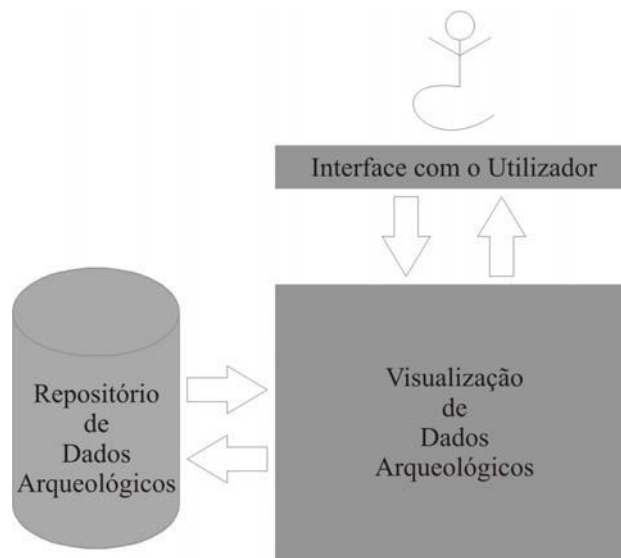


Figura 1: Arquitectura global da aplicação

5.2.1 Interface e Repositório de Dados

Não obstante a importância destas duas componentes, nesta fase do desenvolvimento da aplicação a maior atenção recai sobre a componente de *Visualização de Dados Arqueológicos*. No entanto, sempre vale a pena mencionar que a *Interface com o Utilizador* está a ser desenvolvida em C++ usando o *Visual Studio .NET*.

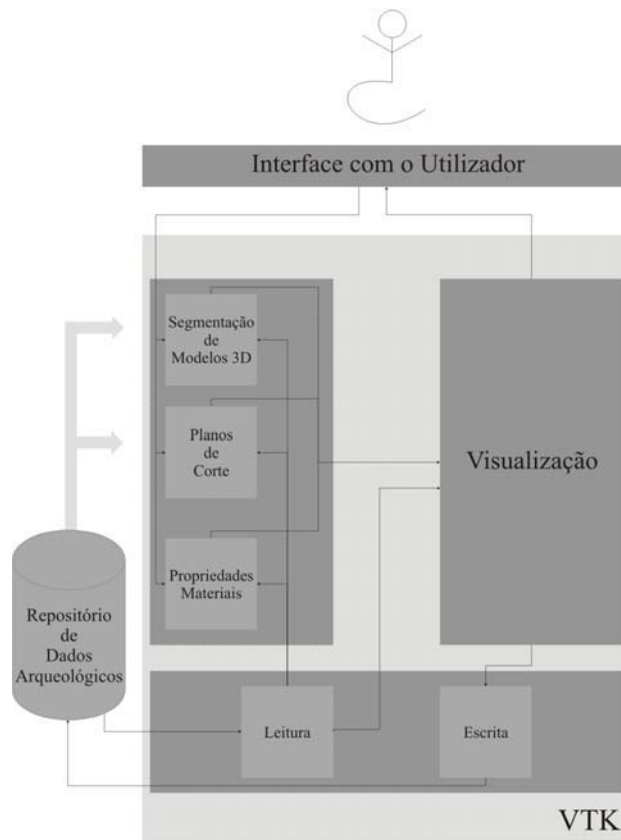


Figura 2: Arquitectura da Visualização de Dados

O *Repositório de Dados Arqueológicos* poderá conter dados muito variados, os quais podem ir desde os modelos 3D de arqueosítios até aos desenhos de planos de escavação, e à informação dos cadernos de campo. Como de momento se está a trabalhar com um número relativamente reduzido de dados, o seu armazenamento é feito numa estrutura de directorias.

5.2.2 Visualização de Dados

A figura 2 representa a arquitectura da componente de *Visualização de Dados Arqueológicos*. Esta é também implementada em C++ e constituída por três unidades fundamentais:

- Módulo de Leitura e Escrita de Dados;
- Módulo de Tratamento de Dados;
- Módulo de Visualização.

Depois do modelo 3D de um sítio arqueológico ser lido pela aplicação, o utilizador pode accionar um conjunto de filtros que têm por fim criar mecanismos que auxiliem o arqueólogo na leitura do modelo. Esta alteração do modelo inicial é essencial para criar vários modelos que representam diferentes interpretações do arqueosítio. Ao observar o mesmo modelo a partir de diferentes pontos de observação ou com diferentes propriedades materiais, o utilizador pode criar diferentes leituras do mesmo modelo, que representam diferentes modelos cognitivos.

Os filtros que, no âmbito desta aplicação, irão estar disponíveis aos utilizadores são:

- Segmentação dos modelos – permite diferenciar as partes do modelo que resultam dos dados que efectivamente são encontrados durante a escavação e as que resultam da interpretação de quem estudou o arqueosítio;
- Definição de planos de corte – faculta ao utilizador leituras parciais dos modelos. Associando a este filtro a informação dos planos de escavação e dos cortes estratigráficos, podemos também obter uma diferente leitura do arqueosítio;
- Alteração de propriedades materiais – permite alterar as várias propriedades (cor, opacidade, textura, etc.) dos materiais que caracterizam os vários elementos (objectos) que compõem os modelos virtuais.

Estes filtros podem ser aplicados individualmente ou combinados entre si. Os modelos resultantes podem então ser novamente visualizados e, posteriormente, armazenados no *Repositório de Dados Arqueológicos*.

6. EXEMPLO DE APLICAÇÃO

Um primeiro exemplo de aplicação do protótipo desenvolvido permitiu visualizar um modelo virtual de uma *domus* romana que faz parte da cidade de *Bracara Augusta* (actual cidade de Braga). Esta *domus*, geralmente conhecida por Casa das Cavalheiras, foi reconstituída com base em dados arqueológicos recolhidos durante várias campanhas de escavação realizadas entre 1983 e 1995 [Martins98].

O modelo 3D utilizado neste exemplo possui 6,022 polígonos distribuídos por 849 objectos. Cada objecto tem associadas diversas propriedades (cor, transparência, brilho, reflexão difusa e reflexão especular) e não foram aplicadas texturas às faces dos objectos.

6.1 Leitura do Modelo Virtual

O modelo virtual da Casa das Carvalheiras está em formato *VRML 2.0* que assegura uma correcta estrutura hierárquica do modelo. Essa é uma estrutura em árvore onde as folhas representam os vários elementos (i.e., objectos) do modelo e os nós correspondem a agrupamentos de objectos que possuem o mesmo tipo de material. O modelo em formato *VRML* é importado pela aplicação usando a classe do *VTK* que permite a leitura de modelos *VRML* (*vtkVRMLImporter*) e imediatamente visualizado na janela da aplicação (ver figura 3).

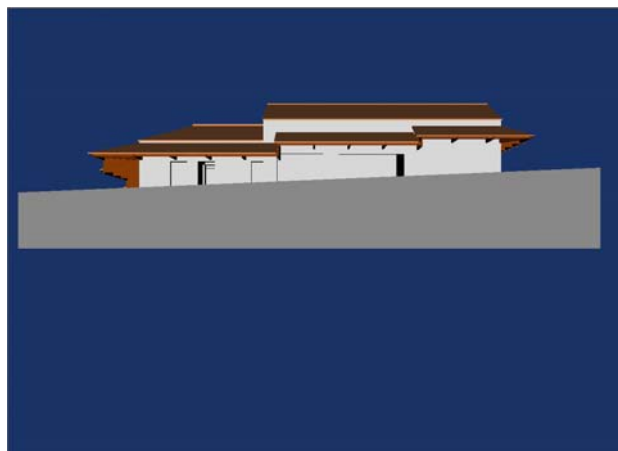


Figura 3: Modelo da Casa das Carvalheiras

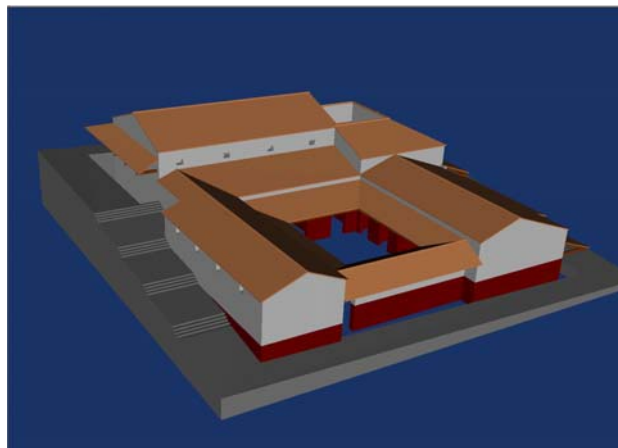


Figura 4: Alteração do ponto de observação

6.2 Interação com o Modelo Virtual

Depois de visualizado o modelo 3D, o utilizador pode interagir com o modelo ou com os diferentes objectos que o compõem. Actualmente a interação com o modelo é realizada recorrendo ao módulo de interação do *VTK* (classe *vtkRenderWindowInteractor*) no modo *camera*, onde os eventos do rato afectam a posição do observador. A figura 4 mostra o resultado da interação com o mode-

lo virtual da Casa das Carvalheiras. A interacção representada, levada a cabo apenas com o rato, corresponde a uma operação de translação e a duas operações de rotação em torno do centro do modelo 3D. Esta combinação de operações é importante para que o utilizador possa, livremente, escolher vários pontos de vista para apreender a volumetria da representação virtual.

A figura 5 representa um *zoom* do modelo, que corresponde a uma operação de translação no sentido do observador. Com esta operação o utilizador pode aproximar-se do modelo para observar em detalhe algum aspecto específico, ou pode afastá-lo para que volte à sua posição inicial.

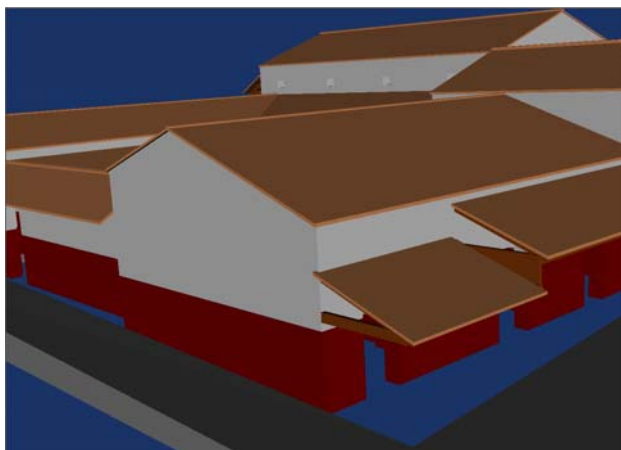


Figura 5: Zoom do modelo virtual

Para além deste tipo de interacção global, é também possível seleccionar um dos elementos constituintes do modelo virtual e operar sobre ele. Nesta fase de desenvolvimento do protótipo, a interacção sobre elementos particulares é, também, operada com o apoio do módulo de interacção do *VTk* em modo *object*. Neste modo os eventos do rato apenas afectam o objecto seleccionado. Assim, a figura 6 mostra uma operação de *picking*, em que objecto seleccionado fica envolvido numa *bounding box*.

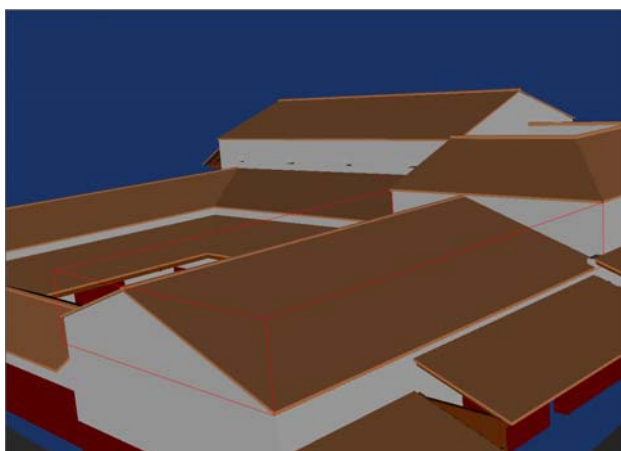


Figura 6: Selecção um elemento constituinte do modelo 3D



Figura 7: Interiores do modelo 3D

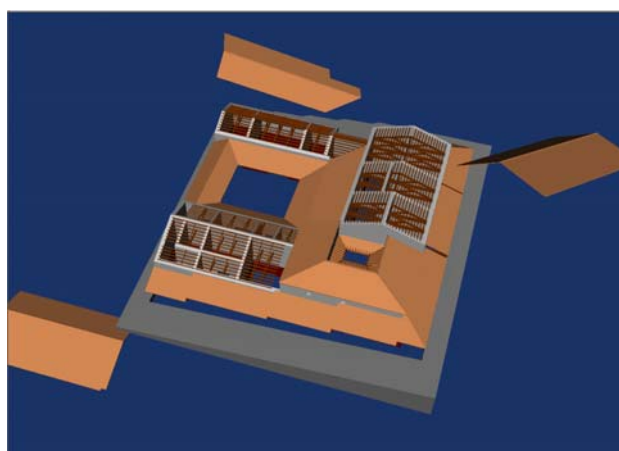


Figura 8: Interação com um objecto do modelo 3D

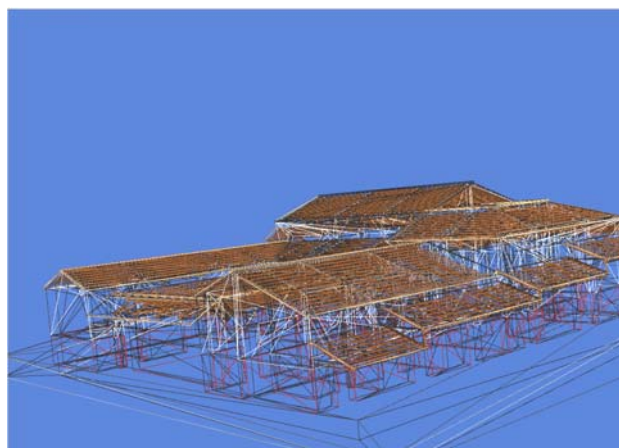


Figura 9: Representação em *wireframe* do modelo 3D

Depois de seleccionar um objecto, o utilizador pode interagir arbitrariamente com ele. Na figura 7 podem observar-se os resultados da aplicação de operações de translação a diferentes elementos do modelo. Este tipo de operação vai possibilitar ao utilizador visualizar os interiores de edifícios, bem como efectuar a desconstrução de um modelo virtual que representa um arqueosítio. Esta operação de desconstrução é uma característica que se

pode revelar importante para a compreender todo o processo que levou à formação de um registo arqueológico.

A figura 8 é uma evolução da anterior, na medida em que o observador se desloca para outro ponto de observação e efectua uma operação de rotação 3D sobre um outro objecto (um telhado da ala norte do piso inferior da Casa das Carvalheiras).

6.3 Modos de Visualização

Como se pode observar na figura 9, a aplicação permite ainda visualizar as restituições de sítios arqueológicos usando uma representação em modelo de linhas (*wire-frame*).

7. CONCLUSÕES E TRABALHO FUTURO

Foi apresentado um primeiro protótipo, com base no *VTK*, de uma aplicação para visualização e manipulação interactivas de dados arqueológicos. O exemplo de aplicação apresentado mostra que o protótipo desenvolvido e, em particular, as funcionalidades oferecidas pelo *VTK* são adequados à visualização de modelos de estruturas arquitectónicas encontradas em sítios arqueológicos. É de notar que, tanto quanto os autores puderam apurar, não existem outros exemplos de aplicação do *VTK* à Arqueologia.

O modelo virtual da Casa das Carvalheiras comporta-se como uma maquete que o utilizador pode manipular livremente, inclusivamente removendo e repondo alguns dos elementos que a constituem. Esta capacidade é uma clara vantagem para o arqueólogo, na medida em que pode desconstruir o modelo virtual e efectuar novas leituras do sítio arqueológico. Estas leituras irão influenciar a sua interpretação do arqueosítio e a construção de novos modelos cognitivos a ele associados.

Futuramente pretende-se importar não só dados arqueológicos referentes a estruturas arquitectónicas, mas também referentes ao espólio que é encontrado durante o processo de escavação. Em casos onde a estratigrafia do arqueosítio esteja bem representada, poder-se-á realizar também a desconstrução da estratigrafia de uma escavação. Este aspecto é fundamental para a correcta leitura e interpretação do sítio arqueológico.

8. AGRADECIMENTOS

Os autores gostariam de agradecer à Professora Doutora Manuela Martins (UAUM/NARQ – Universidade do Minho) e à Professora Doutora Beatriz Sousa Santos (IEETA/DET – Universidade de Aveiro) o seu precioso contributo para as ideias apresentadas neste artigo, e o seu constante apoio e incentivo.

9. BIBLIOGRAFIA

[Acevedo00] D. Acevedo, E. L. Vote, D. Laidlaw e M. S. Joukowsky, ARCHAVE: A Virtual Environment for Archaeological Research, in *Proc. IEEE Visualization 2000*, Salt Lake City, Utah, USA, 2000.

[Avila 04] L. Avila, S. Barre, R. Blue, et. al. *The VTK User's Guide*. Published by Kitware, Inc., USA, 2004.

[AVS05] Advanced Visual System, Abril 2005. <http://www.avs.com/index_wf.html>

[Badler78] N. I. Badler e V. R. Badler, Interaction with a Color Computer Graphics System for Archaeological Sites, in *Proc. SIGGRAPH'78*, 1978.

[Barcelo03] J. Barcelo, O. de Castro, D. Travet, et. al., A 3D Model of an Archaeological Excavation, in *The Digital Heritage of Archaeology*, M. Doerr, et al., Eds., Hellenic Ministry of Culture, 2003.

[Barcelo05] J. Barcelo, Virtual Archaeology and Artificial Intelligence, in *Computer Applications in Archaeology – CAA2005 Proc.*, Tomar, 2005.

[Bernardes02] P. Bernardes. *Arqueologia Urbana e Ambientes Virtuais: Um Sistema para Bracara Augusta*. Tese de Mestrado, Departamento de História, Universidade do Minho, Braga, 2002.

[Boismier87] W. A. Boismier, P. Reilly, Expanding the Role of Computer Graphics in the Analysis of Survey Data, in *Computer Applications and Quantitative Methods in Archaeology - CAA'87 Proc., British Archaeological Reports*, C. L. Ruggles e S. P. Q. Rahtz, Eds., BAR International Series 393, Leicester, 1987.

[Brodlie87] K. W. Brodli, Standardisation in Computer Graphics: an introduction to GKS, in *Computer Applications and Quantitative Methods in Archaeology - CAA'87 Proc., British Archaeological Reports*, C. L. Ruggles e S. P. Q. Rahtz, Eds., BAR International Series 393, Leicester, 1987.

[Cosmas01] J. Cosmas, T. Itegaki, D. Green, et. al., 3D MURALE: A Multimedia System for Archaeology, in *Intl. Conference on Virtual Reality, Archaeology and Cultural Heritage – VAST 2001 Proc.*, Atenas, 2001.

[Cosmas03] J. Cosmas, T. Itegaki, N. Joseph, et. al., Providing Multimedia Tools for Recording, Reconstruction, Visualisation and Database Storage/Access of Archaeological Excavations, in *4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage – VAST 2003 Proc.*, 2003.

[Forte00] M. Forte. About Virtual Archaeology: Disorders, Cognitive Interactions and Virtuality, in *CAA 2000 Proc.*, Juan A. Barceló, Maurizio Forte & Donald H. Sanders, Eds., BAR S843, págs. 247-259, Archaeopress, Oxford, 2000.

[Harris87] T. Harris, Digital terrain modelling and three-dimensional surface graphics for landscape and site analysis in archaeology and regional planning, in *Computer Applications and Quantitative Methods in Archaeology - CAA'87 Proc., British Archaeological Reports*, C. L. Ruggles e S. P. Q. Rahtz, Eds., BAR International Series 393, Leicester 1987.

- [IRIS05] IRIS Explorer, Abril 2005.
<http://www.nag.co.uk/Welcome_IEC.html>
- [Martinez92] V. Martinez. *Teoría y Método de la Arqueología*. Editorial Síntesis, 3ª Edição, Madrid, 1992.
- [Martins98] M. Martins. A Zona Arqueológica das Carvalheiras. Balanço das Escavações e Interpretação do Conjunto. *Cadernos de Arqueologia*, vol. 14-15, págs. 23-45, 1998.
- [Reilly89] P. Reilly. Data Visualization in Archaeology. *IBM Systems Journal*, vol.28, num.4, págs. 569-579. 1989.
- [Roussou03] M. Roussou, G. Drettakis. Photorealism and Non-Photorealism in Virtual Heritage Representation, in *4th International Symposium on Virtual Reality, Archaeology and Cultural Heritage - VAST 2003 Proc.*, 2003.
- [Teixeira97] J. C. Teixeira, A. Côte-Real, P. Bernardes e F. P. de Macedo, Virtual Environments and Cultural Heritage, in *Proc. Fifth International Conference on CAD and Graphics*, China 1997.
- [VisiQuest05] VisiQuest, Abril 2005.
<<http://www.accusoft.com/>>
- [Vote01] E. L. Vote. *A New Methodology for Archaeological Analysis: Using Visualization and Interaction to Explore Spatial Links in Excavation Data*. Tese de Doutoramento, Brown University, Providence, RI, 2001.
- [VTK05a] The Visualization Toolkit – What is VTK?, Abril 2005.
<<http://public.kitware.com/VTK/index.php>>
- [VTK05b] The Visualization Toolkit – VTK in Use, Abril 2005.
<<http://www.kitware.com/case/vtkinuse.html>>
- [VTK05c] The Visualization Toolkit – VTK Technical Documentation, Abril 2005.
<<http://public.kitware.com/VTK/documents.php>>

Visualização de edifícios a partir de dados LIDAR

Nuno Rodrigues
ESTG
Instituto Politécnico de Leiria
nunorod@estg.ipleiria.pt

Adriano Lopes, Próspero Santos
DI-CITI/FCT
Universidade Nova de Lisboa
{alopes, ps}@di.fct.unl.pt

Gil Gonçalves
DM/FCT
Universidade de Coimbra
gil@mat.uc.pt

Resumo

Os dados do tipo LIDAR (Light Detection and Ranging) têm suscitado, nos últimos anos, um interesse crescente enquanto fonte de dados para criação de modelos digitais de superfície (MDS), nomeadamente na localização, caracterização e monitorização de malhas urbanas.

Neste artigo descreve-se uma metodologia para identificação e visualização de edifícios com base exclusivamente em dados LIDAR. Após o processo de modelação de dados, o algoritmo começa por identificar, em primeiro lugar, as paredes dos edifícios e só depois as restantes componentes. O processo de classificação recorre também a células com informação adicional que entretanto são criadas a partir dos dados iniciais. No final do artigo são apresentados alguns testes realizados.

Palavras-chave

Identificação e visualização de edifícios, LIDAR, modelo digital de superfície.

1. INTRODUÇÃO

No passado recente tem-se assistido a um esforço de investigação em mecanismos automáticos de identificação e visualização de edifícios a partir de diferentes fontes de dados [Gruen97, Mayer99]. As fontes mais utilizadas têm sido as imagens aéreas e os dados LIDAR. Recentemente, muitos dos estudos apontam para soluções híbridas utilizando ambos os tipos de dados, tentando assim tirar partido das vantagens inerentes a cada uma das metodologias [Liang-chien04].

Neste trabalho pretendemos mostrar que é possível obter resultados satisfatórios recorrendo apenas a dados LIDAR. Estes dados são obtidos através de tecnologia por varrimento laser, que detecta a altura dos objectos acima da superfície do terreno. É comum instalar este tipo de plataforma num avião, onde um sensor envia feixes de luz laser em direcção à superfície em estudo e mede o tempo que estes demoram a ser reflectidos. Assim, é possível calcular a distância entre o sensor e o terreno (ver Figura 1), e gerar uma nuvem de pontos 3D de coordenadas (x,y,z). A partir desta nuvem de pontos é habitualmente interpolada uma grelha altimétrica que representará o modelo digital de superfície descrito na amostra 3D [Gonçalves05]. A densidade típica deste tipo de nuvens de pontos 3D tem aumentado significativamente, o que não só confere uma maior credibilidade à tecnologia enquanto fonte de dados, como também coloca desafios interessantes em termos de modelação e visualização.

A maioria das metodologias de visualização de dados LIDAR existentes sugerem uma abordagem que começa por identificar os telhados dos edifícios e só depois as

restantes componentes [Elaksher02]. Neste artigo apresenta-se uma alternativa oposta, onde se começa por identificar as paredes dos edifícios e só depois as restantes componentes.

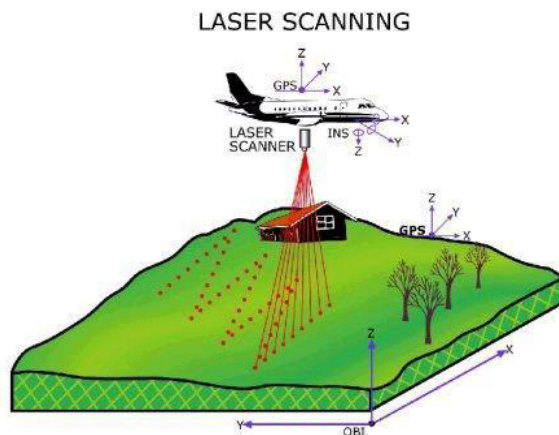


Figura 1: Detecção remota de dados LIDAR [Renslow01].

Para além desta Secção introdutória, o presente artigo está organizado da seguinte forma: na Secção 2 é apresentado o modelo de referência seguido neste trabalho e na Secção 3 é discutida a modelação de dados. Na Secção 4 descreve-se a metodologia seguida na identificação dos vários tipos de região e, na Secção 5, são abordados aspectos da visualização gráfica dos dados de interesse. Finalmente, na Secção 6, são discutidos alguns testes realizados e na Secção 7 são apresentadas conclusões e perspectivas para o trabalho futuro.

2. MODELO DE REFERÊNCIA

Este trabalho tem como suporte o modelo de referência clássico de visualização científica – *dataflow model* [Haber90].



Figura 2: Modelo de referência.

Assim, numa primeira fase efectua-se a modelação dos dados LIDAR, o que consiste na geração, por interpolação, de uma grelha de pontos a partir de uma nuvem de pontos 3D. Os dados são ainda enriquecidos através da geração de informação adicional. De seguida, os dados são classificados e posteriormente mapeados em entidades geométricas. Finalmente, procede-se à visualização dos mesmos, incluindo eventualmente uma operação de simplificação quando tal se revelar apropriado.

3. MODELAÇÃO DE DADOS

Os dados LIDAR são frequentemente disponibilizados em ficheiros no formato *ASCII xyz file*. Este formato corresponde a um ficheiro de texto que utiliza a codificação ASCII (*American Standard Code*) para armazenar os dados numa nuvem de pontos 3D de amostragem.

3.1 Interpolação de dados

O objectivo principal desta fase é gerar, por interpolação, uma grelha de pontos. Por norma essa grelha é rectangular, igualmente espaçada nos eixos *X* e *Y*, com o valor de elevação no eixo *Z*. De facto, muitos pacotes de software de visualização de modelos digitais de superfície requerem que os pontos sejam interpolados numa grelha regular.

Existem vários métodos de interpolação espacial, os quais podem ser classificados de diversas formas: locais ou globais, exactos ou aproximados, e determinísticos ou estocásticos. Muito sumariamente, nos métodos globais cada valor *Z* interpolado (nó da grelha) é influenciado por todos os valores *Z* dos pontos dados (pontos LIDAR), enquanto que, nos métodos locais, cada valor interpolado é apenas influenciado pelos valores *Z* dos pontos contidos numa dada vizinhança. Os métodos de interpolação exactos produzem uma superfície que passa pelos pontos dados. No caso dos métodos aproximados, a superfície gerada não passa necessariamente pelos pontos dados, sendo implícita a existência de alguma incerteza nos valores *Z*. Os métodos estocásticos incorporam o conceito de geo-estatística para produzir superfícies com determinados níveis de incerteza. Finalmente, os métodos determinísticos não utilizam quaisquer elementos da teoria das probabilidades.

Neste trabalho, tendo em consideração a natureza irregular de uma nuvem de pontos LIDAR, foi utilizado um interpolador local, exacto e determinista: interpolação linear baseada numa rede triangular irregular (TIN –

Triangular Irregular Network). Este algoritmo gera uma estrutura TIN a partir de uma triangulação de Delaunay. Os pontos LIDAR são ligados de modo a que nenhuma aresta de um triângulo intersecte outro triângulo. De seguida, utilizando uma pesquisa sequencial, o algoritmo procura, para cada nó da grelha, o triângulo que o contém e interpola o valor *Z* deste nó, recorrendo ao gradiente do triângulo.

O espaçamento da grelha gerada é concordante com a densidade da nuvem de pontos que a originou.

3.2 Enriquecimento de dados

A utilização de apenas informação local (*x,y,z*) na identificação de edifícios tornaria esta operação algo errática. Assim, optou-se por criar células rectangulares contíguas com variáveis derivadas de modo a facilitar a tarefa posterior de classificação de dados (terreno ou edifício).

Cada célula é criada a partir de um conjunto de quatro pontos vizinhos, e podemos ter como variáveis associadas o comprimento, a altura e a normal unitária da célula. Os valores destas variáveis derivam essencialmente das características dos referidos pontos. Por exemplo, a altura de uma célula pode ser definida como a média (ou a maior/menor) das alturas dos pontos associados.

Estes novos dados vão substituir os dados existentes, com a particularidade de serem armazenados em ficheiros binários, atendendo às vantagens inerentes a este tipo de formato, tais como o espaço ocupado e a flexibilidade e rapidez de acesso.

Paralelamente ao enriquecimento de dados, optou-se também por providenciar uma análise estatística com o objectivo de auxiliar os utilizadores na visualização de dados. Esta análise descreve essencialmente a forma como as variáveis mais significativas na identificação de células variam ao longo da grelha. O utilizador pode, assim, observar a coloração das células da grelha de acordo com o valor da variável em causa.

4. MAPEAMENTO

Após a modelação de dados, é chegada a altura de definir a informação geométrica a visualizar. Em primeiro lugar, as células da grelha de dados são classificadas em termos de tipos de região: terreno ou edifício. Ainda dentro da categoria de edifício, tem de ser feita a distinção entre paredes e telhados.

4.1 Algoritmo Proposto

O algoritmo proposto para a identificação dos edifícios é do tipo *Region-Growing*, começando primeiro pela identificação das paredes e só depois das restantes componentes. A Figura 3 ilustra o algoritmo.

Após a especificação do método de identificação, parametrizações e restrições iniciais como sejam altura e área mínimas do edifício, o algoritmo começa por definir células contíguas que satisfazem os requisitos. No essencial, ao ser localizada uma célula que satisfaça esses parâmetros, o algoritmo processa sucessivamente as células vizinhas que também satisfazem tais requisitos, de forma a criar regiões (*Region-Growing*). A definição

destas regiões permite, por outro lado, obter informação adicional, agora de cariz mais global. Assim, estas regiões são submetidas de seguida a um processo de filtragem, com o objectivo de excluir regiões de células que possam pôr em causa alguma das restrições iniciais. Por exemplo, a dimensão do edifício, ou a área mínima das paredes. Em suma, as operações de selecção e filtragem são realizadas com o propósito de melhorar o desempenho do algoritmo (note-se que apenas um subconjunto de células passa à fase seguinte).

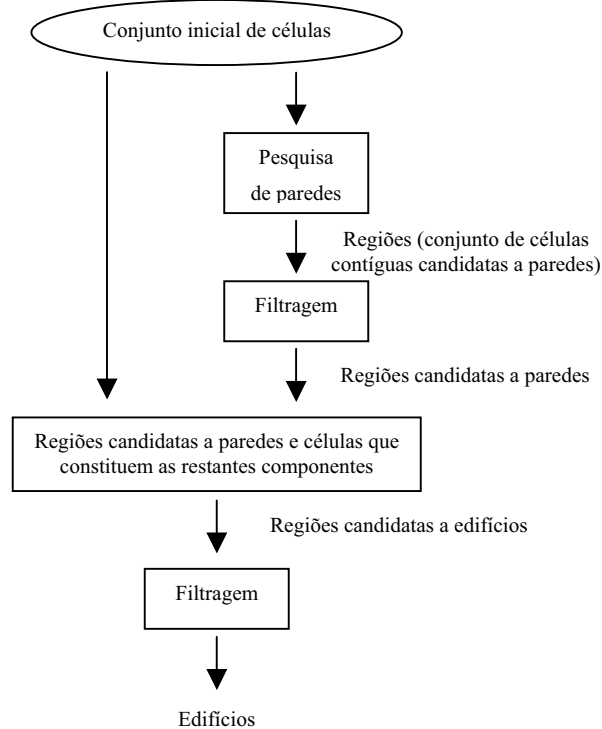


Figura 3: Algoritmo para a identificação de edifícios.

Identificadas as regiões candidatas a paredes, segue-se a identificação das restantes componentes, as quais se encontram no interior dessas regiões. Assim, o novo conjunto de células em análise é constituído pelas regiões identificadas na etapa anterior e pelas células localizadas no seu interior. Esta operação de reunião é relativamente delicada em consequência de situações particulares de localização de edifícios, as quais são abordadas na Secção seguinte. Tal como foi feito anteriormente, o resultado desta operação é novamente submetido a um processo de filtragem para remoção das regiões que não satisfazem as restrições iniciais. É de referir que esta duplicação de filtrações não prejudica o desempenho do algoritmo, uma vez que são realizadas ao nível de regiões e não de células *per si*. Por exemplo, para o MDS da Figura 9, composto por 40260 células, apenas foram detectadas 7 regiões candidatas.

4.2 Casos Problemáticos

Tal como foi referido na Secção anterior, existem algumas situações reais que podem colocar alguns obstáculos a uma correcta classificação. Essas situações derivam sobretudo de:

- resolução insuficiente de dados;
- sobreposição de entidades distintas;
- dados iniciais incompletos.

A Figura 4 ilustra uma situação de resolução insuficiente de dados, em que dois ou mais edifícios estão fisicamente mais próximos entre si do que o intervalo de amostragem da grelha. Deste modo, os edifícios são reconhecidos como um único. Na prática, é de esperar que esta situação seja pouco usual se atendermos às regras de construção de edifícios.



Figura 4: Conjunto de edifícios classificados como um único edifício.

A Figura 5 mostra um edifício assente num terreno com declive. No limite, a inclinação pode ser tal que se venha a anular a altura de uma ou mais paredes, não permitindo reconhecimento. A resolução deste problema passa pela definição de um intervalo de valores para a base das edificações dos edifícios. De facto, a base de muitos edifícios tem a configuração de um plano inclinado.

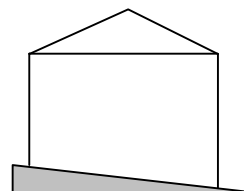


Figura 5: Sobreposição de entidades que são na realidade distintas: edifício–terreno com declive.

Por fim, a existência de dados iniciais incompletos. Esta situação pode não permitir a obtenção de uma região fechada, tal como se pode ver na Figura 6. No caso em apreço, a não identificação de uma ou mais paredes implica que não seja possível reconhecer as restantes componentes do edifício.

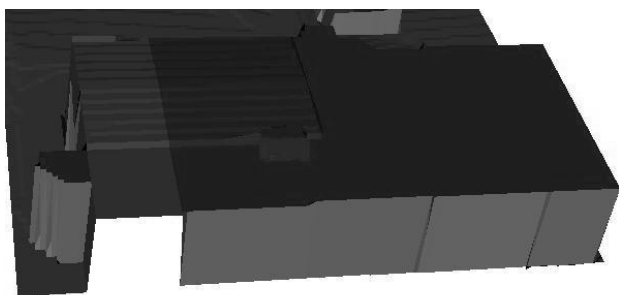


Figura 6: Dados iniciais incompletos: devido à ausência de parte da parede do edifício (cor mais clara), uma parte do telhado (cor mais escura) foi erradamente reconhecida como terreno.

5. REPRESENTAÇÃO GRÁFICA

Seguindo o modelo de referência tem-se finalmente a representação gráfica. No essencial, pretende-se mostrar valores escalares Z numa grelha rectangular X,Y , ou seja, representar superfícies. A técnica é simples e não requer cuidados especiais. De qualquer forma, as superfícies podem ainda ser coloridas de acordo com variáveis derivadas dos dados a representar, cuja utilidade é particularmente notória em fases de estudo do modelo digital de superfície.

A Figura 7 e a Figura 8 mostram algumas colorações típicas, em função da área (Figura 7) e da normal (Figura 8). De todas as variáveis estudadas, estas foram seguramente as mais interessantes.



Figura 7: Modelo de superfície colorido de acordo com a área de célula.

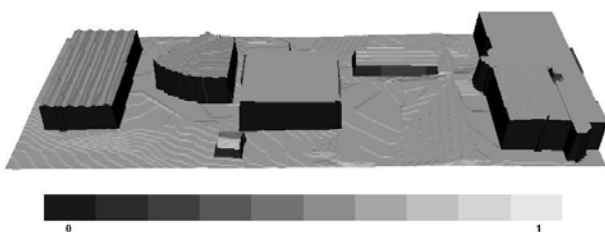


Figura 8: Modelo de superfície colorido de acordo com a direcção da normal de célula.

Tal como é patente na Figura 7, as células de maior área representam paredes de edifícios, o que se explica pelo facto de, ao estarem os pontos de amostragem igualmente espaçados em X,Y e ao surgirem paredes, as diferenças de elevação em Z entre os diversos pontos que constituem as células serem superiores às das células quase na horizontal.

Relativamente às normais, é de salientar que estas são sempre imprescindíveis para a representação gráfica do modelo 3D, independentemente da sua utilização no contexto de estudo do modelo.

Convém ainda referir que, em grandes extensões de dados, é em geral notória a existência de um gradiente praticamente nulo de valores Z . É o caso de regiões do tipo terreno. Deste modo, optou-se por facultar uma operação de dizimação (*decimation*) tendo em vista uma simplificação da malha de pontos a representar. A remoção de pontos é parametrizada pelo utilizador. Em suma, consegue-se uma melhoria de desempenho sem pôr em causa a qualidade de representação.

5.1 VTK

O VTK (*The Visualisation ToolKit*) é uma biblioteca de rotinas gráficas 3D, de código aberto, que desde há alguns anos vem sendo utilizada por muitos investigadores em todo o mundo. Esta ferramenta é composta por cerca de 700 classes em C++ e as aplicações que a utilizam podem ser implementadas num conjunto vasto de linguagens – C++, Tcl, Java ou Python. A escolha do VTK está relacionada com o facto de estar a um nível de abstracção elevado, ser código aberto e estar particularmente vocacionada para visualização científica. A título de exemplo, o VTK disponibiliza o algoritmo de dizimação referido na Secção anterior, sendo eficiente em termos de desempenho e taxa de redução, com preservação da topologia do modelo [Schroeder04].

6. TESTES

No presente trabalho foram utilizados dois MDS. O primeiro, cuja visualização se apresenta na Figura 9, consiste num MDS obtido por restituição fotogramétrica sobre um par estereoscópico de imagens áreas, onde o operador seleccionou manualmente os detalhes a representar. O segundo, representado na Figura 10, é um MDS interpolado a partir de uma nuvem de pontos LIDAR. Como este MDS não foi submetido a nenhuma operação de pré-processamento, a superfície representada pelo modelo contém, além do terreno natural, todos os objectos que se encontravam sobre a superfície no momento do varrimento laser: edifícios, pontes, vegetação, automóveis, etc.

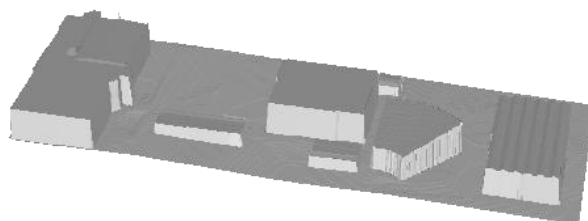


Figura 9: Resultado para o MDS obtido por restituição fotogramétrica.



Figura 10: Resultado para o MDS obtido de dados LIDAR (sem restituição fotogramétrica).

No primeiro caso (Figura 9), quase todas as células dos edifícios foram correctamente identificadas, devido sobretudo à simplicidade do modelo e à ausência de elementos que pudessem perturbar a classificação.

No segundo caso (Figura 10), como era de esperar, a existência de elementos perturbadores, como seja o caso da vegetação, conduziu a uma menor qualidade na classificação dos dados. De facto, a vegetação constitui uma das principais dificuldades na identificação de edifícios [Brunn98]. Por exemplo, mesmo que fosse possível evitar a classificação isolada de árvores como edifícios, recorrendo a restrições iniciais como a área mínima de cada edifício, existiria sempre o problema de um conjunto de árvores estarem relativamente próximas entre si, o que levaria a classificar todo o conjunto como um único edifício; ou, ainda, quando existissem árvores relativamente próximas de um edifício e que seriam classificadas como parte integrante deste. Estas situações são ilustradas na Figura 11.

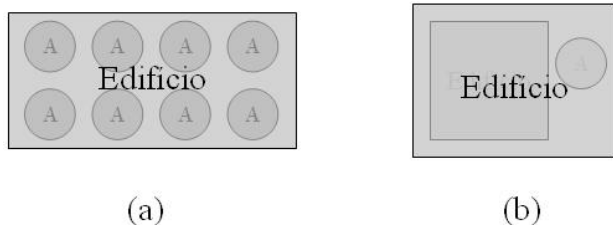


Figura 11: (a) Conjunto de árvores classificadas como edifício (b) Edifício e árvore classificadas como um único edifício.

7. CONCLUSÕES E TRABALHO FUTURO

O desenvolvimento do trabalho permite concluir, desde já, que é possível obter bons resultados na visualização de modelos digitais de superfícies utilizando apenas dados LIDAR. Por outro lado, a estratégia seguida de identificação inicial das paredes e só depois das restantes componentes revelou-se eficaz na minimização do número de testes necessários a uma correcta classificação das células.

No entanto, continuam a subsistir problemas de resolução algo complicada, nomeadamente os casos que decorrem de resolução insuficiente de dados ou de dados iniciais incompletos. Por exemplo, a classificação incorrecta devido a dados iniciais incompletos, patente na Figura 6, não surgiria se fosse seguida a estratégia clássica de identificação dos telhados em primeiro lugar. No entanto, essa estratégia também não se prestaria a uma correcta classificação de superfícies topográficas relativamente pouco planas. Nesses casos existem grandes diferenças de elevação de terreno e assim, a altura mínima a partir da qual as células são classificadas como telhado teria apenas um valor relativo – por exemplo, uma célula com determinada altura mínima pode estar associada a um telhado ou a uma montanha. Naturalmente que, em situações em que tal seja computacionalmente aceitável, poder-se-á sempre equacionar a utilização de estratégias híbridas.

A investigação futura estará direccionada para a melhoria da análise estatística de dados, derivando e conjugando novas variáveis recorrendo, inclusive, a métodos de análise inteligente sobre dados correctamente identificados. Neste campo incluem-se alguns algoritmos de inteligência artificial, nomeadamente de redes neuronais, utilizando, por exemplo, os pacotes de *software* SAS ou Clementine.

É importante referir, a terminar, que o presente trabalho se enquadra no da construção de um ambiente interactivo de visualização de dados georreferenciados, com características evolutivas em termos espaciais e temporais.

8. REFERÊNCIAS

- [Brunn98] A. Brunn, U. Weidner. Hierarchical Bayesian nets for building extraction using dense digital surface models. *ISPRS Journal of Photogrammetry & Remote Sensing*, 53(5): 296-307, 1998.
- [Elaksher02] Ahmed Elaksher, James Bethel. Reconstructing 3D buildings from LIDAR data. Technical Commission IV Symposium 2002 - Geospatial Theory, Processing and Applications, 2002.
- [Gonçalves05] G. Gonçalves. Análise do erro de interpolação em MDS urbanos obtidos por varrimento Laser. *Actas da IV Conferência Nacional de Cartografia e Geodesia*. pp 23-32, Lidel - Edições Técnicas, 2005.
- [Gruen97] A. Gruen, E.P. Baltsavias, O. Henricsson, (Eds.) *Automatic Extraction of Man-Made Objects*

from Aerial and Space Images (II), Birkhaeuser Verlag, 1997.

[Haber90] R. Haber, D. McNabb. Visualization idioms: a conceptual model for scientific visualization idioms. Visualization in Scientific Computing, pp. 74-93. IEEE Computer Society Press, 1990.

[Liang-chien04] C. Liang-chien, T. Tee-ann, S. Yi-chen, L. Yen-chung, R. Jiann-yeou. Fusion of Lidar Data and Optical Imagery for Building Modeling. XXth ISPRS Congress, Istanbul, Turkey, p. 732 ff, 2004.

[Mayer99] H. Mayer. Automatic object extraction from aerial imagery: a survey focusing on buildings. Com-

puter Vision and Image Understanding, 74(2): 138-149, 1999.

[Renslow01] M. Renslow. Development of a Bare Ground DEM and Canopy Layer in NW Forestlands Using High Performance LIDAR. ESRI international user conference, 2001.

[Schroeder04] William Schroeder, Ken Martin, Bill Lorensen. The Visualization Toolkit, 3rd edition. Kitware, Inc. Publishers, 2004.

Qualidade Medida vs. Qualidade Percebida em Malhas Triangulares dos Pulmões

Samuel Silva
IEETA – Univ. de Aveiro
Aveiro
sss@ieeta.pt

Joaquim Madeira
DET / IEETA – Univ. de Aveiro
Aveiro
jmadeira@det.ua.pt

Carlos Ferreira
CIO / DEGEI – Univ. de Aveiro
Aveiro
carlosf@egi.ua.pt

Beatriz Sousa Santos
DET / IEETA – Univ. de Aveiro
Aveiro
bss@det.ua.pt

Sumário

Em diversas áreas de aplicação, a simplificação das malhas poligonais usadas para representar os modelos, ao diminuir a sua complexidade, constitui uma possível solução para permitir a sua utilização interactiva. Em geral, uma malha obtida após um processo de simplificação possui diferenças em relação à malha original, que podem ser avaliadas usando diferentes índices de qualidade. No entanto, não é ainda claro o modo como essas medidas quantitativas (p.ex., o desvio geométrico) se relacionam com a percepção que os utilizadores têm da qualidade dos modelos simplificados.

Neste artigo, com base num conjunto de modelos pulmonares definidos por malhas triangulares e para três métodos de simplificação particulares, apresentam-se os resultados da avaliação da qualidade dos modelos simplificados usando dois processos distintos: o cálculo de índices de qualidade e a realização de uma experiência controlada com observadores. Os dados obtidos pelos dois processos são comparados, sendo possível estabelecer uma relação entre ambos. Verifica-se que, nestas condições específicas, alguns dos índices quantitativos usados podem servir como estimadores da qualidade das malhas percebida pelos utilizadores.

Palavras-chave

Malhas triangulares, Qualidade, Medidas Quantitativas, Experiência Controlada

1. INTRODUÇÃO

As malhas poligonais são utilizadas em diversas áreas, em particular em Visualização Médica, para representar modelos de órgãos e outras estruturas internas (ver, p.ex., [Bartz03]). Por vezes, devido à sua complexidade, estas não permitem a manipulação interactiva dos modelos, nem o seu armazenamento e visualização em dispositivos de fracos recursos. Isto leva a que se procure diminuir essa mesma complexidade (i.e., o número de vértices e faces de uma malha) usando, por exemplo, métodos de simplificação [Luebke01]. Este processo de simplificação resulta em diferenças (erros ou desvios) entre a malha original e a processada. Em muitas situações, é importante avaliar essas diferenças, no sentido de se perceber se são aceitáveis para a aplicação em causa.

Existem diversas ferramentas que permitem avaliar as diferenças entre malhas poligonais [Cignoni98, Roy04] e que têm sido usadas na literatura para testes de qualidade (p.ex., ver [FSilva04] e [Valette04]). As diferenças entre

malhas são habitualmente medidas através de índices de qualidade que comparam propriedades intrínsecas de duas malhas, como a posição dos seus vértices ou a curvatura das superfícies por elas definidas.

No entanto, este tipo de medidas fornece, essencialmente, uma quantificação da diferença, não se sabendo, ainda, que índices quantitativos melhor poderão expressar, e em que grau, a forma como os utilizadores percebem a qualidade dos modelos simplificados. Assim, aquando do desenvolvimento de novos métodos de processamento de malhas poligonais, ou para a comparação do desempenho de vários algoritmos para um mesmo tipo de operação, esta percepção da qualidade de uma malha, por parte dos utilizadores, terá de ser avaliada através de uma experiência controlada com observadores [Dix04]. É de notar que existe ainda pouca literatura sobre este assunto, sendo de referir o artigo de Watson *et al.* [Watson01].

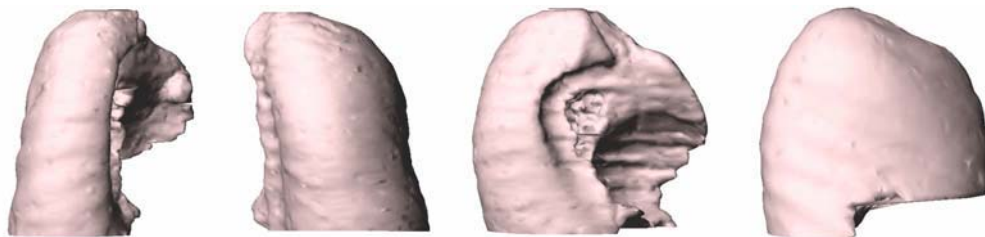


Figura 1 – Modelos pulmonares a partir dos quais foram gerados os modelos simplificados.

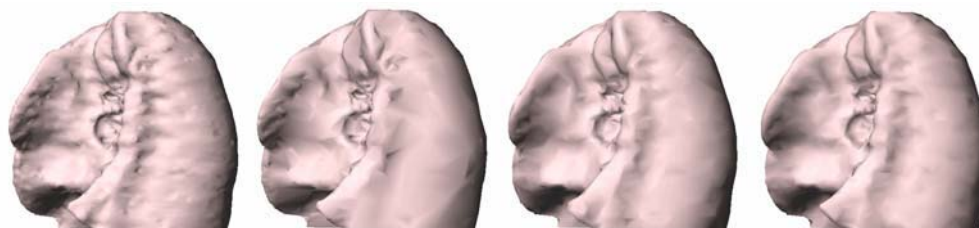


Figura 2 – Da esquerda para a direita: modelo pulmonar e as suas três versões simplificadas para 20% do número inicial de faces, usando os algoritmos de simplificação.

O presente artigo apresenta, para um conjunto de modelos pulmonares definidos por malhas triangulares e para três métodos de simplificação particulares, os resultados da avaliação da qualidade dos modelos simplificados usando dois processos distintos: o cálculo de índices de qualidade e a realização de uma experiência controlada com observadores. Os dados obtidos pelos dois processos são analisados separadamente e, depois, comparados, para que possa ser estabelecida uma relação entre os índices de qualidade e os resultados da apreciação dos observadores.

Nas secções seguintes são apresentados os modelos triangulares e os métodos de simplificação utilizados, e analisados os resultados da avaliação obtidos usando índices de qualidade. De seguida é descrita a experiência controlada e apresentados os seus resultados. Finalmente, é feita uma análise comparativa entre os resultados de ambos os processos de avaliação da qualidade das malhas, e são apresentadas as conclusões.

2. CRIAÇÃO DOS MODELOS SIMPLIFICADOS

Partindo de quatro modelos pulmonares (figura 1), construídos tal como descrito em [Silva04] e representados usando a biblioteca OpenMesh [Botsch02], foram usados três métodos de simplificação de malhas:

- QSLim [Garland97],
- algoritmo de simplificação disponibilizado na biblioteca OpenMesh [Botsch02] (OMeq),
- e esse mesmo algoritmo com um critério de simplificação adicional, relacionado com a variação permitida para as normais (*“normal flipping”*) à superfície dos modelos (OMeqNF);

e estabelecidos dois níveis de simplificação (para 20% e 50% do número original de faces) de modo a permitir avaliar o seu efeito na qualidade dos modelos. Assim, avalia-se a qualidade quer de modelos moderadamente simplificados (50%), quer bastante simplificados (20%).

Foram gerados, para cada modelo original, 6 modelos simplificados. A figura 2 apresenta um dos modelos originais e as suas três simplificações para 20% do número inicial de faces, usando os algoritmos de simplificação.

3. ÍNDICES DE QUALIDADE

Para medir as diferenças entre os modelos simplificados e os respectivos modelos originais foi usado o MeshDev [Roy04]. Esta ferramenta permite avaliar as diferenças geométricas locais entre duas malhas, ou seja, a distância entre pontos correspondentes após a simplificação, bem como as variações locais das normais à superfície das malhas. Os resultados disponibilizados incluem os valores máximo, médio e mínimo para os desvios calculados, e a respectiva variância.

3.1 Resultados

Para cada um dos modelos simplificados foram calculados os desvios geométrico e das normais. Para cada um destes desvios foram registados os valores máximo, médio e variância. Com base em cada um desses índices de qualidade, foi atribuída, a cada um dos modelos simplificados de um modelo original, e para um mesmo nível de simplificação, uma classificação (primeiro, segundo e terceiro lugares). O gráfico da figura 3 mostra a soma dos lugares atribuídos a cada método e para cada nível de simplificação. O valor apresentado

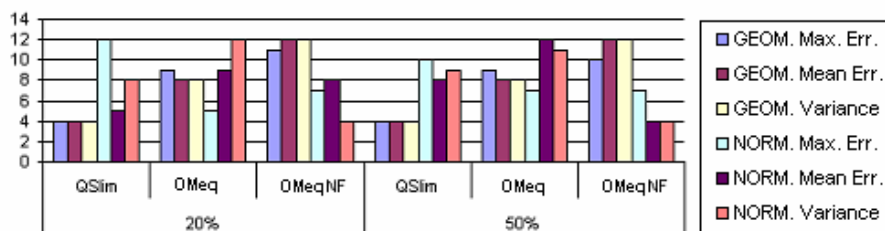


Figure 3 – Soma das posições (1º, 2º e 3º lugares) obtidas pelos três métodos, para cada nível de simplificação, usando seis índices de qualidade.

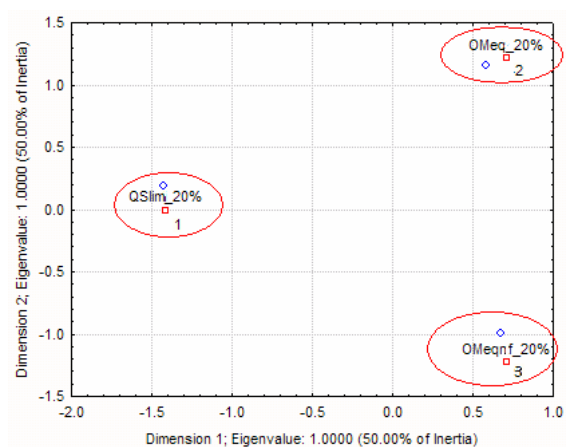


Figura 4 – Associações entre as classificações obtidas para os valores médios do desvio geométrico e os métodos de simplificação (20%).

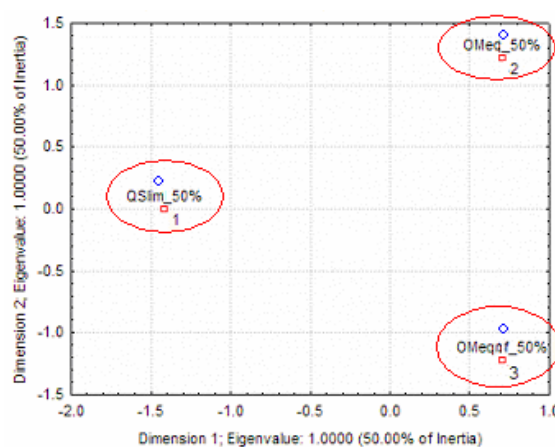


Figura 5 – Associações entre as classificações obtidas para os valores médios do desvio geométrico e os métodos de simplificação (50%).

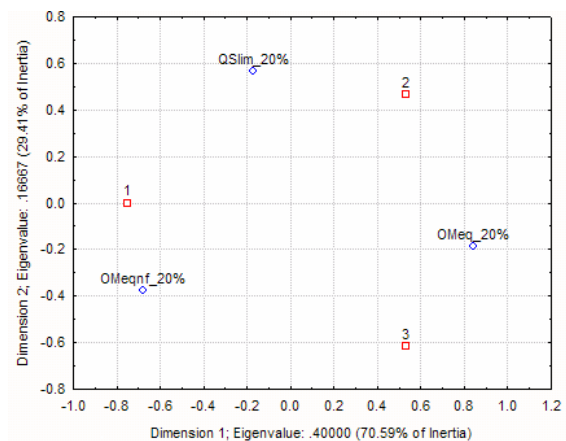


Figura 6 – Associações entre as classificações obtidas para os valores médios do desvio das normais e os métodos de simplificação (20%).

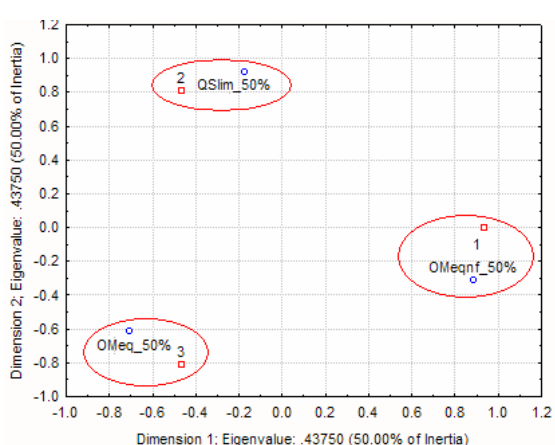


Figura 7 – Associações entre as classificações obtidas para os valores médios do desvio das normais e os métodos de simplificação (50%).

varia entre 4 (quatro primeiros lugares) e 12 (quatro terceiros lugares).

Da análise da figura 3 verifica-se que o método QSlm é o melhor classificado para ambos os níveis de simplificação

de acordo com os três índices geométricos. Por outro lado, os índices de variação das normais classificam o método OMeqNF como o melhor. Este resultado pode ser uma consequência natural do facto de, neste método, ser

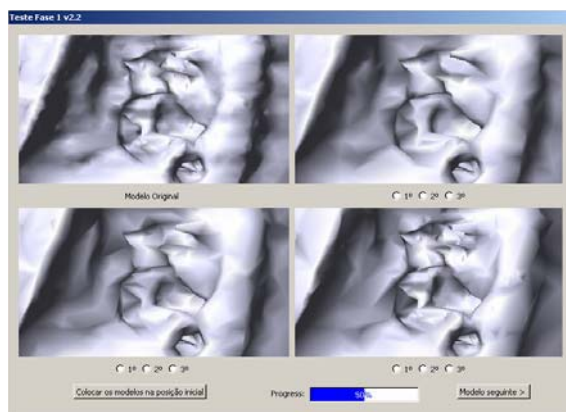


Figura 8 – Primeira fase da experiência: os observadores são confrontados com o modelo original (à esquerda, em cima) e três suas versões simplificadas. É apresentada uma situação em que foi feita uma ampliação de uma parte do modelo.

usado um critério relacionado com a variação das normais à superfície dos modelos.

De modo a estudar uma possível associação entre os métodos de simplificação e as classificações obtidas usando os valores médios dos desvios geométrico e de normais, recorreu-se a tabelas de contingência [Daniel99], optando-se por ilustrar o resultado de forma gráfica através da Análise das Correspondências [Johnson98]. Nas figuras 4 e 5, correspondentes ao valor médio do desvio geométrico, é possível verificar, para ambos os níveis de simplificação, uma clara associação entre os métodos de simplificação e as classificações: em primeiro lugar o método QSlim, seguido do OMeq e, finalmente, o método OMeqNF.

Por outro lado, para o valor médio do desvio das normais, as Análises de Correspondências apresentadas nas figuras 6 e 7 mostram resultados diferentes para cada um dos níveis de simplificação. Para o nível de simplificação de 20% não é possível observar uma associação entre os métodos de simplificação e as classificações. Para o nível de simplificação de 50%, o método OMeqNF aparece associado ao primeiro lugar, seguido do QSlim, no segundo, e, finalmente, o método OMeq no terceiro lugar.

4. EXPERIÊNCIA CONTROLADA

O objectivo da experiência foi o de comparar modelos simplificados, usando os três métodos de simplificação, de acordo com a sua qualidade percebida pelos observadores. Para efectuar as comparações entre modelos foram utilizadas preferências — em que o observador escolhe o estímulo (i.e., o modelo) no qual identifica maior qualidade —, e classificações — em que o observador atribui uma nota ao estímulo (i.e., a um modelo) de acordo com uma escala definida *a priori*.

De seguida são apresentados os aspectos principais da realização da experiência controlada, bem como os

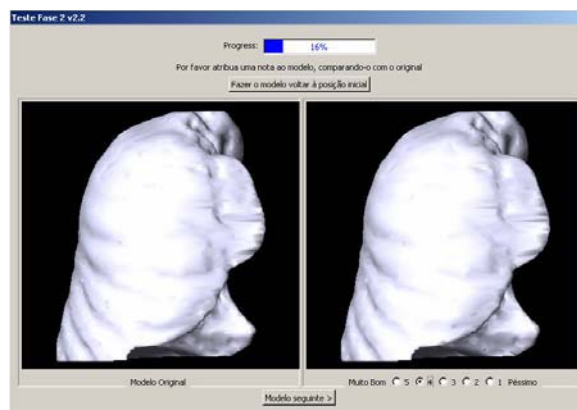


Figura 9 – Segunda fase da experiência: os observadores são confrontados com o modelo original e uma sua versão simplificada, retirada do conjunto de todos os modelos simplificados.

principais resultados obtidos. Descrições mais detalhadas destes aspectos podem ser encontradas em [Silva05].

4.1 Observadores e Métodos

O grupo de observadores que participou na experiência era constituído por 32 estudantes da Universidade de Aveiro: 18 estudantes de Engenharia e 14 de Radiologia (20 homens e 12 mulheres). As suas idades situavam-se, na maioria, entre os 18 e 25 anos. Como resultado adicional, pretendeu-se também verificar se a experiência dos estudantes de Radiologia em operações de visualização influenciava os resultados obtidos.

Para esta experiência foi escolhida uma metodologia “dentro de grupos” na qual todos os estímulos (i.e., todos os modelos de teste, no caso) são apresentados a cada um dos observadores. De modo a minimizar alguns efeitos (aprendizagem ao longo da experiência, nervosismo na primeira situação experimental ou fadiga na última) que poderiam influenciar os resultados, os diferentes (conjuntos de) modelos foram apresentados de modo aleatório a cada um dos observadores.

Todos os modelos foram representados usando sombreamento de Gouraud e foi dada a possibilidade aos observadores de alterarem a sua orientação, posição e tamanho. De modo a não causar desorientação aos observadores, as transformações resultantes da manipulação de qualquer modelo presente no ecrã eram aplicadas a todos os outros, mantendo-os sincronizados. Esta sincronização só era efectuada após a operação de manipulação terminar, ou seja, apenas o modelo manipulado era visto a movimentar-se em tempo real. Esta característica foi imposta de forma a reduzir o peso computacional destas operações. Para além disso, era disponibilizado um botão que permitia colocar todos os modelos na situação inicial.

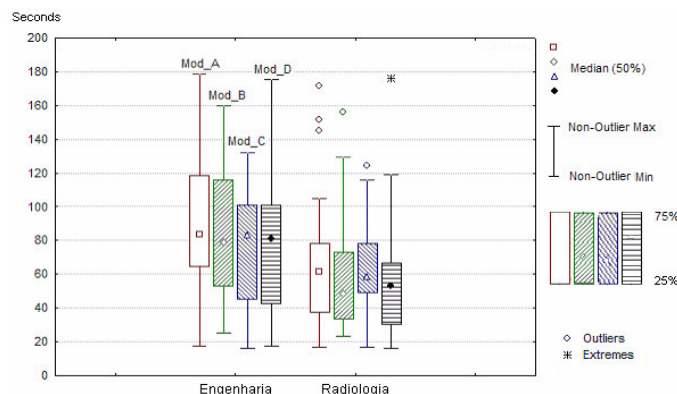


Figura 10 – Tempos de decisão agrupados por área de estudos.

4.2 Protocolo

Todos os observadores receberam, inicialmente, uma explicação sobre o enquadramento da experiência e sobre as tarefas que teriam de efectuar.

A experiência foi dividida em duas fases como se explica em seguida.

4.2.1 Fase 1 – Preferências

Nesta fase, foram apresentados a cada observador conjuntos contendo um dos modelos originais e três suas versões simplificadas, usando os três métodos de simplificação, para apenas um dos níveis de simplificação (figura 8). Era pedido aos observadores que atribuísem um primeiro, segundo e terceiro lugares aos modelos simplificados apresentados, de acordo com a sua qualidade e comparativamente ao original.

4.2.2 Fase 2 – Classificações

Nesta fase, foram apresentados a cada observador conjuntos contendo um dos modelos originais e uma sua versão simplificada (figura 9). Era pedido que fosse atribuída uma nota ao modelo simplificado usando uma escala de 1 (muito mau) a 5 (muito bom), de acordo com a sua qualidade e comparativamente ao modelo original.

4.2.3 Aplicação de suporte

De forma a permitir a implementação do protocolo definido, foi desenvolvida uma aplicação para plataformas Windows usando a linguagem C++, e as bibliotecas OpenMesh, para armazenamento das malhas poligonais, e Fox Toolkit [Fox05], para implementação da interface. Os modelos utilizados, assim como as suas versões simplificadas, foram todos lidos de disco antes do início da experiência, de modo a reduzir os tempos de espera entre situações experimentais. Para melhor desempenho, durante o *rendering* dos modelos apresentados, foram construídas listas de desenho (*"display lists"*).

Os resultados obtidos com cada observador foram, no final da experiência, reunidos num ficheiro e enviados automaticamente para um servidor.

4.3 Dados Recolhidos

No início da experiência, foram recolhidos alguns dados de modo a caracterizar o perfil de cada um dos observadores como, por exemplo, o sexo, a idade e a área de estudos.

Além das preferências e classificações atribuídas pelos observadores nas duas fases da experiência, foram também registados dados quanto ao tempo necessário, a cada um deles, para tomar uma decisão e ao respectivo número de operações de manipulação efectuadas com os modelos, de modo a se obter alguma informação sobre o grau de dificuldade da decisão.

4.4 Resultados

Nesta secção são apresentados os principais resultados obtidos na experiência controlada. Uma análise mais detalhada pode ser encontrada em [Silva05].

4.4.1 Fase 1 – Preferências

Da análise dos dados obtidos nesta fase, foi possível verificar que os observadores demoraram menos tempo a decidir quando confrontados com modelos com um nível de simplificação de 20%. Os estudantes de Radiologia foram, em geral, mais rápidos a decidir do que os estudantes de Engenharia (figura 10).

Para além disso, os observadores do sexo feminino interagiram menos com os modelos antes de tomar a decisão.

Para o nível de simplificação de 20%, os observadores preferiram os modelos simplificados usando o método QSlim (maior número de primeiros lugares) (figura 11). Em segundo lugar ficou o método OMeq seguido do OMeqNF. A Análise de Correspondências apresentada na figura 12 permite observar essa associação.

Para o nível de simplificação de 50%, os resultados não foram tão claros. Como se pode verificar pela análise do gráfico da figura 13, e pela Análise de Correspondências da figura 14, apenas se pode associar o método OMeqNF ao primeiro lugar, não sendo muito clara qualquer outra associação.

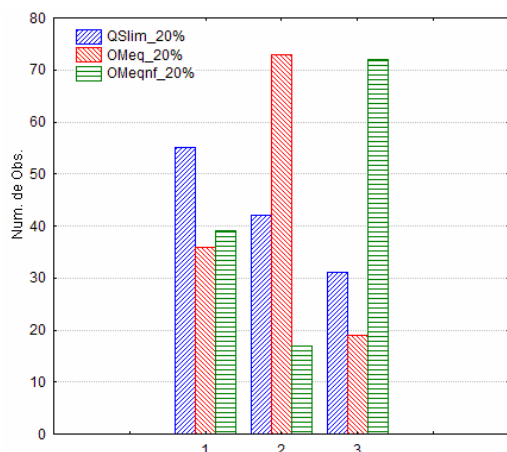


Figura 11 – Gráfico de barras mostrando o número de primeiros, segundos e terceiros lugares atribuídos a cada um dos métodos, para um nível de simplificação de 20%.

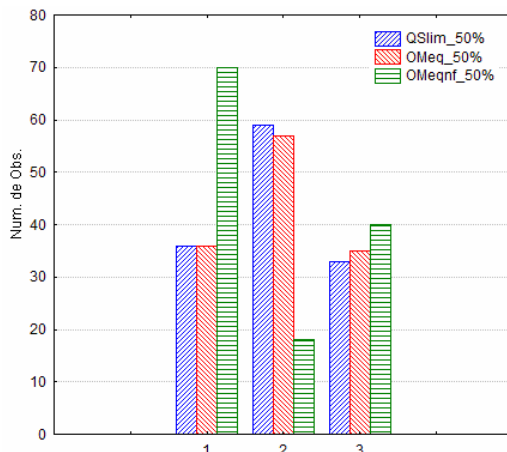


Figura 13 – Gráfico de barras mostrando o número de primeiros, segundos e terceiros lugares atribuídos a cada método para um nível de simplificação de 50%.

4.4.2 Fase 2 – Classificações

Nesta fase voltou a verificar-se que os estudantes de Radiologia decidiram mais depressa que os de Engenharia, e que os observadores do sexo feminino decidiram interagindo menos com os modelos.

Os observadores atribuíram, em geral, melhores classificações aos modelos com um nível de simplificação de 50% (figuras 15 e 16). Para o nível de simplificação de 20% o método OMeqNF é claramente o pior classificado. A análise de dados desta fase permitiu concluir que os observadores só foram sensíveis ao método utilizado, para um nível de simplificação de 20%, já que as classificações atribuídas para o nível de simplificação de 50% foram muito semelhantes entre os três métodos. A Análise de Correspondências da figura 17 ilustra esse comportamento.

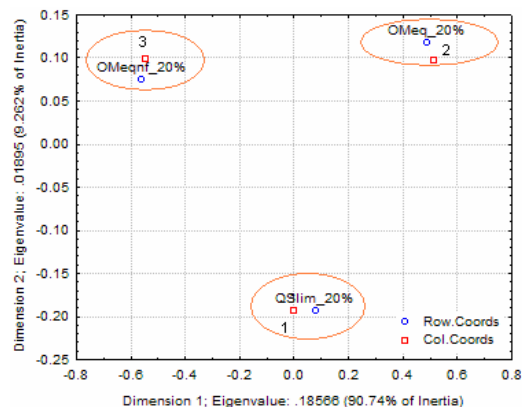


Figura 12 – Análise de correspondências mostrando a associação entre os métodos de simplificação e as preferências para o nível de simplificação de 20%.

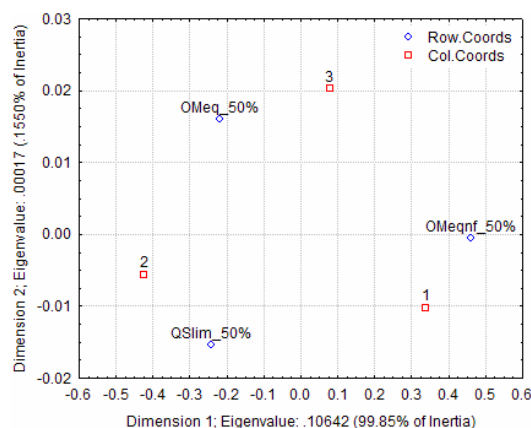


Figura 14 – Análise de correspondências mostrando a associação entre os métodos de simplificação e as preferências para o nível de simplificação de 50%.

4.5 Discussão dos resultados

Na primeira fase da experiência (preferências) foi possível fazer uma ordenação dos métodos de simplificação usados, quanto à qualidade dos modelos por eles produzidos. Essa ordenação não é a mesma para os dois níveis de simplificação. Para o nível de simplificação de 20% o método QSlm foi o preferido, seguido do OMeq e do OMeqNF. Para o nível de simplificação de 50% o preferido foi o método OMeqNF, não sendo muito clara a ordenação dos dois outros métodos.

Na segunda fase (classificações), os resultados não foram tão claros, uma vez que os observadores só se mostraram sensíveis ao método de simplificação para o nível de 20%, tendo o método OMeqNF obtido pior resultado.

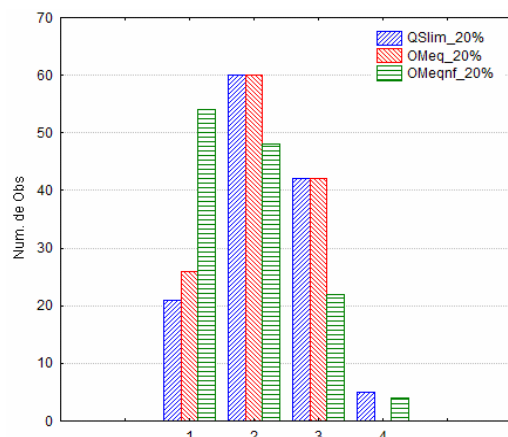


Figura 15 – Gráfico de barras mostrando as classificações atribuídas a cada um dos métodos de simplificação para o nível de simplificação de 20%.

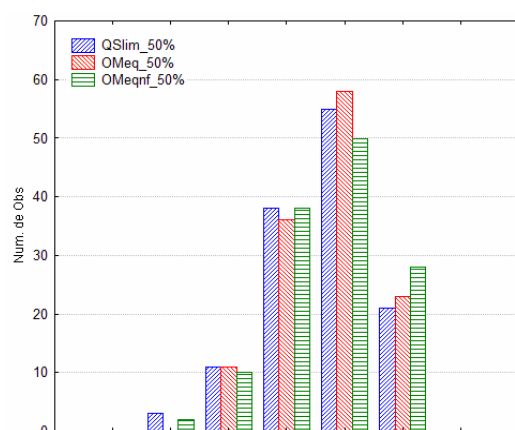


Figura 16 – Gráfico de barras mostrando as classificações atribuídas a cada um dos métodos de simplificação para o nível de simplificação de 50%.

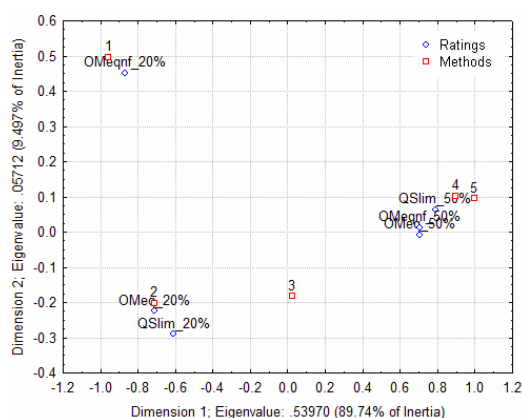


Figura 17 – Análise de correspondências mostrando a associação entre os métodos de simplificação, para ambos os níveis, e as classificações atribuídas pelos observadores.

Para o nível de simplificação de 50% todos os métodos receberam classificações semelhantes.

Resumindo, e tendo em conta as duas fases da experiência, para um nível de simplificação de 20% o melhor método foi o QSlim e para um nível de simplificação de 50% o melhor método foi o OMeqNF.

5. COMPARAÇÃO DE RESULTADOS

Os índices de qualidade usados, assim como os resultados obtidos da experiência controlada com observadores, permitem classificar a qualidade dos modelos produzidos pelos diferentes métodos utilizados. No entanto, coloca-se a questão de saber se os resultados de ambos os processos de avaliação se relacionam entre si, ou seja, se as medidas quantitativas dos índices vão ao encontro dos resultados qualitativos obtidos com observadores.

Ao efectuar uma comparação entre as Análises de Correspondências das figuras 4 e 5, e as das figuras 12 e 14, onde são mostradas projecções equivalentes para o desvio geométrico médio e para as preferências dos observadores, verifica-se que este índice de qualidade tem um comportamento semelhante ao das preferências dos observadores para um nível de simplificação de 20%.

Se a comparação for efectuada entre as Análises de Correspondências das figuras 6 e 7, e as das figuras 12 e 14, verifica-se que o desvio médio das normais tem um comportamento semelhante ao das preferências dos observadores para o nível de simplificação de 50%.

Estes resultados mostram que o valor médio do desvio geométrico expressa bem a qualidade percebida pelos observadores, quando confrontados com modelos simplificados para 20%. Para o nível de simplificação de 50%, parece ser o valor médio do desvio das normais que melhor exprime a qualidade percebida pelos observadores. Isto pode ser um indício de que estes são mais sensíveis ao desvio geométrico quando confrontados com menos informação, e ao desvio das normais quando confrontados com mais informação.

6. CONCLUSÕES

Neste artigo foi feita uma avaliação da qualidade de diversos modelos simplificados, usando três métodos e dois níveis de simplificação.

Esta avaliação foi executada usando dois processos. O primeiro envolveu a utilização de uma ferramenta que permitiu calcular as diferenças (i.e., os índices de qualidade) das malhas simplificadas, em relação às originais, em termos de características geométricas locais e da variação das normais entre as suas superfícies. O segundo processo consistiu numa experiência controlada com utilizadores em que lhes foi pedido que, usando preferências e classificações, expressassem a sua percepção quanto à qualidade dos modelos simplificados, relativamente aos originais.

Os resultados obtidos, usando estes dois processos de avaliação da qualidade, foram depois comparados, no sentido de se averiguar da existência de algumas semelhanças.

Verificou-se que, nas condições específicas descritas, alguns dos índices de qualidade utilizados (nomeadamente, o valor médio do desvio geométrico e o valor médio do desvio das normais) podem ser usados como estimadores da qualidade das malhas percebida pelos utilizadores.

No entanto, no sentido de poder generalizar os resultados obtidos, torna-se agora necessário efectuar uma experiência similar utilizando modelos de diferente natureza e usando um número mais elevado de observadores, no sentido de conferir maior significância estatística aos resultados. É também interessante aumentar o número de índices de qualidade analisados.

7. AGRADECIMENTOS

O primeiro autor agradece à Unidade de Investigação 127/94 IEETA, da Universidade de Aveiro, a bolsa que tem permitido a realização do seu trabalho.

8. REFERÊNCIAS

- [Bartz03] D. Bartz, "Virtual Endoscopy in Research and Clinical Practice", State of the Art Report – EUROGRAPHICS 2003, 2003.
- [Botsch02] M. Botsch, S. Steinberg, S. Bischoff, and L. Kobbelt, "Openmesh - a Generic and Efficient Polygon Mesh Data Structure", *1st OpenSG Symp.*, Darmstadt, Germany, Jan. 2002.
- [Cignoni98] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring Error on Simplified Surfaces", *Computer Graphics Forum*, vol. 17, pp. 167-174, 1998.
- [Daniel99] W. W. Daniel, *Biostatistics: a Foundation for Analysis in the Health Sciences*, John Wiley & Sons, 1999.
- [Dix04] A. Dix, J. Finlay, G. Abowd and R. Beale, *Human Computer Interaction*, 3rd ed., Prentice Hall, 2004.
- [Fox05] J. van der Zipp., "FOX TOOLKIT", <http://www.fox-toolkit.org>, (Abril 2005).
- [FSilva04] F. Silva, A. Gomes, "Normal-based simplification algorithm for meshes", *Proc. Theory and Practice of Computer Graphics*, pp. 211-218, 2004.
- [Garland97] M. Garland and P. Heckbert, "Surface Simplification Using Quadric Error Metrics", *Proc. SIGGRAPH '97*, pp. 209-216, Aug. 1997.
- [Johnson98] D. G. Johnson, *Applied Multivariate Methods for Data Analysis*, Duxbury, 1998.
- [Luebke01] D. Luebke, "A Developer's Survey of Polygonal Simplification Algorithms", *IEEE Computer Graphics and Applications*, vol. 21, no. 3, pp. 24-35, 2001.
- [Roy04] M. Roy, S. Foufou, and F. Truchetet, "Mesh Comparison Using Attribute Deviation Metric", *Int. Journal of Image and Graphics*, vol. 4, pp. 1-14, 2004.
- [Silva04] S. Silva, J. Madeira e B. Sousa Santos, "Construção e Visualização de Modelos Pulmonares usando Malhas Triangulares", *Actas Interacção 2004*, pp. 229-231, Lisboa, Julho 2004.
- [Silva05] S. Silva, B. S. Santos, J. Madeira, and C. Ferreira, "Comparing Three Methods for Simplifying Mesh Models of the Lungs: An Observer Test to Assess Perceived Quality", *Proc. SPIE 2005, vol 5749, Image Perception, Observer Performance, and Technology Assessment*, pp. 484-495, San Diego, Feb. 2005.
- [Valette04] S. Valette and J. Chassery, "Approximated Centroidal Voronoi Diagrams for Uniform Polygonal Mesh Coarsening", *Computer Graphics Forum*, vol. 23, no. 3, pp. 381-389, 2004.
- [Watson01] B. Watson, A. Friedman, and A. McGaffey, "Measuring and Predicting Visual Fidelity", *Proc. SIGGRAPH 2001*, pp. 213-220, 2001.

Caracterização do Movimento Tridimensional de Corpos Articulados

Pedro Mogadouro Couto
CETAV - U.T.A.D
Vila Real
pcouto@utad.pt

Luís Gonzaga Magalhães
U.T.A.D
Vila Real
lmagalha@utad.pt

Ronaldo Gabriel
CETAV - U.T.A.D
Vila Real
rgabriel@utad.pt

Pedro Melo Pinto
CETAV - U.T.A.D
Vila Real
pmelo@utad.pt

Sumário

Este trabalho recorre a técnicas de Visão por Computador e de Computação Gráfica para caracterizar o movimento 3D de corpos articulados. As técnicas de Visão por Computador são utilizadas para efectuar o seguimento de marcas e para obter a sua posição tridimensional em relação às câmaras. Com recurso a técnicas de Computação Gráfica é efectuada uma animação do movimento tridimensional do corpo articulado, a qual poderá ser um auxiliar precioso, por exemplo, no domínio da Biomecânica. No âmbito deste trabalho, foi efectuada uma aplicação ao caso do estudo da estabilidade articular do tornozelo. Os resultados obtidos mostram que a metodologia é viável.

Palavras-chave

Análise de movimento, seguimento de objectos, alta resolução temporal, reconstrução 3D, calibração, visualização 3D, biomecânica.

1. INTRODUÇÃO

Com este trabalho, que deriva da dissertação de mestrado do primeiro autor [Couto, 2003], pretende-se desenvolver e testar uma metodologia que permita caracterizar o movimento tridimensional de corpos articulados, com vista à sua aplicação no domínio da Biomecânica, mais precisamente na locomoção bípede.

A metodologia proposta neste trabalho tem por base técnicas de Visão por Computador, para efectuar o seguimento de marcas nas sequências de imagens obtidas pelas câmaras, e para estimar (reconstruir) a posição tridimensional dessas marcas. O recurso às técnicas de Computação Gráfica é realizado com o objectivo de visualizar o movimento tridimensional das marcas, fornecendo uma ferramenta para análise do movimento tridimensional do corpo articulado.

No âmbito deste trabalho, a metodologia proposta é aplicada ao caso do estudo 3D do comportamento da estabilidade articular do tornozelo. Os resultados obtidos permitiram obter dados importantes para o estudo biomecânico da estabilidade articular do tornozelo.

Estes resultados mostram a viabilidade da metodologia proposta e também que as aplicações desenvolvidas são ferramentas que podem constituir um auxiliar precioso

para o estudo, de um ponto de vista biomecânico, da locomoção bípede.

2. METODOLOGIA PROPOSTA

Esta metodologia vem no seguimento dos trabalhos de Gabriel [Gabriel, 1998], [Gabriel, 1999] e [Mourão, 1999], nos quais foram já identificados parâmetros biomecânicos para a análise de estabilidade articular. Estes parâmetros podem ser determinados com recurso a sequências de imagem de alta resolução temporal descritivas do movimento, em sincronismo com dados dinâmográficos no apoio [Mourão, 1999].

O processo de extracção da informação tridimensional, presente nas imagens, pode ser dividido em duas fases distintas: a da análise/detecção de movimento 2D (independente para cada sequência de imagens) e a da reconstrução 3D propriamente dita. Numa terceira fase é efectuada a visualização dos resultados, utilizando técnicas de computação gráfica, no sentido de recriar o movimento original.

2.1 Aquisição das imagens

As sequências adquiridas à taxa habitual, de 25 imagens por segundo, mostram-se insuficientes para caracterizar o comportamento do movimento humano, durante a locomoção bípede com velocidade e cadência de passada natural. Este problema foi identificado por diversos

autores, nomeadamente por Krebs [Krebs, 1995], no âmbito da biomecânica, e também por Singh [Singh, 1991] e Fleet [Fleet, 1992] em visão por computador. Justifica-se assim o recurso a frequências de amostragem mais elevadas, como forma de diminuir uma possível ocorrência de *aliasing*. No trabalho experimental, que realizámos, optámos por utilizar uma taxa de aquisição de 125 imagens/segundo. Esta opção foi baseada em alguns critérios que serão abordados na secção 3.

Para se poder efectuar uma análise biomecânica do Movimento Humano, a partir de sequências de imagens, é, numa primeira fase, necessário identificar em cada imagem da sequência determinados pontos considerados significativos para a caracterização do movimento.

A escolha das marcas a utilizar para a sinalização dos pontos obedece, entre outros, a dois critérios fundamentais: as marcas devem ser confortáveis para os indivíduos, de forma a não condicionarem a execução da tarefa, e devem, pelas suas características ópticas em confronto com o meio envolvente, facilitar o processo de análise de imagem via computador.

Deste modo, foram considerados vários tipos de marcas, que variavam em tamanho, forma e cor, tendo sido escolhidas esferas negras, com 0,5cm de diâmetro e 10 mg de peso, que foram “coladas” no indivíduo por intermédio de uma fita adesiva. O diâmetro e peso das marcas foram escolhidos de maneira a que, quando reproduzida na imagem, a marca represente uma área reduzida, idealmente 1 *pixel*, e de modo a não dificultarem a execução da tarefa por parte do indivíduo. A cor negra foi escolhida por apresentar grande contraste em relação a todo o meio envolvente e facilitar, deste modo, a sua identificação e posterior seguimento nas imagens. Este tipo de marcas provou ser bastante fiável, com a vantagem adicional de ser uma solução bastante económica.

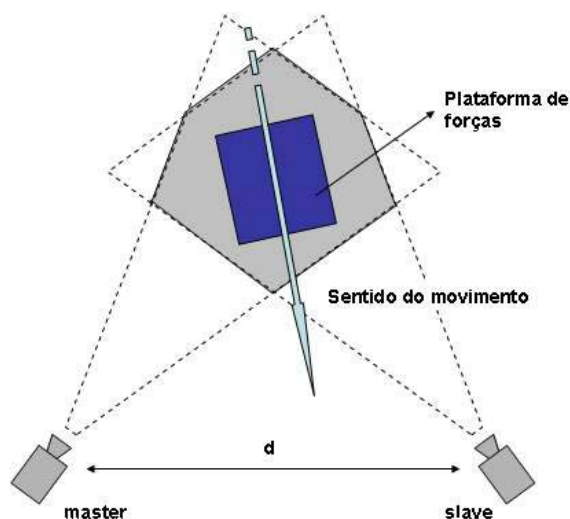


Figura 1 - Disposição das câmaras

Nos objectos articulados a estrutura do objecto pode mudar de instante para instante, o que pode inviabilizar a sua reconstrução a partir de uma única câmara. A solução encontrada, passa pela utilização de duas câmaras

sincronizadas, uma vez que com duas imagens do objecto obtidas no mesmo instante e a partir de duas posições diferentes, o problema da reconstrução 3D, para cada instante, transforma-se num problema de estimação do modelo 3D de um objecto rígido a partir de duas imagens, para o qual existem diversos métodos robustos. Na verdade, o recurso a duas câmaras sincronizadas pode tornar também possível, a utilização desta metodologia mesmo num contexto de corpos não rígidos, ou mesmo no caso limite da existência de vários corpos em movimento.

A disposição das câmaras obedece a dois critérios que a figura 1 ajuda a ilustrar: têm de estar dispostas de maneira a que as marcas sejam sempre visíveis pelas duas câmaras, devendo ser a distância entre estas (valor d na figura), no mínimo, equivalente à distância das câmaras em relação à zona de filmagem (área definida pela intersecção das áreas de visão de cada uma das câmaras, como se pode observar na figura). Este problema foi identificado através de testes laboratoriais, tendo-se verificado que quanto maior for a abertura das câmaras, respeitando o primeiro critério atrás descrito, maior é a eficácia das técnicas de reconstrução. Esta conclusão advém do facto de, ao se aumentar a abertura das câmaras, aumenta-se consequentemente o rácio, entre a amplitude do movimento de translação das marcas, e o ruído inerente às imagens, diminuindo assim, significativamente, a influência deste último na qualidade dos resultados.

2.2 Análise das imagens

Um dos problemas mais complexos e actuais com que se depara o processamento digital de imagem é, sem dúvida, o da detecção e seguimento de objectos em movimento numa sequência de imagens, ou seja, a implementação numa máquina de uma das principais características da visão humana - a apreensão do movimento.

Para a reconstrução tridimensional ser possível são necessárias, no mínimo, duas vistas da cena obtidas a partir de duas posições diferentes. Para isso, foram utilizadas duas câmaras sincronizadas com elevadas taxas de aquisição.

Uma vez que as câmaras estavam sincronizadas, as sequências por elas adquiridas foram analisadas separadamente, reduzindo a questão a um problema de detecção de movimento em sequências de imagens digitais adquiridas com câmara fixa. Esta abordagem é exequível, visto que, nesta fase, o único objectivo é determinar as trajectórias 2D das marcas quando “vistas” por cada uma das câmaras independentemente da outra. Só numa fase posterior (estimação das coordenadas 3D) haverá cruzamento entre a informação resultante da análise das sequências adquiridas por cada uma das câmaras.

Para efectuar a detecção das marcas e o seu seguimento ao longo da sequência de imagens foi desenvolvida uma aplicação para obtenção das trajectórias 2D de cada marca, com base na aplicação desenvolvida por Mourão [Mourão, 1999]. Essa aplicação foi desenvolvida com o

objectivo de comparar diferentes técnicas de *Block Matching*, e tinha também por objectivo facilitar a tarefa de marcação de pontos e o seu seguimento, ao longo de uma sequência de imagens digitais gravadas em disco. Apesar de algumas das suas funcionalidades terem sido mantidas, a dita aplicação necessitou de alterações de fundo de maneira a corresponder às exigências deste trabalho, visto ter-se revelado ineficaz no processamento das sequências de imagens digitais utilizadas no âmbito deste trabalho.

Nas técnicas de *Block Matching* (Figura 2) podem ser distinguidas duas etapas: numa primeira fase é necessário definir, numa das imagens da sequência, uma área de referência (*template window*) que englobe o objecto a “seguir”, neste caso as marcas. Numa segunda fase, para as restantes imagens da sequência, é preciso encontrar a melhor correspondência (*match*) numa área maior (área de pesquisa, *search window*) onde se supõe ocorrer o padrão inicial.

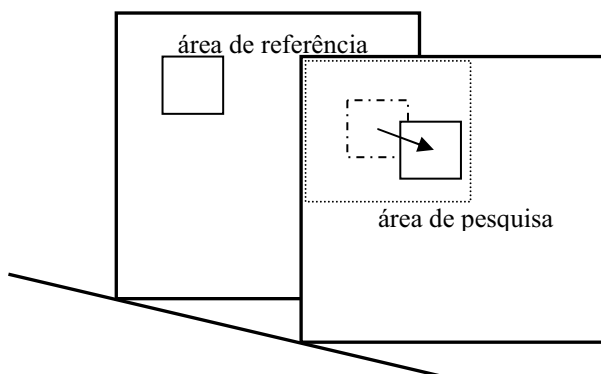


Figura 2 - Funcionamento das técnicas baseadas em *Block Matching*

Na primeira fase, de extrema importância neste tipo de estudos [Melo-Pinto, 1995], é necessário efectuar a marcação das marcas na primeira imagem da sequência. O tamanho da área de referência deve ser suficiente para englobar as marcas, assim, e como o tamanho das marcas pode ser diferente dependendo da câmara com que são “vistas” (a trajectória do indivíduo não é exactamente perpendicular em relação à linha imaginária que une as duas câmaras, o que devido à ampliação usada pelas câmaras origina diferenças na ordem do 4/5 *pixels* no tamanho das marcas), foram definidos tamanhos distintos consoante as sequências tenham sido adquiridas pelas câmaras *master* ou *slave* (Figura 1).

Na segunda fase do processo, é necessário definir a área de pesquisa onde se supõe encontrar o objecto. As dimensões da área de pesquisa são directamente proporcionais ao peso computacional do processo e inversamente proporcionais ao desempenho dos algoritmos de *Block Matching*. A escolha da área de pesquisa foi feita a partir dos seguintes pressupostos: Devido às elevadas taxas de amostragem usadas na aquisição das imagens, os deslocamentos das marcas de imagem para imagem serão pequenos em amplitude e,

tendo em conta a existência da inércia, nenhum objecto que se encontre em movimento poderá mudar “repentinamente” de direcção de uma imagem para a seguinte [Mourão, 1999]. Por outro lado, e como consequência das características do movimento das marcas serem diferentes, devido à disposição das câmaras (Figura 1), à semelhança da área de referência, também a escolha das dimensões da área de pesquisa foi feita de maneira distinta para cada uma das sequências, dependendo da câmara com que foram adquiridas.

Desta forma, as sequências, adquiridas por cada uma das câmaras, foram analisadas recorrendo a dois algoritmos de *Block Matching* distintos (o mesmo algoritmo com diferentes parametrizações), que produziram as trajectórias 2D das marcas, as quais foram armazenadas em ficheiros para posteriormente serem usadas na estimação 3D do seu movimento.

Outro factor que pode influenciar a velocidade de processamento é a forma como é implementado o procedimento de procura na área de pesquisa [Mourão, 1999], assim, foi mantida a pesquisa em espiral usada por Mourão que foi aplicada à semelhança de [Mendonça, 1993]. A medida de avaliação usada para calcular o ponto de maior semelhança foi a Correlação Normalizada (CN) [Mourão, 1999].

2.3 Reconstrução 3D

As imagens são captadas por dispositivos de aquisição de imagem tais como câmaras fotográficas ou de vídeo. À semelhança do que acontece na visão humana e em grande parte dos sistemas de visão naturais, o processo de formação de uma imagem inicia-se quando as amostras de intensidade luminosa, reflectida pelos objectos da cena, atingem os sensores da câmara (pupila). Ou seja, uma imagem é uma função bidimensional da intensidade luminosa [Niblack, 1986].

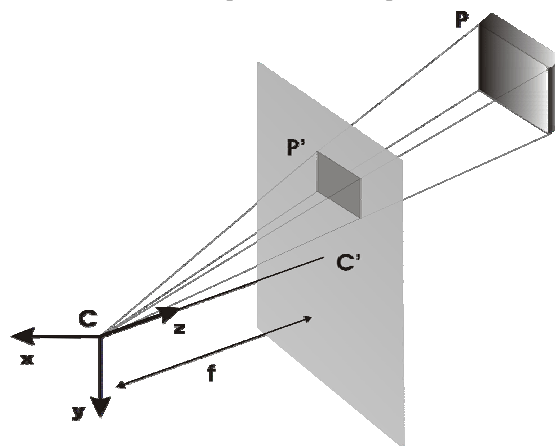


Figura 3 – Modelo perspectivo

A imagem criada pela câmara representa uma projecção bidimensional (2D) da cena tridimensional (3D) visualizada. O processo de aquisição da imagem pode ser assim entendido como uma transformação, na qual um modelo 3D dá origem a um modelo 2D, a imagem. O grande objectivo da reconstrução 3D é recriar (estimar) o modelo 3D que deu origem à projecção 2D.

O processo de aquisição da imagem tem implícita uma transformação geométrica, através da qual se relacionam os pontos da cena com os seus correspondentes na imagem. Para se poder efectuar a reconstrução 3D da cena é necessário perceber esta transformação geométrica, associada ao processo da aquisição da imagem, pois, o que se pretende é efectuar a transformação inversa.

O modelo adoptado neste trabalho é o modelo perspectivo (Figura 3) ou *pinhole model*, o qual pressupõe que a câmara executa uma projecção perspectiva. Neste modelo a relação entre as coordenadas imagem de um ponto e as coordenadas 3D desse mesmo ponto é definida pela equação [Faugeras, 1993]:

$$\tilde{p} \cong M \cdot \tilde{P} \quad (1)$$

com

$$M = K \cdot (R \quad -R \cdot t) \quad (2)$$

em que $\tilde{p}$ e $\tilde{P}$ correspondem, respectivamente, às coordenadas homogéneas do ponto imagem e do ponto 3D.

A matriz K contém os parâmetros internos da câmara, isto é, aqueles que só dependem das características da câmara. Esta matriz tem a seguinte composição:

$$K = \begin{bmatrix} f \cdot ku & -f \cdot ku \cdot \cot g(\theta) & u_0 \\ 0 & \frac{f \cdot kv}{\sin(\theta)} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

em que ku e kv representam respectivamente o factor de escala em cada um dos eixos da imagem, θ o ângulo entre os eixos, f a distância focal e u_0 e v_0 as coordenadas do centro de projecção ou de perspectiva.

A matriz R e o vector t formam os parâmetros externos da câmara, os quais definem, respectivamente, a orientação e a posição da câmara em relação a um referencial global.

Para ser possível a reconstrução tridimensional são necessárias duas ou mais vistas da cena obtidas, a partir de posições diferentes, e conhecer as matrizes de projecção para cada uma das vistas. O conhecimento das matrizes de projecção implica a obtenção dos parâmetros internos e externos da câmara. Numa abordagem convencional para a reconstrução tridimensional são executados os seguintes passos: em primeiro lugar são obtidos os parâmetros internos da câmara, de seguida são determinados os parâmetros externos e só depois é possível a reconstrução tridimensional (triangulação).

Neste trabalho, e uma vez que as câmaras se mantêm estáticas, o processo de calibração permite-nos obter os parâmetros internos e os externos. Na verdade só estamos interessados na obtenção da matriz de projecção para cada uma das câmaras, as quais se supõe que se mantêm constantes ao longo do tempo. Uma vez obtidas as matrizes de projecção para cada câmara e os pares de pontos imagem correspondentes, para cada ponto (marca)

é estimada a sua posição em relação às câmaras usando uma técnica que se designa por triangulação.

2.3.1 Calibração

No âmbito do trabalho desenvolvido neste trabalho, o termo calibração designa a tarefa de obtenção de um conjunto de parâmetros que caracterizam a câmara, normalmente designados por parâmetros internos e externos.

Se forem conhecidas as coordenadas 3D de alguns pontos da cena é possível obter a matriz de projecção para cada vista, desde que se encontrem visíveis, em cada vista, um número suficiente de pontos. Este número mínimo está directamente relacionado com o número de parâmetros a obter, ou seja, com o número de incógnitas do sistema de equações (5).

Esses pontos podem ser conhecidos a priori, sendo designados na fotogrametria por pontos de controlo, ou fazendo parte de um objecto de calibração do qual se conhece a estrutura ([Faugeras, 1987b] e [Tsai, 1986]).

Vamos supor que a distorção radial e a tangencial podem ser consideradas desprezáveis. Desta forma a relação entre cada ponto $P_i = (x_i \ y_i \ z_i)^T$ e a sua projecção $p_i = (u_i \ v_i)$ é definida por

$$s \cdot p_i = M \cdot \tilde{P}_i \quad (4)$$

em que s representa um factor de escala, M a matriz de projecção e $\tilde{P}_i$ as coordenadas homogéneas do ponto P_i .

Esta relação pode ser escrita, após algumas manipulações matemáticas, na forma matricial

$$B_i \cdot m = 0 \quad (5)$$

com

$$B_i = \begin{pmatrix} -x_i & -y_i & -z_i & -1 & 0 & 0 & 0 & 0 & u_i x_i & u_i y_i & u_i z_i & u_i \\ 0 & 0 & 0 & 0 & -x_i & -y_i & -z_i & -1 & v_i x_i & v_i y_i & v_i z_i & v_i \end{pmatrix} \quad (6)$$

e

$$m = (m_1 \ m_2 \ m_3 \ m_4 \ m_{21} \ m_{22} \ m_{23} \ m_{24} \ m_{31} \ m_{32} \ m_{33} \ m_{34})^T \quad (7)$$

em que m_{ij} representa o elemento na posição (i,j) da matriz M .

Conhecendo um número suficiente de coordenadas P_i e a sua projecção p_i é possível resolver o sistema de equações (5) (de preferência o sistema de equações deverá ser sobredeterminado uma vez que um maior número de pontos originará, à partida, um menor valor do erro na obtenção dos parâmetros) e determinar (7). Este problema foi abordado por Abdel-Aziz e Karara [Abdel-Aziz, 1971], donde resultou um método conhecido por DLT (*Direct Linear Transformation*). Este método foi mais tarde revisto por diversos autores (ex: [Faugeras, 1987, Melen, 1994]).

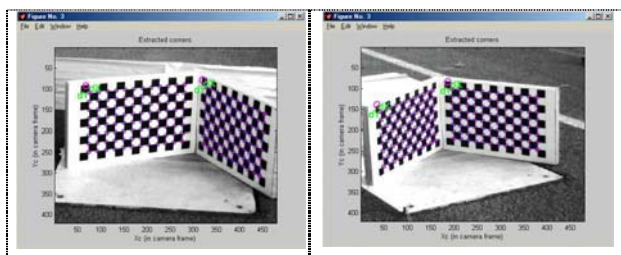


Figura 4 - Procedimento para a calibração das imagens

Para calibrar as câmaras foi desenvolvida uma aplicação em Matlab com base nas aplicações de Bouguet [Bouguet, 2001] e de Pribanic [Pribanic, 1999]. Primeiro é necessário identificar nas imagens alguns dos pontos característicos do objecto de calibração, mais concretamente, os cantos dos quadrados que fazem parte do padrão do objecto (Figura 4). Posteriormente são fornecidas ao programa as coordenadas (x,y,z) reais dos referidos pontos do objecto de calibração, a partir das quais são determinadas as matrizes de projecção que deram origem às imagens.

2.3.2 Triangulação

Designa-se por triangulação o processo de reconstruir espacialmente um ponto P a partir das suas projecções p1 e p2, em duas vistas diferentes, e conhecendo as matrizes de projecção para cada uma das câmaras (podem ser câmaras diferentes ou a mesma câmara mas em posições diferentes). Geometricamente este processo resume-se à determinação do ponto de intersecção de duas rectas (ver figura 5), a recta que passa pelo centro de projecção C1 e pela projecção p1 e a recta que passa pelo centro de projecção C2 e pela projecção p2, o que é um problema de fácil resolução.

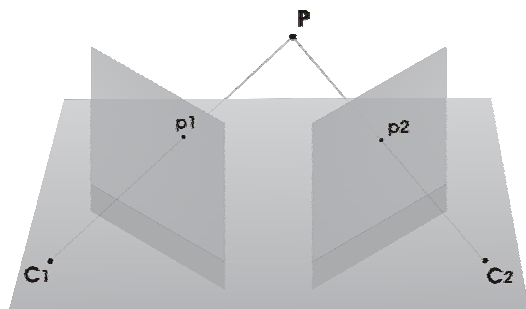


Figura 5 - Triangulação

Contudo, devido ao ruído introduzido pelos erros cometidos na obtenção das coordenadas das projecções p1 e p2, pelo facto de as matrizes de projecção serem unicamente aproximações, ou por erros cometidos pelos arredondamentos nos cálculos, o problema deixa de ter uma resolução simples, uma vez que as duas rectas podem não se intersectar. Torna-se então necessário usar métodos de triangulação, que nos permitam obter a solução que minimiza um determinado erro.

Uma possível solução pode ser encontrada na análise numérica, a qual apresenta métodos para a resolução de sistemas de equações lineares tendo em conta a minimização de um determinado erro. O sistema de equações pode ser resolvido, usando o método dos mínimos quadrados. Em Matlab é possível resolver um

sistema de equações sobredeterminado, que é o caso, usando o método dos mínimos quadrados.

2.4 Visualização 3D

Com base na informação da posição das marcas para cada instante é possível efectuar uma animação do movimento tridimensional do corpo (neste caso das marcas). Para visualizar o movimento 3D do corpo articulado foi desenvolvida uma aplicação em OpenGL que apresenta para cada instante a posição das marcas (ver figura 6). Esta aplicação permite mudar a posição da câmara de visualização enquanto se observa a animação, fornecendo uma ferramenta de análise do movimento tridimensional do corpo articulado.

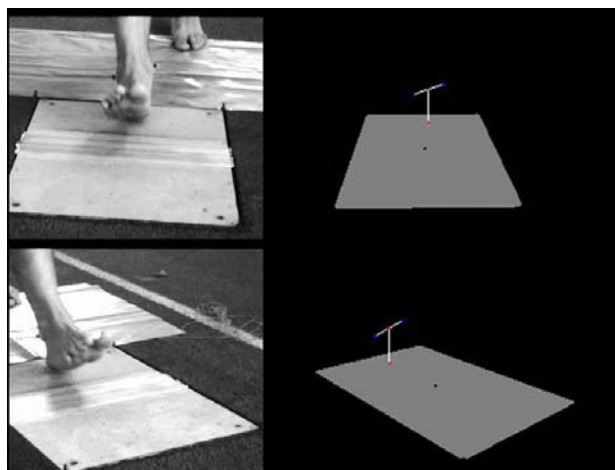


Figura 6 – Visualização do movimento 3D em OpenGL

3. TRABALHO EXPERIMENTAL

Pretende-se aplicar a metodologia proposta ao estudo do comportamento de estabilidade articular, feita através da análise de duas sequências de imagens de alta resolução temporal, que torne possível a caracterização tridimensional do movimento.

O parâmetro biomecânico que se pretende caracterizar é o comportamento 3D do centro articular do tornozelo do indivíduo, durante a passada normal. Assim, torna-se necessário a colocação de marcas no executante da tarefa, que serão posteriormente identificadas em cada uma das imagens das sequências.

A colocação das marcas nos executantes foi feita de acordo com as recomendações do comité de terminologia e standardização (STC – *Standardization and Terminology Committee*) da sociedade internacional de biomecânica (ISB – *International Society of Biomechanics*) no que respeita ao sistema de coordenadas das articulações (JCS – *Joint Coordinate System*) [jbiomech, 2002], [Ying, 2002]. Assim, foram colocadas duas marcas no tornozelo do indivíduo, uma no maléolo interno (MM – *medial malleolus*) e outra no maléolo externo (LM – *lateral malleolus*) (ver figura 7). Esta disposição permite o posterior cálculo do centro articular do tornozelo (IM – *inter-malleolar point*), localizado no ponto médio do segmento de recta definido pelos pontos MM e LM [jbiomech, 2002], [Ying, 2002].

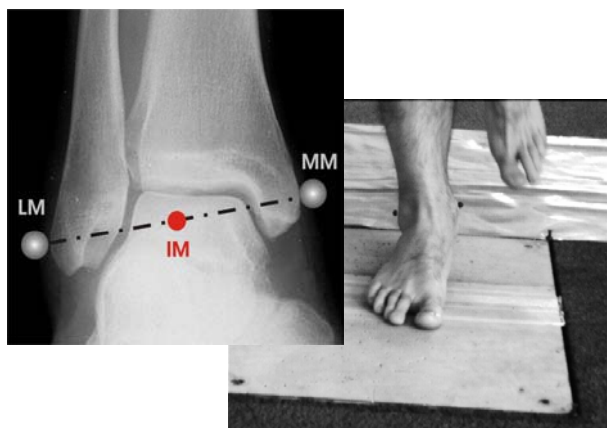


Figura 7 – Colocação das marcas

Para a aquisição das sequências de imagens foram utilizadas duas câmaras de alta resolução temporal REDLAKE PCI 500S, pertencentes ao departamento de Engenharias da Universidade de Trás-os-Montes e Alto Douro. Estas câmaras permitem adquirir imagens com dimensões e taxas de aquisição apresentadas na tabela 1.

Taxa de aquisição (imagens/seg)	50	125	250	500
Resolução do sensor (pixels)	480x420	480x420	480x420	320x280

Tabela 1 – Relação Taxa de aquisição versus resolução do sensor

Como já foi referido, as sequências, adquiridas à habitual taxa de 25 imagens por segundo, mostram-se insuficientes para caracterizar o comportamento do movimento humano, durante a locomoção bípode com velocidade e cadência de passada natural. Em princípio, escolher-se-ia a taxa de aquisição mais elevada para evitar uma possível ocorrência de *aliasing*. Contudo, existem vários factores que condicionam esta escolha, tornando-se necessário encontrar uma solução de compromisso de maneira a facilitar o posterior processamento de imagem, uma vez que estes se revelam determinantes no que respeita à qualidade das imagens.

Assim, a escolha da taxa de aquisição a utilizar foi baseada nos seguintes critérios:

- Quanto maiores forem as dimensões, em *pixels*, da imagem, mais fácil e eficaz se torna a sua análise, uma vez que estas determinam a resolução espacial da mesma.
- As taxas de aquisição mais elevadas acarretam dois tipos de problemas: por um lado, quanto maior é a taxa de aquisição menor é a resolução da imagem, por outro lado, maiores serão os problemas relacionados com as exigências nas condições de iluminação.

Após vários testes preliminares, concluiu-se que o uso de taxas de aquisição de 250 e 500 imagens por segundo implica o recurso a um complexo sistema de iluminação, pois quanto maior for a taxa de aquisição menor será o tempo de exposição do sensor para cada imagem. O recurso a sistemas de iluminação comuns revelou-se

ineficaz, uma vez que a estas velocidades de aquisição, até mesmo as flutuações na intensidade das lâmpadas, devidas à natureza alternada da rede de alimentação (corrente alternada a 50Hz), são visíveis e provocam diferenças de luminosidade, acentuadas de imagem para imagem. Aconselha-se o uso da fibra óptica ou de lâmpadas fluorescentes de alta-frequência.

A taxa de amostragem escolhida para a realização deste trabalho, segundo os critérios supra mencionados, foi de 125 imagens/segundo, garantindo a não ocorrência de *aliasing* nos dados e, também, porque a esta velocidade não é necessário o recurso a iluminação artificial. Assim sendo, as imagens das diferentes sequências são de dimensões 480x420 *pixels* com 8 bits de resolução (256 níveis de cinzento).

As sequências foram adquiridas na Nave de Desportos da Universidade de Trás-os-Montes e Alto Douro sem o recurso a iluminação artificial.



Figura 8 – Disposição das câmaras

Para guardar as sequências de imagens, foi utilizado um computador pessoal IBM-PC Pentium III 866 MHz, com sistema operativo windows, equipado com 512 Mbytes de RAM e dois discos rígidos de 40 Gbytes cada.

As câmaras foram dispostas como se indica na Figura 8, encontrando-se sincronizadas, entre elas e a plataforma de forças (KISTLER 9281-B), por intermédio de um sinal de *trigger*. No âmbito deste trabalho, os dados adquiridos pela plataforma de forças serviram apenas para comprovar a imagem das sequências em que ocorre o contacto inicial do pé do indivíduo com a plataforma.

As câmaras foram calibradas com um objecto de calibração tridimensional (dois planos ortogonais, com um padrão de xadrez 7x11 quadrados, em que cada quadrado tem as dimensões de 2x2 cm. Ver Figura 9).

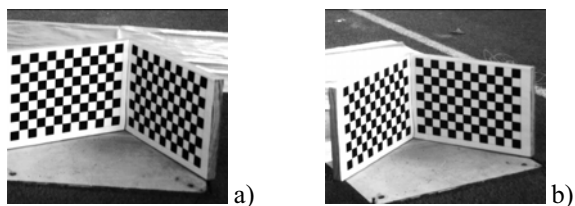


Figura 9 - Objecto de calibração "visto" pelas câmaras
a) *master* b) *slave*

O recurso a elevadas taxas de amostragem (125 imagens/segundo) coloca o problema da localização das marcas, colocadas nos executantes, em todas as imagens da sequência. Com efeito, sendo cada sequência composta por 360 imagens e sendo necessário identificar 2 marcas em cada imagem, esta tarefa, quando executada manualmente por um operador (teria de marcar 720 pontos), seria um processo moroso, cansativo e também sujeito a erros. Assim, este trabalho é levado a efeito, de uma forma semi-automática (só é necessário marcar os pontos na 1ª imagem), pela aplicação desenvolvida no âmbito deste trabalho. Esta aplicação permite o seguimento semi-automático das marcas colocadas nos membros inferiores dos indivíduos.

A segunda aplicação desenvolvida faz a estimação 3D da posição relativa das marcas, a partir de cada par de coordenadas 2D correspondentes a duas imagens adquiridas no mesmo instante de tempo. Posteriormente, toda a trajectória 3D do centro articular do tornozelo é caracterizada.

A terceira aplicação é utilizada para visualização e análise de resultados.

4. RESULTADOS

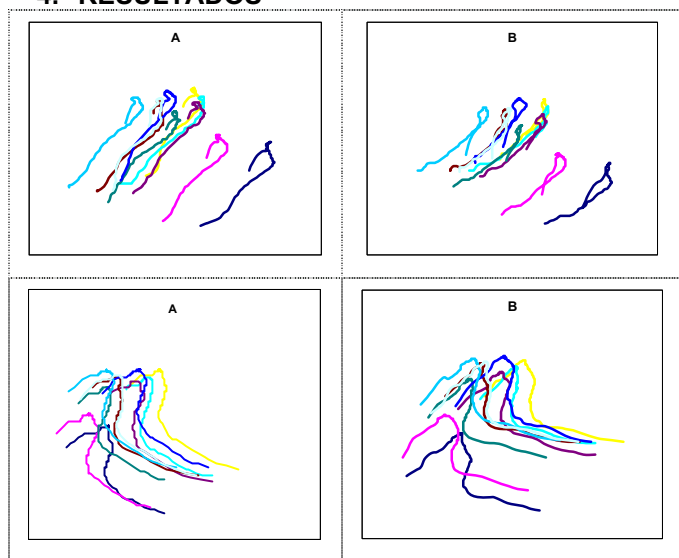


Figura 10 – Trajectórias 2D das marcas

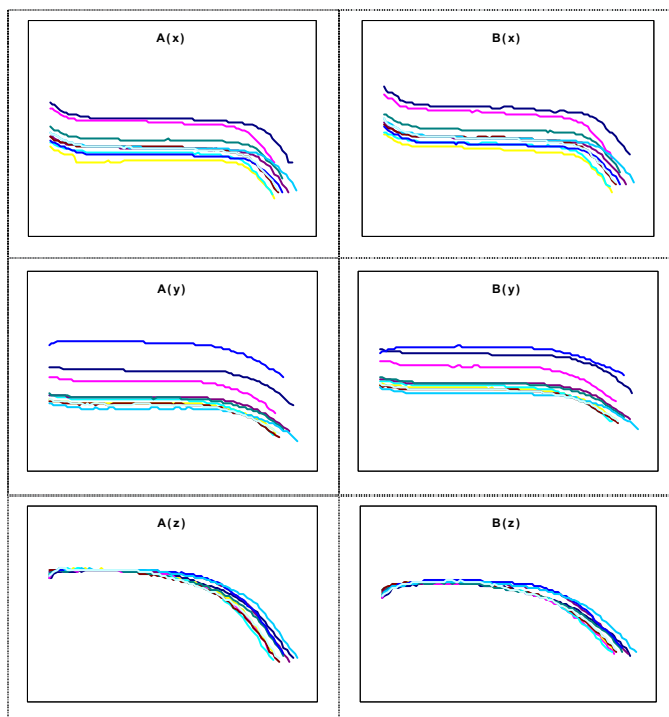


Figura 11 – Coordenadas x, y e z das marcas A e B ao longo das sequências

Neste ponto são apresentados alguns dos resultados obtidos na determinação das trajectórias 2D das marcas por aplicação de técnicas de *Block Matching*.

Na figura 10 pode ser visto na 1ª linha: gráficos (x,y) das trajectórias, das marcas A e B, obtidas a partir da análise das sequências adquiridas pela câmara *master*; 2ª linha: gráficos (x,y) das trajectórias obtidas a partir da análise das sequências adquiridas pela câmara *slave*.

Neste ponto são apresentados, também, alguns dos resultados obtidos na determinação das trajectórias 3D das marcas.

Na figura 11 pode ser visto o comportamento das coordenadas (x,y,z) das marcas A e B, coordenadas essas que definem as trajectórias 3D das mesmas.

Na figura 12 podem ser vistas, num gráfico (x,y,z), as trajectórias 3D do centro articular do tornozelo. As coordenadas do centro articular do tornozelo foram determinadas, achando o ponto médio do segmento de recta que une as duas marcas.

Observando as figuras 11 e 12 facilmente se verifica que não existem diferenças significativas nas trajectórias 3D das marcas e do centro articular entre cada uma das passagens. Por outro lado, a distância entre as marcas manteve-se constante ao longo do tempo e com valores coincidentes com os valores obtidos através de medição. Estas observações apontam para a fiabilidade da metodologia proposta.

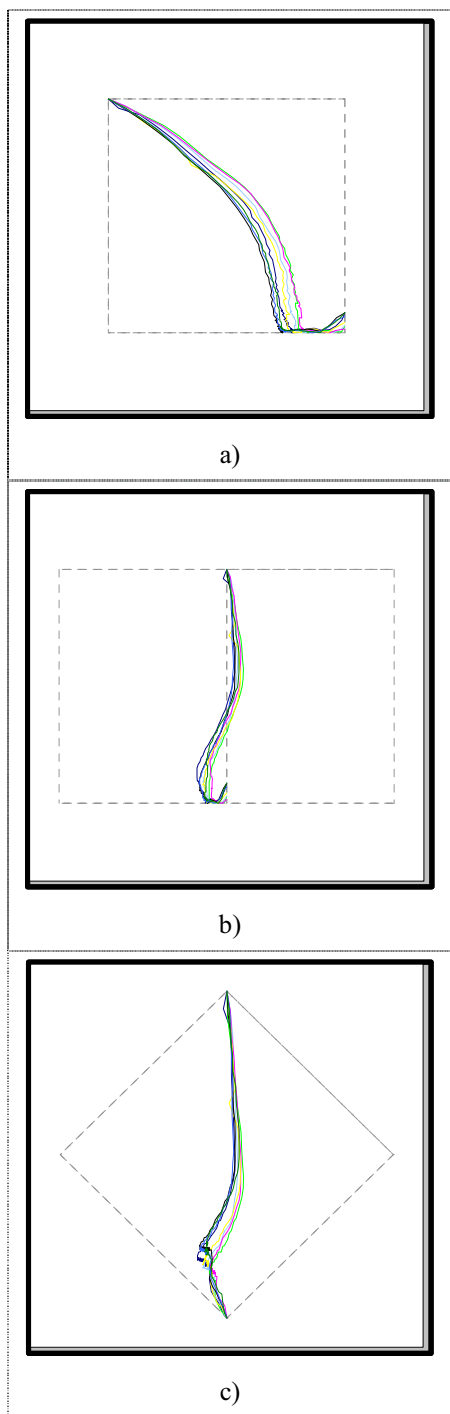


Figura 12 – Trajectórias 3D do centro articular do tornozelo

a) Vista lateral; b) vista anterior; c) vista superior

A aplicação desenvolvida em OpenGL destina-se a permitir a visualização de resultados num ambiente tridimensional. Note-se que a trajectória visualizada é a trajectória do centro articular do tornozelo e não das marcas.

Na figura 13 pode-se ver a referida trajectória, relativa a uma passagem do executante da tarefa, de uma perspectiva lateral.

A figura 14 mostra a mesma trajectória que a figura anterior, vista de cima.

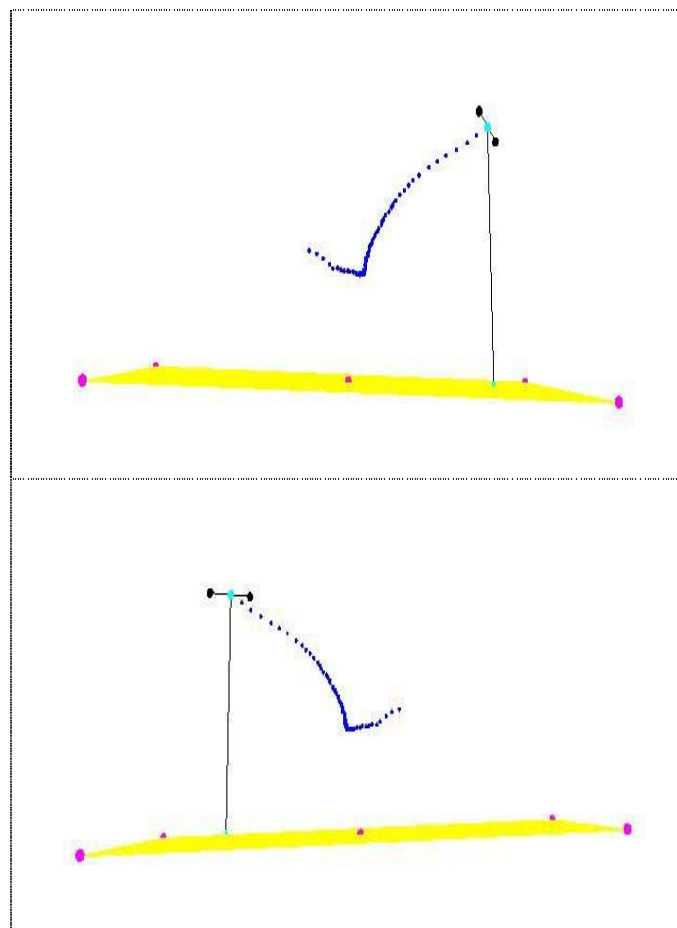


Figura 13 – Trajectórias 3D do centro articular do tornozelo (perspectivas direita e esquerda do plano sagital)

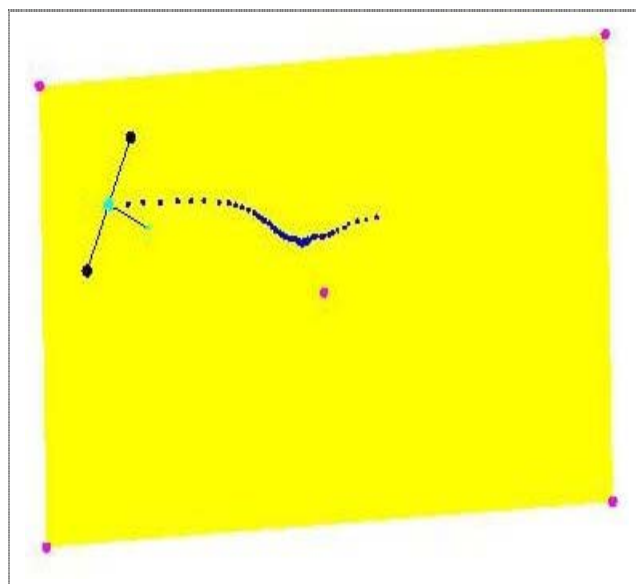


Figura 14 – Trajectória 3D do centro articular do tornozelo (plano horizontal)

Na figura 15 pode-se observar a trajectória do centro articular do tornozelo, relativa à mesma passagem da figura anterior, de duas perspectivas, anterior e posterior do plano frontal.

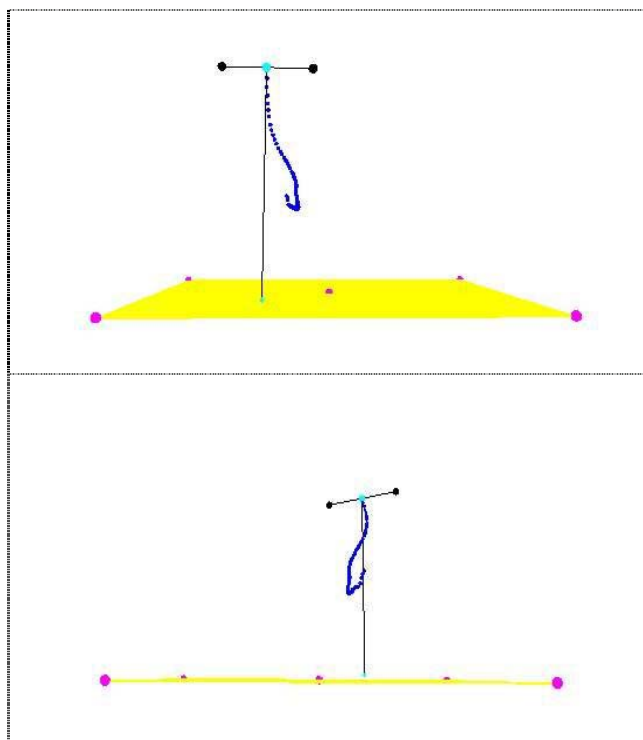


Figura 15 – Trajectórias 3D do centro articular do tornozelo (perspectivas anterior e posterior do plano frontal)

5. CONCLUSÕES E TRABALHO FUTURO

Neste trabalho foram desenvolvidas três aplicações, que em conjunto permitem a caracterização biomecânica do movimento 3D de um corpo articulado. Uma primeira aplicação faz o seguimento de objectos em imagens de alta resolução temporal, a segunda estima a posição 3D dos mesmos, e a terceira permite a visualização do movimento dos pontos num ambiente tridimensional.

A metodologia foi aplicada com sucesso no caso da caracterização 3D do movimento articular do tornozelo, constituindo assim uma ferramenta auxiliar para a respectiva caracterização biomecânica. Os resultados obtidos nessa experiência mostram a viabilidade da metodologia proposta.

De referir, a relevância da metodologia desenvolvida, na medida em que permite ter acesso à análise biomecânica, no caso particular, do comportamento do tornozelo, mais especificamente do seu centro articular. Este ponto é de extrema importância na medida em que possibilita a execução dos cálculos posteriores dos momentos articulares naquela articulação, cálculos estes indispensáveis para se ter acesso ao estudo de estabilidade articular.

Os métodos desenvolvidos poderão ser facilmente adaptados para a avaliação funcional de próteses dos membros inferiores/superiores, tendo em vista a realização de uma ferramenta capaz de auxiliar o especialista de reabilitação na avaliação da adequação, e eventual ajuste, desse tipo de próteses.

Como trabalho futuro, julgamos ser necessário o desenvolvimento de metodologias que permitam uma

maior compreensão dos resultados em termos biomecânicos. Também julgamos ser necessária a evolução da metodologia desenvolvida para um maior número de câmaras utilizadas, como forma de, simultaneamente dar ainda maior robustez ao método desenvolvido e permitir alargar o campo de estudo em termos de objectivos biomecânicos.

6. REFERÊNCIAS

- [Abdel-Aziz, 1971] Abdel-Aziz, Y. I. e Karara, H. M. - "Direct Linear Transformation Into Object Space Coordinates in Close-Range Photogrammetry" - In *Proc. Symposium on Close-Range Photogrammetry*, Urbana, Illinois, USA, p. 1-18, 1971.
- [Bouguet, 2001] Bouguet, Jean-Ives - Camera Calibration Toolbox for Matlab, Março de 2001.
http://www.vision.caltech.edu/bouguetj/calib_doc
- [Couto, 2003] Couto PA - Visão por Computador no estudo 3D do comportamento de estabilidade articular - Tese de Mestrado, Universidade de Trás-os-Montes e Alto Douro, 2003.
- [Faugeras, 1993] Faugeras, O. - Three-dimensional Computer Vision: a geometric viewpoint - MIT Press, Cambridge, Massachusetts, 1993.
- [Faugeras, 1987a] O.D. Faugeras, F. Lustman, and G. Toscani. - Motion and structure from point and line matches. - In *Proceedings of the International Conference on Computer Vision*, London, England, pages 25-34. IEEE Computer Society Press, June 1987.
- [Faugeras, 1987b] O.D. Faugeras and G. Toscani. - Camera calibration for 3D computer vision. - In *Proceedings of International Workshop on Machine Vision and Machine Intelligence*, Tokyo, Japan, 1987.
- [Fleet, 1992] Fleet, D.J. - Measurement of Image Velocity - Kluwer Academic Publishers, Boston, 1992.
- [Gabriel, 1998] Gabriel, R.; Mourão, A.; Filipe, V.; Filipe, S.; Melo, P.; Bulas-Cruz, J.; Abrantes, J. - "A method for automatic relocation of skinn markers in rearfoot motion analysis" - In: Hartmut J. Riehle & Manfred M. Vieten (Eds), *Proceedings I of ISBS'98*, University of Konstanz, Germany, 264-465, 1998.
- [Gabriel, 1999] Gabriel, R. - Biomecânica da estabilidade articular. Análise do comportamento do tornozelo na mudança de direcção do caminhar – Tese de Doutoramento, Universidade de Trás-os-Montes e Alto Douro, 1999.
- [jbiomech, 2002] ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion – Part I: ankle, hip, and spine – In *Journal of Biomechanics*, 35, 543-548, 2002.
- [Krebs, 1995] Krebs, David E. - Interpretation of Standards in Locomotor Studies in Human Gait - in Rebecca L. Craik & Carol A. Oatis, *Gait Analysis*, Mosby, St. Louis, Missouri, USA, 1995.

- [Melen, 1994] Melen, T. - "Geometrical Modelling and Calibration of Video Cameras for Underwater Navigation" - Tese de doutoramento, Norges tekniske hogskole, Institutt for teknisk kybernetikk, 1994.
- [Melo-Pinto, 1995] Melo-Pinto, P.; Bulas-Cruz, J. e Couto C. - A aplicação de visão por computador à análise biomecânica - 2º Encontro Nacional do Colégio de Engenharia Electrotécnica, IST, Lisboa, 1995.
- [Mendonça, 1993] Mendonça, Ana M. - Métodos de tratamento digital de imagem - aplicação no arquivo, processamento e análise de imagem de oftalmologia - Tese de doutoramento, Universidade do Porto, 1993.
- [Mourão, 1999] Mourão, A. - Análise de imagem na caracterização Biomecânica do movimento dos membros inferiores - Tese de Mestrado, Universidade do Porto, Faculdade de Engenharia, 1999.
- [Niblack, 1986] Niblack, W. - An Introduction to Digital Image Processing - Prentice Hall International, UK, 1986.
- [Pribanic, 1999] Pribanic, Tomislav - *Matlab Function mdl1.m*, Outubro de 1999.
<http://isb.ri.ccf.org/software/movanal.htm>
- [Singh, 1991] Singh, A. - Optic Flow Computation: A Unified Perspective - IEEE Computer Society Press, USA, 1991.
- [Tsai, 1986] R. Y. Tsai - An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision - Proceedings, CVPR '86 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition Miami Beach, FL, June 22--26, 1986)", IEEE Publ.86CH2290-5, pp. 364-374, IEEE, 1986.
- [Ying, 2002] Ying, N.; Kim, W. - Use of dual Euler angles to quantify the three-dimensional joint motion and its application to the ankle joint complex - In Journal of Biomechanics 35, 1647-1657, 2002.

COMUNICAÇÕES CURTAS

Jogo Educacional para Telemóveis

Ana Prata

Susana Lobão

Mário Rui Gomes

Dep. Eng. Informática, Instituto Superior Técnico

Avenida Professor Cavaco Silva, 2780-990 Porto Salvo

{ana.prata, susana.lobao, mario.gomes}@tagus.ist.utl.pt

Sumário

A comunicação descreve os passos fundamentais da concepção e implementação de um jogo 3D para telefones inteligentes. O trabalho está a ser realizado no âmbito de um Trabalho Final de Curso e teve como intuito o desenvolvimento de um protótipo de um jogo educacional para dispositivos móveis. Para além da descrição do jogo são descritas as tecnologias escolhidas para a sua implementação.

Palavras-Chave

Educacional, Dispositivos Móveis, CLDC, J2ME, M3G, MIDlet, MIDP,

1. INTRODUÇÃO

A Sociedade da Informação do século XXI está, cada vez mais, dependente dos dispositivos móveis. Aqueles que começaram por ser telefones com poucas funcionalidade e com uma interface pouco atractiva mostram hoje um vasto leque de oportunidades.

Nesse contexto evolutivo a maior área da Indústria de Software, a do Entretenimento, tem vindo a adquirir um papel cada vez mais importante. Neste contexto, a opção de um Trabalho Final de Curso que, ouvidos os potenciais clientes, apostasse na Educação à Distância e no Entretenimento afigurou-se como uma decisão fácil de tomar.

1.1 A realidade nacional

Acreditamos que a existência de jogos educacionais para telemóveis, focados em conteúdos leccionados nas escolas, pode contribuir para o sucesso escolar dos alunos. O facto destes jogos serem desenvolvidos para telemóveis pode constituir uma mais valia, quer pela acessibilidade quer pela disponibilidade.

Desse modo, pretende-se despertar a curiosidade dos alunos para conteúdos que, à partida, são pouco atractivos, como os da Matemática, área que, em paralelo com a Física, é bastante fraca em Portugal.

1.2 Trabalhos relacionados

De uma maneira geral os jogos existentes para telemóveis são do género de arcade, puzzles, jogos de desporto, estratégia, corridas de carros, *first person shooter*, luta, *role-playing games* e outros. Descurando a parte educa-

cional que qualquer jogo pode conter, no que concerne ao género do jogo o tema educacional é pouco explorado. Encontram-se muito poucos jogos sendo que os que existem estão relacionados com jogos de palavras.

Ao nível Europeu existe um projecto denominado *Growing Interest in the Development of Science Education* (GRID), que visa melhorar o ensino das Ciências nas escolas e aumentar o interesse dos alunos por estas matérias. Um dos meios a ser utilizado para este fim é o desenvolvimento de conteúdos educacionais para dispositivos móveis. Presentemente ainda não existe nenhum conteúdo desenvolvido uma vez que esta iniciativa é muito recente.

2. DESCRIÇÃO DO JOGO

O jogo baseia-se num conjunto de salas que contêm desafios a serem superados pelo utilizador no menor tempo possível.

A **acção do jogo** desenrola-se em salas com vários tipos de objectos tais como as convencionais mobílias. O utilizador deverá percorrer cada sala à descoberta dos desafios que têm que superar para poderem aceder a outras salas. Cada sala constitui um nível do jogo.

Os **desafios** tomam a forma de respostas a perguntas de escolha múltipla, relacionadas com a Matemática.

Os desafios estão escondidos sob a forma de metáforas. Para que o utilizador possa identificar um objecto, como tendo associado um desafio, terá que aproximar-se dele. O jogo fornece uma realimentação multimodal, através da mudança de cor do objecto, do aparecimento da palavra “desafio” e de um som (ver figura 1).

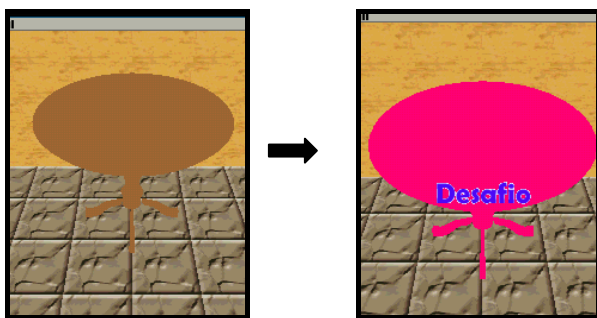


Figura 1: Indicação de desafio

O desafio que está associado à mesa é ilustrado na figura 2:

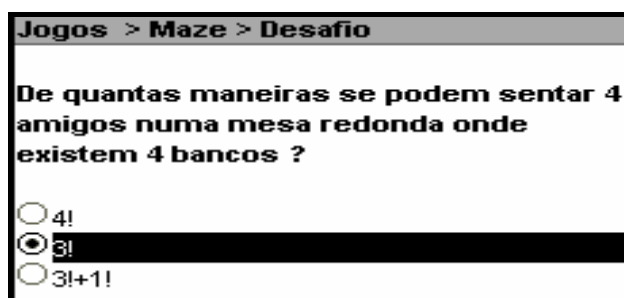


Figura 2: Desafio

O utilizador deverá superar todos desafios no menor tempo possível, sendo penalizado sempre que erra, tal como acontece com os exercícios de escolha múltipla, sendo usada a variável tempo e não a variável classificação numérica, usualmente de 0 a 20.

3. IMPLEMENTAÇÃO

A tecnologia baseada em máquinas virtuais foi um importante avanço, para a criação de aplicações cada vez mais eficientes e compatíveis com diferentes dispositivos. Para este trabalho a escolha da tecnologia recaiu numa das mais avançadas plataformas de desenvolvimento de aplicações para dispositivos móveis – *Java 2 Platform, Micro Edition da SUN*, mais conhecida por J2ME. Esta plataforma foi complementada com a tecnologia de M3G, API para gráficos 3D.

3.1 A Tecnologia utilizada

A plataforma J2ME suporta Java e um conjunto de tecnologias para o desenvolvimento de soluções móveis, como telemóveis, *palms*, *hand helds*, as quais possuem capacidade de memória e processamento limitados.

O J2ME contém configurações e perfis (*profiles*) normalizados e desenhados de modo a atingir um bom compromisso entre a portabilidade e o desempenho, para um vasto leque de dispositivos móveis. Actualmente o J2ME define duas configurações, cada qual com uma máquina virtual e uma **coleção de classes**:

- a *Connected Limited Device Configuration* (CLDC) define um ambiente de execução e um conjunto de APIs (*Application Program Interface*) específicas para ambientes extremamente restritos, com capacidade mínima de processamento, fonte de energia

limitada, *display* reduzido, conectividade intermitente entre outros requisitos não funcionais;

- a *Connected Device Configuration* (CDC) especifica um conjunto de tecnologias destinadas à utilização em dispositivos mais potentes que aqueles contemplados na configuração CLDC. Utilizando CDC pode-se construir interfaces gráficas muito mais poderosas e aplicações muito mais “inteligentes” e robustas em termos computacionais, se comparadas às aplicações feitas em CLDC.

Os **perfis** disponibilizam uma série de bibliotecas que auxiliam o desenvolvimento de aplicações para um tipo de dispositivos. O *Mobile Information Device Profile* (MIDP) é um perfil para dispositivos móveis ligado ao CLDC, com recurso de comunicação, como o telemóvel. O MIDP define funcionalidades como a utilização da interface do utilizador, o armazenamento persistente, suporte à rede e temporização. Uma aplicação conforme com o *standard* MIDP é apelidada de MIDlet.

Mobile 3D Graphics API (M3G), também conhecida como *Java Specification Request 184* (JSR-184), é um pacote adicional do J2ME que permite a renderização de gráficos 3D em dispositivos móveis. O M3G serve como uma ferramenta de gestão de cenas e animações e, inclui também um formato de ficheiro que simplifica bastante a integração de conteúdos 3D nas aplicações.

A figura que se segue ilustra a arquitectura tecnológica escolhida para este projecto.

J2ME

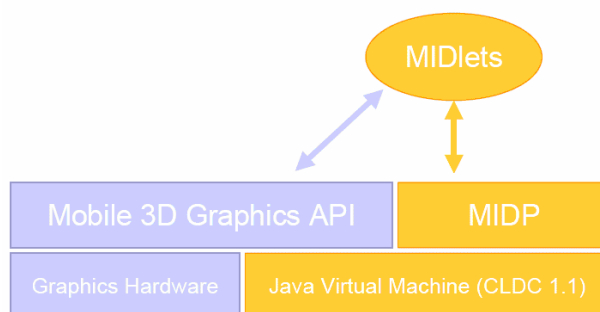


Figura 3: Arquitectura Tecnológica

Tratando-se as máquinas destino de dispositivos com capacidades mínimas de processamento, entre outras limitações, a escolha da configuração recaiu sobre a CLDC.

3.2 Desenvolvimento

A ferramenta de desenvolvimento (IDE) utilizada neste projecto é o Eclipse 3.0.1. O seu *plugin* EclipseME 0.5.0 permite que sejam desenvolvidas MIDlets, para a plataforma J2ME, bem como o estabelecimento de comunicação com o emulador da SUN *J2ME Wireless Toolkit 2.2*. No capítulo 4, entraremos em mais detalhe no tema dos emuladores.

Como para o desenvolvimento de qualquer aplicação Java foi necessária a instalação da plataforma J2SE (1.4.2_05 JRE e SDK) da SUN, mas para este tipo de aplicações destinadas a dispositivos móveis, são necessárias ferramentas e APIs mais especializadas. Assim, para tirar partido da plataforma J2ME instalou-se o *J2ME Wireless Toolkit 2.2*, igualmente da SUN. Este pacote inclui a importante API M3G e um emulador que permite o *deployment* directo e sessões de *debug*. O *J2ME Wireless Toolkit 2.2* é em conjunto com o Eclipse a base para o desenvolvimento deste projecto.

Para que a compilação fosse independente do IDE utilizado, instalou-se o Apache Ant 1.6.2, bem como um conjunto de tasks Ant, destinadas ao desenvolvimento de aplicações *wireless* Java MIDP – Antenna 0.9.13.

3.3 A Modelação

Os conteúdos geométricos utilizados na aplicação são na sua maioria de autoria própria, para tal foi necessário recorrer a uma ferramenta de modelação 3D. A ferramenta escolhida foi o *3D Studio MAX 7*, não só pelas suas características de modelação, mas também por permitir exportar directamente as cenas no formato M3G.

Infelizmente ainda existem muitas limitações ao nível da exportação de objectos com materiais mais elaborados, assim sendo as texturas têm um importante papel no aspecto dos objectos criados. As imagens e texturas utilizadas, no projecto, foram criadas a partir da utilização da ferramenta para tratamento de imagem *PhotoShop CS*.

4. EMULADORES

Existem actualmente uma série de emuladores que são proporcionais em número aos dispositivos para MIDP disponíveis no mercado.

Idealmente, os emuladores deveriam comportar-se exactamente como o dispositivo que representam, mas torna-se difícil desenvolver um emulador com um grande grau de exactidão. Uma aproximação é criar um emulador a partir do *J2ME Wireless Toolkit* dando-lhe não só uma aparência diferente, mas também um comportamento diferente aos botões e até adicionar novos controladores de modo a representar o dispositivo pretendido. Outra aproximação é desenvolver codificação que emule o processador e outros *chips* do dispositivo móvel. O resultado é um emulador que tem um comportamento muito próximo do dispositivo que representa, mas torna-se uma aproximação difícil de alcançar e cara comparativamente com a anterior.

Os emuladores podem ainda ser classificados de conceptuais ou reais. Um emulador conceptual não representa um dispositivo específico, mas sim um certo tipo de dispositivos demonstrando as características gerais comuns a todos. O emulador da SUN o *J2ME Wireless Toolkit*, por exemplo, é um emulador conceptual que pode ser utilizado para representar uma variedade de dispositivos que suportam MIDP. Os emuladores reais pretendem ter o mesmo comportamento do dispositivo que representam, suportando algum ou todo o código binário que se executa no dispositivo.

Neste momento, para o desenvolvimento do protótipo do jogo, está a ser utilizado o emulador da Sun, o *J2ME Wireless Toolkit 2.2*, mas pretende-se utilizar, até ao fim do trabalho que ocorrerá em meados de Setembro, outro emulador, que represente um dispositivo pré-seleccionado, para que no futuro o jogo se possa executar neste dispositivo.

5. CONCLUSÃO E TRABALHO FUTURO

Este artigo descreveu, sucintamente, o desenvolvimento de um jogo educacional para telemóveis. Na fase de concepção teve um papel determinante a colaboração de docentes do Ensino Secundário.

Tentando aproveitar uma oportunidade de mercado, surgiu a ideia de desenvolvimento de um protótipo de um jogo educacional para telemóveis, especializado na aprendizagem da Matemática. Este pretende aliar o aspecto lúdico ao aspecto didáctico, incidindo a sua temática em conteúdos matemáticos que são aqueles onde os alunos sentem maior dificuldade.

Presentemente o desenvolvimento do jogo está praticamente acabado, faltando apenas a modelação de alguns objectos e desafios para adicionar a cada uma das salas existentes. A fase de testes nos diferentes emuladores é a que se seguirá após o desenvolvimento concluído para que se possa determinar qual o telemóvel mais adequado para executar o jogo.

Como potencial trabalho futuro no âmbito de novos Trabalhos Finais de Curso pretende-se propor a criação de uma maior interacção entre os vários utilizadores que estão a jogar, para que estes possam trocar impressões entre si, nomeadamente ajudarem-se mutuamente.

Está previsto também, uma futura inclusão de profissionais de áreas como a matemática e usabilidade, o que trará uma mais valia para o aperfeiçoamento do protótipo.

Será feita uma demonstração do jogo aquando da apresentação deste *short paper* no décimo terceiro *Encontro Português de Computação Gráfica*.

6. BIBLIOGRAFIA

[Amaro04] Amaro, Pedro. Game Development, 2004

<<http://pedroamaro.pt.vu/>>

[AnfyMobile03] Anfy Mobile. Original games, 2003.

<<http://www.anfymobile.com/products/original.htm>>

[Bradford05] University of Bradford. Games on mobile phones could improve science education across Europe, May 2005

<<http://www.brad.ac.uk/admin/pr/pressreleases/2005/grid.php>>

[Gamito04] Gamito, Carlos. Aplicação de Entretenimento em 3D para Dispositivo Móvel de Última Geração, 2004

[Hui04] Hui, Ben. First Encounter with 3D Studio MAX version 7 and M3G exporter, October 2004

[http://www.benhui.net/modules.php?name=Mobil
e3D&page=First-Encounter-3DStudio-
Exporter.html](http://www.benhui.net/modules.php?name=Mobil
e3D&page=First-Encounter-3DStudio-
Exporter.html)

[Knudsen03] Knudsen, Jonathan. MIDP Emulators, Sep-
tember 2003.

[http://developers.sun.com/techttopics/mobilit
y/midp/articles/emulators/](http://developers.sun.com/techttopics/mobilit
y/midp/articles/emulators/)

[Knudsen03] Knudsen, Jonathan. The J2ME Wireless
Toolkit 2.1, December 2003

[http://developers.sun.com/techttopics/mobilit
y/midp/articles/wtk21/](http://developers.sun.com/techttopics/mobilit
y/midp/articles/wtk21/)

[Midlet05] Midlet Review. Mobile Gamer

[http://www.midlet-
review.com/index?content=home](http://www.midlet-
review.com/index?content=home)

[ZGroup05] ZGroup Mobile. Educational Mobile Games,
2005.

[http://www.zgroup-
mobile.com/games_educational.html](http://www.zgroup-
mobile.com/games_educational.html)

InStory Client – a Browser for Spatial Narratives and Gaming Activities

Tiago Martins, Nuno Correia
Dep. de Informática, FCT/UNL
Monte de Caparica, 2829-516 Caparica
{tms.martins, nmc}@di.fct.unl.pt

Filipe Barrenho, Teresa Romão
Dep. de Informática, Univ. de Évora
Rua Romão Ramalho, 59, 7000-671 Évora
l14066@alunos.uevora.pt,
tir@di.uevora.pt

Abstract

This paper introduces the InStory client program, part of the InStory project, as a location-aware hypermedia browser. The client is capable of obtaining the user's absolute or relative geographic position in order to request and display relevant contextual data such as area maps, quizzes and various hypermedia documents, allowing also for the creation of digital memories within the system. Together with the server and content documents they form a system for mobile storytelling, where a user can both navigate and participate in the narrative while touring a physical space.

Keywords

Spatial storytelling, hypermedia, mobile devices, mobile entertainment

1. INTRODUCTION

Storytelling is one of the oldest methods known to mankind for disseminating culture. We've come a long way from the traditional campfire tales to the modern blockbuster movies and promising next steps are being taken towards interactive narratives that can be navigated in space. Recent advances in technology along with the widespread use of mobile devices (whether they're Personal Digital Assistants – PDA's, cellphones or something in between) facilitate spatial positioning of users and provide a channel to deliver context-related information to them. This article introduces the InStory client program as part of the InStory project [Correia05a] [Correia05b], which aims at defining a system for interactive narratives and gaming activities in physical spaces.

The client program is to be run on a mobile device (currently a PDA) and is responsible for ascertaining the user's position using one of two different positioning systems – the Global Positioning System or the Ekahau Positioning Engine [Ekahau] – or by explicit user command. It periodically informs the server of the actual user location and displays the hypermedia contents sent by the server as response to these position updates or to explicit user commands – such as following a link in a document. It will also allow a user to contribute to the story by uploading media to the server.

This paper is structured as follows: in the next section we present a short set of related approaches to mobile storytelling and gaming; next we introduce the InStory project in order to contextualize the client in this larger system and its architecture; after that we take on the client itself,

taking time to discuss the architecture, the positioning methods, the media upload module, the implementation and the issues it presents; finally we draw some conclusions and introduce future additions and possible improvements to the client.

2. RELATED WORK

The emerging mobile technologies took storytelling a step forward. A mobile, location-based, multimedia system is basically composed of: (1) Mobile Devices; (2) Location Technology; (3) Outdoor and indoor location devices (4) Wireless Networking; (5) Service Technologies and (6) Content Creation, which determines the attractiveness of the complete system. [Hightower01] presents an overview of the basic techniques used for location-sensing and a taxonomy of location system properties. The paper [Domnitcheva01] analyzes the means of representing spatial information. Also, several experiments have been made concerning location-based systems. TOI - Traveler's On-line Information System was one of the first mobile context-aware multimedia geo-referenced information system to be conceived [Dias96], providing an extensive set of functionalities useful to a visitor of a large city. The system is built upon digitized maps, and hypermaps are used to provide the user with spatially referenced multimedia information. Georgia Tech developed the CyberGuide [Long96]. The MIT Laboratory for Computer Science developed a location-support system for in-building, mobile, location-dependent applications [Nissanka00]. In [Yu-Chee01] a location and context aware tour-guide system is described. Lancaster Univ. and Univ. of Arizona developed a system that provides an electronic handheld guide to Lancaster [Davies01]. A system for

creating and participating in mobile context-sensitive mobile cinematic narratives is also being developed in MIT [Crow03]. It enables users to experience mobile cinema, taking them on a journey through the physical world while pieces of a story (in the form of media clips) appear on their handheld according to their movements and location.

3. THE INSTORY PROJECT

The InStory project aims at defining and implementing a platform for the development of narratives that are navigable in physical space. Thus, while touring a given physical space of cultural or natural interest (such as a museum or a natural landmark) users will be able to receive multimedia content on their mobile devices. This content offers complementary information about the specific location being visited, and may also propose activities (involving the physical space and possibly other users) that immerse the user in an interactive story/narrative running parallel to the real world. The intended application scope of such a system is threefold: tourism, as a way to enhance the visitor's experience and access to information; entertainment, such as a paper-chase or a treasure-hunt that leads the user through the actual physical space making her part of the story; and interaction with other users, as a mean to share experiences, information and digital memories (such as a picture taken with a mobile device).

The system is based upon a client-server architecture, with the communication between these two components being made through a wireless network by means of the HTTP protocol. The client program is able to ascertain the user location through GPS (Global Positioning System), Wi-Fi triangulation (using the Ekahau Positioning Engine) or explicit user positioning and periodically requests updates from the server based on the coordinates it obtains by any of the three different methods. This is done issuing a HTTP request to the server in which the positioning information is sent as a well-defined set of variables in the requested URL. The server runs a script to produce an XML document in response to the client's request, keeping track of world and user state in a database. The server also houses static documents, whether they are XML documents describing content to be presented (much in the same way a HTML document does) or media that will integrate the content. In addition, the server may also house the Ekahau Positioning Engine, responsible for the Wi-Fi triangulation.

The XML documents sent by the server obey to a structure defined by a set of XML schemas (which can be used for document validation) and can be divided in two main types: updates and content descriptions. Updates are sent to the client in response to update requests. These updates consist basically of a table-of-contents for the given location and game state, referencing one or more content description documents (that the client can then request through HTTP if it chooses to do so) and may carry textual server messages. Content descriptions are very much

like HTML documents in the sense that they describe the content page to be presented, referencing any media that composes the page and describing the way the media should be presented. The client parses these files using an appropriate library (see section 4.1) and obtains by HTTP any media the presentation may require.

The InStory project is being developed and tested in partnership with Quinta da Regaleira, Sintra, Portugal. This is a place of uncommon richness in architectural, scenic and cultural content and benefits of both the mystical traditions it represents and the local aura (the town of Sintra is well-known for its own foggy microclimate and tales of it being a "place of magic"). The peculiarity of this estate – which comprises a palace, a chapel, groves, towers and even natural caves and underground passages, among other landmarks – presents many benefits for interactive storytelling and gaming activities as well as many challenges, derived from the location's architectural and geographic nature.

4. THE INSTORY CLIENT PROGRAM

The InStory client runs on the user's mobile device (currently a PDA) and it could shortly be described as a location-aware hypermedia browser. Along with the basic functionality as a browser of hypermedia documents, the client is responsible for determining the user's current location, informing the server about it and selecting the relevant content sent to be presented, as well as allowing the user to create digital memories within the system. It possesses no native interface elements other than an (optional) toolbar, definable in an XML configuration file.

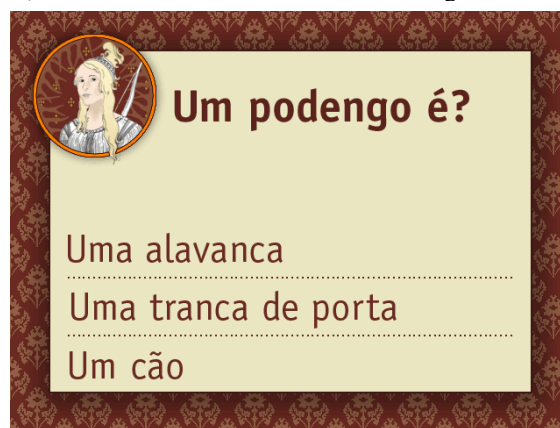


Fig. 1 – The InStory client presenting a quiz.

The client is designed to be as generic as possible in the sense that its interface is embedded in the content pages it displays – either local ones or those sent by the server – in order to allow the existence of different graphical environments and interfaces for different locations and/or narratives. Still, as a common base, the client can operate in one of five modes, which organize and somewhat define the client's behaviour relative to different types of content. These modes are:

- Options – this mode allows for the configuration of the client's behaviour, as well as providing access to

the app's help documents and also to exit the application; it primarily displays content stored in local files, so a connection to the server won't be needed for launching and configuring the client.

- **Player** – this mode is used to show information regarding player/user status such as her nickname, objects in her possession, current tasks or objectives and a log of her participation in the narrative, among others.
- **Game** – all story-relevant content, such as quizzes (Fig. 1), narrative related videos, interaction with virtual characters and so will be displayed in this mode.
- **Map** – this mode is used to display a map of the area where the user is currently located (Fig. 2); in map mode a user can also explicitly provide her position if both positioning systems (GPS and Ekahau) are unavailable.
- **Info** – not all content stored in the server is story-dependant; to explicitly access and browse this static information relative to the location being visited the client uses this mode.

Upon launching of the client, each of the modes has a default content file associated with it; these files are local and they're specified in the client's XML configuration file, in order to allow the app to initially present any of the modes even if the server is unavailable.



Fig. 2 – The InStory client in Map mode, displaying a tooltip; the tabs can be used to switch modes.

The client is currently being developed for a PDA (acronym for *Personal Digital Assistant*) running the Pocket PC 2003 platform. Even a device such as the one being currently used in developing (a HP iPAQ hx4700, which has a screen of 640x480 pixels) imposes some limitations and requires a measure of careful consideration in designing and implementing the program. As an example, polling the GPS antenna or rendering videos is power consuming, and care must be taken so that the user won't have to recharge the device every 30 minutes or so.

The range of handheld devices commercially available nowadays includes devices with distinct processing, audio

and rendering capabilities. This leads to a concern in future support for different devices, which will be made easier through the definition of an XML-based markup language to describe the InStory content and, in addition, possible XSLT transformations that produce content suitable for presentation on devices with different technical specifications.

4.1 Architecture

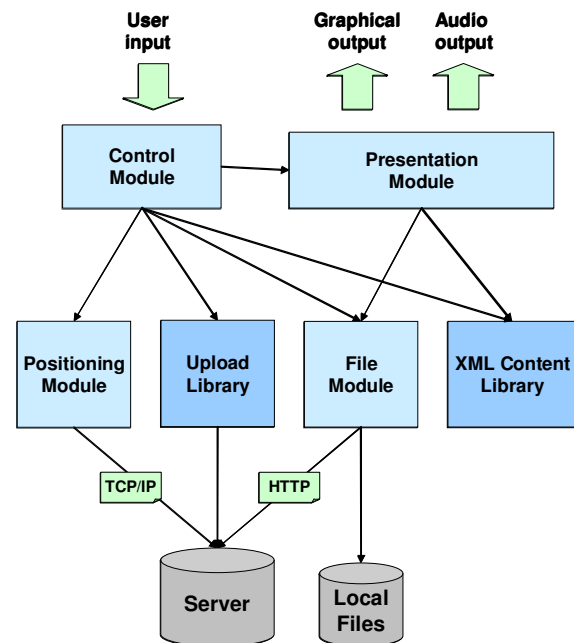


Fig. 3 – InStory client architecture

The current version (v2.0) of the InStory client is organized in modules and libraries (Fig. 3), both for tidiness and to facilitate the contribution of the many people involved in the project. Thus, and starting with the lower layer, we have:

- a module for (implicit) positioning, in charge of communicating both with the GPS receiver and the Ekahau Positioning Engine, covered in more detail in the next section;
- a library for uploading content to the server, allowing for the creation of digital memories, which is described in section 4.3;
- a file module which encompasses HTTP transaction handling, caching of files, conversion of remote filenames to local filenames and retrieval of local files;
- a library for accessing the XML elements in the content file, which is built upon the TinyXML open library, and is forcibly tied to the markup language that defines the contents.

The upper layer is composed of a control module, responsible for the application settings, control, bookkeeping and user input handling, and a presentation module that manages the sound and graphical output.

This modular architecture presents many benefits, one of them being reutilization, along with the aforementioned ones – tidiness and facilitated collaboration. Care has been taken to separate positioning, contributing (upload of files to the server – see section 4.3), file handling and content-parsing, anticipating changes in organization/behaviour of the client or even to make the libraries publicly available in the future.

4.2 Positioning Methods

It is the responsibility of the client to ascertain the user's location coordinates by any means available. In this project two positioning technologies are being used: the first one is GPS, the second Wi-Fi triangulation. When neither is available, the user is allowed to explicitly indicate her position by clicking a map (Fig. 1).

GPS is a system consisting of 24 satellites which can be used to accurately locate a receiver device almost anywhere on the planet's surface and orbit. For the positioning to succeed, the device must have a clear line of sight to at least four satellites; when it does, it can retrieve information about its latitude, longitude, elevation and the precise time. Obtaining a valid GPS fix may take (more or less) from two to five minutes, depending on the surrounding topography or weather conditions, and additional satellites in sight can be used. It is practically impossible to obtain a fix indoors or underground. GPS receivers use the NMEA protocol to output information [NMEA]. This RS-232 compatible protocol is based in the transmission of a well-defined set of sentences, some containing fix (positioning) information, others containing navigation data and some others informing of GPS status.

The Ekahau Positioning Engine (or shortly, EPE) is a software-based real-time location system for triangulation in Wi-Fi networks. It is fully compatible with a limited set of wireless devices and can also be used with programmable Wi-Fi tags. This system requires careful calibration, and may require recalibration periodically to accommodate network topography changes. For a device other than a tag to be tracked, it must first be running an Ekahau Client program in order to be visible to the engine. To communicate to the server one can use either the Ekahau Java SDK or the TCP-IP-based YAX protocol. Ekahau can work either indoors or outdoors, always depending on the number and strength of Wi-Fi access points. In order to obtain better positioning estimates the use of "dummy" access points is recommended (the more, the better).

One must acknowledge the fact that at any given time both systems may become unavailable. To prevent the user from being unable to access location-related content and progress in the narrative the client is able to identify this scenario and allow the user to indicate her current location by means of a map, presented in map mode. This requires the client to know the coordinates for the map, more precisely the coordinates for the upper-left and lower-right corners of the map, so that it can translate the

user's clicks on the interface to real coordinates. This information is sent in the content presentation files that describe a map screen, and it also allows the client to mark the user's current position on the map when at least one of the positioning systems is available.

4.3 Content Upload

During a tour a visitor may find small but relevant details which she would like to share with others. By doing so, one is entering a domain of interaction somewhat separate from answering hypertext quizzes or performing certain tasks proposed. In this domain, two kinds of interaction can be defined: User-System and User-User. The first occurs when a user affects the system by adding content to it - and the system may affect other users by choosing to display that content to them. This kind of interaction occurs when a user uploads information to the system, either to register a personal experience or to contribute to the story by adding new content. The user can also affect other users by increasing their awareness to a certain detail, and, as a consequence, those users can also end up affecting the system. The latter happens when particular information is shared between users. It is the user-contribution module that allows this contribution, thus promoting interaction. The module provides for both ways (which, nevertheless, share common aspects): one is the upload of distinct types of data to the system; the other is the ability to post targeted messages to specific users or groups of users. The latter isn't currently being used in the client, but will be made available in a future version.

The upload module can be considered as the soul of the user-contribution. It is comprised of two parts, the client (user) and the server (system), providing a way of establishing a communication link between them. The user can capture and upload several types of data, i.e., still or moving images (photographs and videos), sounds or voice annotations (audio), and text. As these data cannot be simply uploaded to the system, a process must be developed. Whether a user decides to register a personal experience or to contribute to the story, the uploading of the data is only the final stage of a life cycle's first part. That cycle starts on the client side and ends on the server side and begins once the user selects a theme and key-word, initiating the upload process. Contextualization of the upload will ease its integration into the story. On the other hand, when registering an experience, it will be useful for cataloguing and finding out what are the main interests of the person that made the upload. The data are saved after capture. Server, user, captured data as well as space and time-related information (i.e., server's IP address, username, type of request, filename and file path, coordinates, date and time of upload) is then added and an upload request is created. The user is now able to make the upload. A communication link is made and the request is sent to the system. Once that is completed, the captured data can be sent and the cycle now goes on to the server side. Upon receiving the information, the server processes it, checking the type of request. Being an upload request, the

captured data is received and written on the respective user directory. The client is acknowledged and the communication link severed. The processed information is simultaneously registered into a global system upload log and a specific user upload log. The cycle ends after the server updates the database with that same upload related information. The system now has new information from which data elements can be constructed and contextually integrated into the story thread.

Upon user request, the system is also able to generate a tour report in the form of a web page. This report describes the user expedition through the physical space and includes data concerning the path followed by the user, the visualized geo-referenced data and the uploaded data with reference to its location.

4.4 Implementation

The client is to be run on the Windows CE Pocket PC 2003 platform. While programming for WinCE may be very similar to programming for Win32, one must realize that the former is very limited by nature. Also, as mentioned before, additional care must be taken while developing for mobile devices due to their (somewhat) limited processing power and autonomy. Some aspects of the client's behaviour can be customized through its configuration file, written in XML.

The current version of the InStory client program was developed using MS Embedded C++ SP4. Much of the client's capabilities (such as playing sound clips) make use of the Shell API. Client-server communications are accomplished through HTTP, the client's nickname, position and other parameters being passed in the requested URL as HTTP GET variables whenever necessary. Communication with the Ekahau Engine is achieved by TCP/IP using the YAX Protocol, over the Winsock API. As for the GPS receiver, the app reads its NMEA-compliant sentences (see section 4.2) through one of the COM ports and these are then parsed using an NMEA parser class provided in [Variakojis02]. The XML content library is implemented resorting to the open TinyXML Library made public under the zlib/libpng License [TinyXML].

5. CONCLUSIONS AND FUTURE WORK

We have indeed come a long way since the campfire tales, and here we stand on the brink of new forms of storytelling which are interwoven with physical spaces, providing new vehicles for education and augmented tourism. The InStory project brings together several concepts and technologies in its approach to provide a system for spatial interactive narratives. This can be seen in the InStory Client, which combines generic browser capabilities with two distinct positioning technologies, all running in a mobile device in the user's palm.

While all this may sound right enough, there are some pitfalls to be avoided. One of the most important is requiring too much of the user's attention to be focused on the mobile device, consequently spoiling an otherwise

scenic stroll. While, on one hand, the graphical content is being carefully developed concerning design issues and bearing a strong connection to the test location (Quinta da Regaleira, Sintra, as mentioned before), an adequate use of sound in the client application can unobtrusively alert the user whenever new content is received from the server, so she won't have to be constantly looking at the device display or switching modes to check if something happened.

Although the version 2.0 of the client is mostly done, the client is considered work in progress, and this version doesn't yet provide all of the desired functionality, which will be added in the near future. The first one is playing video clips inside the client program, requiring good use of the Windows Active Template Library. The second one is messaging; the upload module (see section 4.3) already provides for user-to-user or user-to-group messaging and the client will soon provide an adequate interface for it.

6. ACKNOWLEDGEMENTS

The InStory project is partly funded by the Foundation for Science and Technology (FCT, Portugal) and POSI. We would also like to thank Glorianna Davenport from the MIT Media Lab by her role as consultant for the project; Rute Frias, in charge of designing the user interface and content for the system; Helena Barbas, for her contributions for the plot and challenges; Joaquim Jorge and Pedro Sousa from IST for their contributions and finally to Quinta da Regaleira/Cultursintra Foundation for their availability and help.

7. REFERENCES

- [Anderson92] R.E. Anderson. Social impacts of computing: Codes of professional ethics. *Social Science Computing Review*, 2 (Winter 1992), 453-469.
- [Correia05a] Correia, Nuno, et al., InStory: A System for Mobile Information Access, Storytelling and Gaming Activities in Physical Spaces, ACM SIGHI ACE 2005, Valencia, Spain, June 2005.
- [Correia05b] Correia, Nuno, et al., Narrativas Interactivas em Dispositivos Móveis, Artech 2005, Vila Nova de Cerveira, Portugal, August 2005.
- [Crow03] Crow, M., Pan, P., Kam, L., Davenport, G. M-Views: A System for Location-Based Storytelling. *Proceedings of Ubiquitous Computing (Ubicomp)*, Seattle, Washington, October 12-15, 2003, pp 31-34.
- [Davies01] Davies N., Cheverst, K., Mitchell, K. and Efrat, A. Using and Determining Location in a Context-Sensitive Tour Guide. *IEEE Computer*, August, 2001, pp. 35-41.
- [Dias96] Dias, Eduardo, Santos, Emanuel, Pimentão, João, Pedrosa, Pedro and Romão, Teresa Bringing Ubiquitous Computing Into Geo-Referenced Information Systems. In Rumor, M., McMillan, R., Ottens, H. (Eds.) "Geographical Information: From

Research to Application through Cooperation”, IOS Press, Amsterdam, 1996, pp. 100-103.

[Domnitcheva01] Domnitcheva, S. Location Modeling: State of the Art and Challenges. Workshop on Location Modeling for Ubiquitous Computing. Ubicomp 2001, Atlanta, Georgia, September 30- October 2, 2001.

[Ekahau] The Ekahau Positioning Engine
<<http://www.ekahau.com>>

[Hightower01] Jeffrey Hightower and Gaetano Borriello, Location Systems for Ubiquitous Computing, *Computer*, vol. 34, no. 8, pp. 57-66, IEEE Computer Society Press, Aug. 2001

[Long96] S. Long et al., Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study, Proc. Mobicom 96, ACM Press, New York, 1996, pp. 97-107.

[Nissanka00] Nissanka B. Priyantha, et. al., The Cricket Location-Support System, 6th ACM International Conference on Mobile Computing and Networking, August 2000.

[NMEA] National Marine Electronics Association, NMEA 0183 Standard For Interfacing Marine Electronic Devices, Version 3.01, January 1, 2002

[TinyXML] TinyXML Project, SourceForge.net
<<http://sourceforge.net/projects/tinyxml/>>

[Variakojis02] Variakojis, Monte, NMEA Parser Design, Rev. 0.2, November 2002, VisualGPS homepage 2002
<http://www.visualgps.net/white_papers.htm>

[Yu-Chee01] Yu-Chee et. al., Location Awareness in Ad Hoc Wireless Mobile Networks, IEEE Computer, Jun2001, pp 46-52.

Construção de Ambientes Virtuais – Um método de trabalho

Fernando Silva

Escola Superior de Tecnologia e Gestão
do Instituto Politécnico de Leiria
fernando.silva@estg.ipleiria.pt

Alexandrino Gonçalves

Escola Superior de Tecnologia e Gestão
do Instituto Politécnico de Leiria
alex@estg.ipleiria.pt

Resumo

Neste artigo apresenta-se um trabalho desenvolvido em colaboração com o Museu Monográfico de Conimbriga. Este consistiu na criação de uma reconstrução virtual da Casa dos Esqueletos de Conimbriga, um imponente conjunto arquitectónico da época romana de inegável valor histórico.

Dado que o modelo deveria ser disponibilizado via Web e tendo em conta as limitações de acesso criadas pela grande dimensão dos ficheiros criados em projectos similares, foram utilizadas, com base numa fundamentada e já testada metodologia de trabalho, diversas técnicas de optimização que levaram a que os ficheiros resultantes tenham dimensão reduzida, facilitando assim o seu acesso a todos os internautas.

Pode fazer a sua “visita” à Casa dos Esqueletos em: www.casadosesqueletos.web.pt.

Palavras-chave

VRML, Metodologia, Optimização, Desempenho, Dimensão dos ficheiros

1. INTRODUÇÃO

A vontade de alargar horizontes e conhecer cada vez mais o mundo que o rodeia, impulsionou o ser humano a expandir cada vez mais o seu território, desenvolvendo para tal meios de comunicação que lhe permitiram chegar cada vez mais longe. A Internet revolucionou o mundo e os hábitos de vida do Homem, tornando global o seu raio de acção. Hoje é possível aceder a conhecimento que há dezenas de anos se encontrava ao alcance apenas dos eruditos e abastados.

A necessidade de apresentar virtualmente ambientes e objectos reais levou ao desenvolvimento de tecnologias de representação tridimensionais, que tornaram mais apelativa a experiência que a Internet permite.

No entanto, a busca pelo perfeccionismo na representação virtual de determinados cenários tem alguns custos computacionais – tanto em performance, como em tamanho dos ficheiros. Este facto tornou visível o calcanhar de Aquiles da Internet – a latência. Quanto mais evolui a tecnologia, menos estamos dispostos a esperar por algo, a realidade virtual não é uma excepção. Ao querermos disponibilizar ao mundo a representação virtual de algo, devemos fazer um compromisso entre realismo, tamanho dos ficheiros e desempenho – factores críticos de sucesso em qualquer projecto desta natureza [Silva, 2004].

A *Virtual Reality Modeling Language* (VRML) – uma linguagem para representação de ambientes tridimensionais, permite conciliar estes três factores,

exigindo no entanto, que sejam aplicadas todas as mais valias que oferece.

Ao reconstruir virtualmente a Casa dos Esqueletos de Conimbriga, pretendeu-se aplicar uma série de técnicas de optimização que permitiram num projecto semelhante – Fórum Flaviano de Conimbriga [Gonçalves, 2003], obter resultados surpreendentes. No entanto a natureza deste modelo é bastante diferente da do Fórum – enquanto que no Fórum existe simetria em quase todo o cenário, na Casa dos Esqueletos a simetria é nula, o que dificulta a aplicação da maioria das técnicas e obriga a uma reengenharia das mesmas.

A Casa dos Esqueletos foi construída com um realismo bastante aceitável, segundo opinião expressa pelos especialistas, estando acessível a todos na Internet em www.casadosesqueletos.web.pt.

2. O CASO DE ESTUDO

Numa das maiores e mais prósperas cidades do antigo Império Romano em território “Lusitano” – Conimbriga, foi construída a chamada Casa dos Esqueletos.

A Casa dos Esqueletos, apesar das suas reduzidas dimensões (945 m²), é considerada um paradigma das residências de aparato de Conimbriga, uma vez que apresenta uma qualidade do plano arquitectónico considerável, economia dos meios, realce no que toca às decorações musivas e um aproveitamento inteligente da fracção autonomizável da construção. Com a construção

da muralha baixo-imperial, uma parte da fachada da casa foi destruída, o que dificulta de alguma forma a análise da mesma.

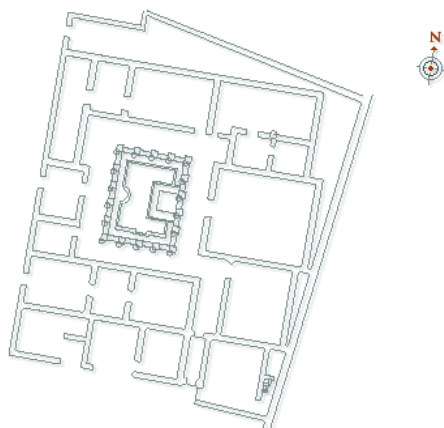


Fig. 1 Planta da Casa dos Esqueletos de Conimbriga

Actualmente, tal como a cidade em si, a Casa dos Esqueletos está em ruínas e praticamente destruída.



Fig. 2 Aspecto actual do peristilo¹ central e do jardim

3. A TECNOLOGIA

Um dos principais objectivos deste trabalho (a pedido do Museu Monográfico de Conimbriga), era a disponibilização da reconstrução virtual da Casa dos Esqueletos a um maior número de pessoas possível, nomeadamente através da Web. Como o VRML é uma linguagem que permite o desenvolvimento e publicação de ambientes virtuais na Web, surgiu como uma escolha natural. No entanto, esse não foi o único motivo da escolha desta linguagem. Este trabalho surge, também, como uma sucessão natural de um outro desenvolvido igualmente nas ruínas de Conimbriga: a reconstrução virtual do Fórum Flaviano de Conimbriga (www.forumflaviano.web.pt) [Gonçalves, 2003]. Além de ser intenção do Director do Museu Monográfico de Conimbriga ter mais uma maqueta virtual disponível na

Web, era intenção dos autores do projecto testar a viabilidade do método de trabalho utilizado no desenvolvimento do Fórum, e que obteve resultados surpreendentes. Essencialmente ao desenvolver outros modelos virtuais com características substancialmente diferentes das existentes no Fórum, nomeadamente ao nível da simetria das superfícies geométricas existentes. O Fórum Flaviano, apesar de ter dimensões substancialmente superiores, possuía uma arquitectura bastante simétrica em quase toda a sua extensão, algo que não acontecia na arquitectura da Casa dos Esqueletos. O desafio que se colocava neste projecto era: será que neste modelo da Casa dos Esqueletos, com muito poucas simetrias, a metodologia de trabalho usada no desenvolvimento do Fórum, obteria resultados idênticos?

3.1 Ferramenta de desenvolvimento

O VRML é uma linguagem com uma sintaxe própria que gera simples ficheiros de texto. Existem, no entanto, ferramentas de desenvolvimento VRML que geram o código VRML. A aplicação utilizada no desenvolvimento deste trabalho foi o *Cosmo Worlds 2.0* da *Silicon Graphics*.

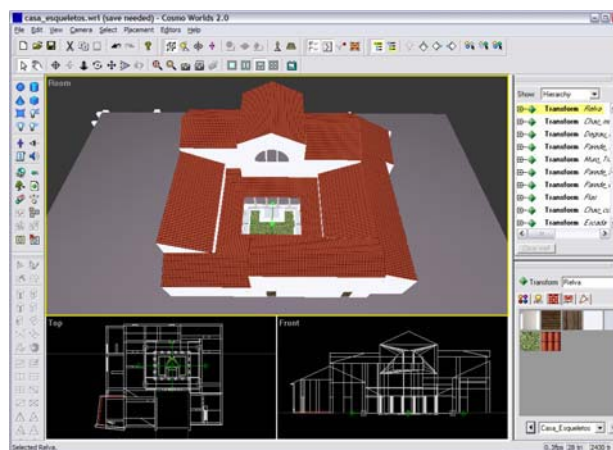


Fig. 3 Interface do *Cosmo Worlds 2.0*

4. OPTIMIZAÇÕES

Este é o ponto-chave no desenvolvimento deste trabalho, pois as optimizações efectuadas foram capitais para os resultados obtidos.

De seguida serão apresentados apenas algumas das optimizações mais relevantes implementadas no desenvolvimento da Casa dos Esqueletos.

4.1 Colisões

Uma vez que toda a movimentação e visualização do espaço tridimensional é feita em tempo real, os testes de detecção de colisões, também eles feitos em tempo real, são geralmente computacionalmente pesados. De modo a simplificar esse processo duas medidas foram implementadas, fazendo uso das propriedades do VRML:

¹ Espécie de corredor ornado de colunas

- Desactivação dos testes de colisão das superfícies com as quais o avatar não tem possibilidades de colidir, como é o caso dos telhados.
- Para facilitar e simplificar os testes de colisão foram definidas formas geométricas menos complexas, para serem usadas nos cálculos da detecção de colisões em substituição das formas originais, estas mais complexas e logo geradoras de cálculos mais pesados. Tal permite uma redução significativa dos testes de colisão que são efectuados entre o avatar e uma qualquer superfície sólida.

4.2 Instanciação (Clonagem)

O VRML possibilita duas formas de inserir, por mais que uma vez, um mesmo objecto na cena. Uma delas é o tradicional *copy/paste*, que cria uma cópia do objecto (na realidade é feita uma cópia do código VRML que gera o objecto). Outra forma de implementar um procedimento semelhante é através da instanciação. Esta técnica permite a reutilização do mesmo objecto, mantendo todas as suas propriedades geométricas, em diferentes partes do espaço virtual, por intermédio de transformações geométricas (translações, rotações e mudanças de escala). Essa propriedade é extremamente útil, pois evita a duplicação desnecessária de código, reutilizando as propriedades do(s) objecto(s) já definidas.

Todas as paredes da casa foram construídas com base em três blocos aos quais posteriormente se aplicaram translações, rotações e mudanças de escala, dando origem às paredes da casa. Todas as paredes em volta do jardim foram igualmente desenvolvidas com base num único objecto.

As colunas do jardim, são outro exemplo claro das vantagens da instanciação. Apenas uma coluna foi modelada, todas as restantes foram obtidas por esta metodologia. Tal permitiu ganhos significativos quer no desempenho quer na dimensão final do ficheiro, principalmente devido ao facto da representação geométrica desta coluna não ser nada trivial.



Fig. 4 Colunas do jardim obtidas por instanciação

A maioria dos telhados de duas águas foi igualmente reproduzida desta forma. Só não foi implementada na sua totalidade porque a mudança de escala aplicada degradava o aspecto das texturas expostas em mosaico.



Fig. 5 Perspectiva dos telhados

Muitas outras superfícies, em todo o espaço da casa, foram reproduzidas usando esta metodologia, o que comprova a viabilidade da mesma.

4.3 Billboard

Um *billboard* em VRML é uma entidade geométrica que acompanha o movimento do avatar, de modo a que se apresente sempre de frente para o avatar.

Analisando a funcionalidade de um *billboard* facilmente se conclui que está mais vocacionado para ser utilizado em objectos simétricos, cuja observação de diversos ângulos não tenha grandes variações.

Na Casa dos Esqueletos, o uso de *billboards* resume-se às colunas Toscanas que suportam o peristilo central. O seu apurado detalhe geométrico e o elevado número de colunas existente em toda a zona do peristilo, faziam com que fosse importante otimizar a sua representação. E pela sua simetria, revelou-se perfeito para a utilização desta técnica.

De modo a potenciar o realismo na representação da coluna, em vez de aplicar o *billboard* num plano bidimensional, como é mais comum, o mesmo foi aplicado a uma superfície semicilíndrica. Desta forma foi possível obter um efeito mais realista após aplicar a iluminação, uma vez que as superfícies curvas “espalham” a luz de uma forma diferente que as superfícies planas.

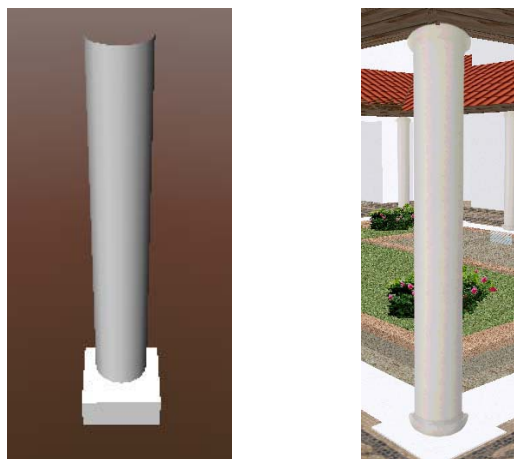


Fig. 6 O billboard na coluna Toscana

4.4 Publicação/Compressão

Um dos problemas dos cibernautas (no entanto, cada vez menos) é a necessidade de banda larga. Mesmo com as inovações tecnológicas ao nível do acesso à Internet, como o ADSL e o cabo, o tempo de espera por parte dos utilizadores dita o sucesso ou insucesso das aplicações disponíveis por este meio de difusão. O VRML não é imune a esta problemática e, como tal, a dimensão final dos ficheiros sempre foi uma preocupação constante dos criadores de ambientes virtuais que utilizam esta linguagem.[Silva, 2004]

Actualmente, grande parte dos *browsers* VRML disponibiliza a descompactação do formato gzip, em tempo real, permitindo assim a leitura e visualização imediata do ficheiro compactado sem qualquer dificuldade.

Neste caso em particular, o processo de compressão foi muito simples, pois a ferramenta de trabalho (Cosmo Worlds) possui um comando que o faz automaticamente e sem dificuldades de maior.

5. CARACTERÍSTICAS PRINCIPAIS

Para melhor se constatarem os resultados práticos da aplicação das técnicas anteriormente explanadas à construção da Casa dos Esqueletos, e não perdendo o foco relativamente ao ponto de referência que foi o Fórum Flaviano, serão apresentadas de seguida algumas das características mais relevantes de ambos os trabalhos.

5.1 Modelo realista

Ambos os modelos virtuais são representações realistas e à escala dos “edifícios” originais. O que permite ao seu visitante (virtual) uma perspectiva real de um habitante desse período, e como tal, ter uma real percepção das cores e formas da arquitectura do antigo Império Romano.

Além disso, todos os movimentos são feitos de modo a que o utilizador tenha uma percepção real do que lhe estaria a suceder, fisicamente, numa visita real, tal como

a noção de subir escadas ou uma queda de um ponto mais elevado.

5.2 Interactividade

Outra das principais características destes trabalhos é a interactividade. Os espaços virtuais foram criados de tal forma que o visitante (utilizador) tenha uma completa autonomia de movimentos quer pela área da Casa dos Esqueletos, quer pelo espaço do Fórum Flaviano. Isto significa que se pode deslocar para onde entender, seguindo o caminho que desejar, bastando para tal um simples dispositivo de interacção (rato e/ou teclado).

5.3 Visita Guiada

Foi também criada uma versão com uma visita guiada por todas as áreas mais importantes de ambos os modelos. Inclusivamente esta é a única forma de ter uma vista aérea da Casa dos Esqueletos, na medida em que na vertente interactiva apenas poderá “andar” pela casa.

5.4 Dimensão dos ficheiros

Este foi um dos principais resultados obtidos com este trabalho. Devido às optimizações efectuadas foi conseguida uma dimensão final dos ficheiros bastante reduzida. Por exemplo, o principal, e único, ficheiro VRML que gera toda o espaço virtual da Casa dos Esqueletos tem aproximadamente 17Kb e na sua totalidade, incluindo as texturas, não chega aos 300Kb – todo o Fórum Flaviano não ultrapassa os 150 Kb. É possível armazenar ambos os modelos virtuais numa pequena e obsoleta disquete, sem perda significativa do realismo.

5.5 Desempenho

Uma vez que todo o processamento de visualização (VRML) é feito em tempo real, na maioria das vezes, a capacidade de processamento, gráfica e não só, de um PC doméstico é insuficiente para a obtenção de um movimento fluido e natural durante a “visita” ao espaço virtual. No entanto, neste caso, tal não acontece devido às optimizações implementadas no desenvolvimento deste trabalho. O seu desempenho é plenamente satisfatório em qualquer computador doméstico.

5.6 Acessibilidade

Este costuma ser um dos problemas de muitos projectos de realidade virtual. Muitas das vezes os requisitos necessários em termos de hardware e/ou software para os visualizar comprometem seriamente o seu acesso a todos os interessados. Neste tipo de projectos, a única condição necessária para visualizar/visitar o espaço virtual é uma ligação à Internet e um *plug-in* VRML disponível gratuitamente na Web. Desta forma, e até pelos motivos já explicados anteriormente, esta recriação virtual de uma casa outrora existente numa cidade do antigo Império Romano em território Lusitano, está acessível a qualquer interessado sem grandes dificuldades.

A Casa dos Esqueletos (virtual) está disponível em: www.casadosesqueletos.web.pt.

6. CONCLUSÃO

A construção de modelos virtuais para disponibilização na Internet, é um modo válido de alargar a oferta que esta possibilita de uma forma mais rica e apelativa. No entanto tais ambientes não devem desmotivar o utilizador da sua consulta, devendo oferecer um equilíbrio entre no realismo que apresentam e as contrapartidas que exigem. Como tal, a aplicação de um método de trabalho metódico, que exija mais ao programador do que ao utilizador final, revela-se uma mais valia para proporcionar modelos apelativos.

A reconstrução da Casa dos Esqueletos obrigou a uma especial atenção na aplicação de técnicas que potenciam a acessibilidade por parte dos utilizadores finais, não lhes exigindo mais do que um computador pessoal normal e um acesso à Internet, de qualquer largura de banda. É claro que para este factor se tornar evidente é preciso ter em conta inúmeros aspectos de optimização, conhecê-los, interiorizá-los e recorrer a eles sempre que possível durante todo o processo de desenvolvimento [Silva, 2004]. O não conhecimento e consequente não utilização dos factores de optimização, resulta, muitas vezes, em modelos que não são representativos das qualidades do VRML. A metodologia que serve para modelos simétricos revelou-se válida para modelos que não apresentam essa mesma característica.

Porém, foi necessário recorrer à criatividade para aplicar de um modo mais alargado métodos como a instanciação – ao contrário do Fórum Flaviano, não foi clonada uma parte da estrutura do edifício, foram sim criados apenas três blocos a partir dos quais se recriaram todas as paredes da casa. Este facto originou um cuidado especial a ter na escolha da textura final, de modo a que não originasse malformações visíveis ao utilizador.

Para conseguir utilizar correctamente o VRML na construção de modelos virtuais, deve ser empregue um esforço na investigação e interiorização das optimizações. Embora o estudo destes factores consuma tempo, uma vez dominados permitem alcançar resultados surpreendentes com tempos de desenvolvimento idênticos aos de outras tecnologias [Silva, 2004].

Em conclusão, podemos dizer que a aplicação de uma metodologia de trabalho que respeite as especificações do VRML e que reúna todas as optimizações que permite, é certamente sinónimo de sucesso.

7. BIBLIOGRAFIA

[Gonçalves, 2003] A. Gonçalves e A. Mendes. *VRML... porque não?*, in Proc. of 12º Encontro Português de Computação Gráfica, Porto, Outubro (2003).

[Silva, 2004] F. Silva, D. Rodrigues e A. Gonçalves. *House of the Skeletons – A Virtual Way*, in Proc. of XXXII CAA – Computer Applications and Quantitative Methods to Archaeology Conference, Prato – Itália, April (2004).

Um Ambiente de Baixo Custo para Aplicações de Realidade Virtual e Aumentada

Paulo Correia
Moreira
Univ. de Aveiro
a19969@alunos.
det.ua.pt

Mário Cruz
Univ. de Aveiro
a23487@alunos.
det.ua.pt

Paulo Dias
DET / IEETA – Univ.
de Aveiro
paulo.dias@det.
ua.pt

Paulo Martins
Univ. de Aveiro
a21097@alunos.
det.ua.pt

Rui Gomes
Univ. de Aveiro
a22282@alunos.
det.ua.pt

Joaquim Madeira
DET / IEETA – Univ.
de Aveiro
jmadeira@det.
ua.pt

Manuel Bernardo
Cunha
DET / IEETA – Univ.
de Aveiro
mbc@det.ua.pt

Beatriz Sousa
Santos
DET / IEETA – Univ.
de Aveiro
bss@det.ua.pt

Sumário

Apesar da sua crescente importância, as aplicações dos ambientes de Realidade Virtual e/ou Aumentada têm sido, de algum modo, ainda limitadas quer pelo custo das tecnologias que lhes são habitualmente associadas, quer pela inexistência de técnicas de interacção apropriadas. Nesta comunicação são descritos os esforços desenvolvidos para a criação de um ambiente de custo reduzido que permite a investigação e o desenvolvimento de aplicações em Realidade Virtual e Aumentada.

A utilização de um “Head-Mounted Display” e de sensores de orientação e de aceleração, conjugados com “toolkits” gráficos de domínio público, permitiram o desenvolvimento da arquitectura de um primeiro protótipo e a rápida implementação de ambientes virtuais, bem como o teste de diferentes métodos de navegação. A posterior adição de uma câmara de vídeo passou a permitir, também, a criação de ambientes aumentados do tipo “video see-through” e com algum grau de interacção.

Palavras-Chave

Realidade Virtual, Realidade Aumentada, Navegação, Interacção

1. INTRODUÇÃO

O desenvolvimento de ambientes de Realidade Virtual ou Aumentada requer a utilização de componentes de *hardware* muito específicos, com destaque para os “*Head-Mounted Displays*” e sensores associados. A disponibilidade deste tipo de equipamento a um preço acessível veio diminuir a fasquia de entrada para a investigação e o desenvolvimento de aplicações nestas áreas. Ao conjugar este factor com a existência de bibliotecas e ferramentas de programação gráfica livremente disponíveis, que diminuem o esforço de desenvolvimento, permitindo a criação rápida de protótipos, torna-se possível conceber uma plataforma de custo reduzido, mas com todas as capacidades necessárias, para investigação e desenvolvimento nos domínios da Realidade Virtual e Aumentada.

Nesta comunicação é descrito o trabalho realizado¹ com o fim de, numa primeira fase, criar um protótipo de uma aplicação genérica para exploração de modelos tridimensionais em Realidade Virtual, utilizando um “*Head-Mounted Display*” e sensores de orientação e de aceleração. A adição de uma câmara de vídeo permitiu, numa segunda fase, o desenvolvimento de um protótipo de um ambiente de Realidade Aumentada do tipo “*video see-through*”, e o teste de técnicas de interacção baseadas no reconhecimento da posição de marcadores.

O trabalho realizado conduziu assim ao desenvolvimento de um sistema de baixo custo que permite quer a concepção rápida de cenários de utilização de ambientes de Realidade Virtual ou Aumentada, quer o teste de técnicas de navegação e interacção nesses cenários.

¹ No âmbito da disciplina de Projecto do 5º ano da Lic. em Engenharia de Computadores e Telemática, da Universidade de Aveiro.

2. REALIDADE VIRTUAL

A concepção de aplicações para Realidade Virtual é um processo complexo [vanDam00]. O objectivo de proporcionar ao utilizador uma experiência sensorial imersiva num ambiente artificial depende dos componentes de *hardware* e *software* que lhe estão associados, pois é por seu intermédio que se influenciam os sentidos, e se consegue interagir com o ambiente e os objectos nele definidos. Para criar esta sensação de imersão e presença são habitualmente usados dispositivos de visualização como os “*Head-Mounted Displays*” (HMD), sensores de diversa natureza e função, e dispositivos de interacção como as “*cybergloves*”, entre outros [SallesDias04].

2.1 Equipamento

Na circunstância do protótipo desenvolvido, foi usado um HMD *i-glasses SVGA Pro* [ISense] capaz de uma resolução de 800x600 *pixels*, um sensor de orientação (“*tracker*”) *InterTrax 2* da *InterSense* [ISense] com três graus de liberdade (“*yaw*”, “*pitch*” e “*roll*”) e um sensor de aceleração estática e dinâmica em três eixos ortogonais, construído a partir de um sensor LIS3L02AQ da *STMicroelectronics* [STMicroelectronics] e de um micro-controlador PIC18F458 da *Microchip* [Microchip].

A comunicação destes equipamentos com o computador realiza-se com fios, sendo no caso do HMD por intermédio de uma interface VGA, do “*tracker*” pela porta USB e do acelerómetro pela porta série RS-232. As características do PC utilizado são um processador *Pentium 4* a 3GHz, 1GB de memória RAM, e uma placa gráfica *GeForce FX 5950 Ultra* com 256MB de memória.



Fig. 1: HMD com sensor de orientação e micro câmara

2.2 Software

Para o desenvolvimento do protótipo do ambiente de Realidade Virtual utilizou-se uma arquitectura modular baseada no modelo de programação orientado a objectos, permitindo gerir de uma melhor forma a complexidade existente, oriunda essencialmente do fornecimento e controlo de dados de diversas fontes.

Para o processamento e a representação dos modelos tridimensionais utilizou-se a “*Visualization Toolkit*”

[VTK]. Esta ferramenta revelou ser útil na implementação de aplicações desta natureza por oferecer diferentes funcionalidades para a manipulação e visualização de dados tridimensionais, importação de modelos e interacção com dispositivos.

No tocante à comunicação com os dispositivos, esta foi, com o “*tracker*”, efectuada recorrendo ao “*driver*” fornecido pelo fabricante, com o acelerómetro, recorrendo a um protocolo simples com a porta série, e relativamente ao HMD, não foi necessário nenhum método específico, sendo as imagens enviadas para o mesmo através da interface VGA da placa gráfica.

2.3 Ambientes

Como já foi mencionado, e como a designação Realidade Virtual alude, os ambientes em aplicações deste género são artificiais. Para que possam ser gerados e exibidos é necessário descrever os modelos tridimensionais que compõem as cenas nas quais o utilizador se movimenta num formato específico, conhecido pela aplicação.

No caso do protótipo desenvolvido foi escolhido o formato VRML 2.0, por ser um formato padrão para envio de modelos pela *Web*. Muito embora a capacidade de interpretar ficheiros VRML 2.0 exista na biblioteca VTK, esta não permite a leitura das texturas dos modelos. Assim, foi implementada de raiz uma biblioteca que permite ler não só a informação geométrica mas também as texturas dos modelos tridimensionais dos ficheiros VRML 2.0, convertendo esta informação no formato nativo da VTK.

2.4 Navegação

A navegação é um inevitável requisito num ambiente de Realidade Virtual, sendo uma componente fundamental para a sensação de presença e imersão. De modo a que esta imersão seja conseguida, é necessária a existência de componentes sensoriais no ambiente, através de dispositivos de entrada e saída.

No caso do sistema desenvolvido, a navegação foi considerada como sendo definida por informação de orientação e de posicionamento, fornecida por dispositivos distintos. Para a orientação, o dispositivo é o sensor de orientação (“*tracker*”) colocado sobre o HMD e, para o posicionamento, o dispositivo que auxilia o utilizador a deslocar-se pelo mundo virtual é o acelerómetro.

2.4.1 Orientação

A orientação é conseguida por intermédio do “*tracker*”, que permite uma correlação entre os movimentos da cabeça de o utilizador e a representação da cena pretendida, permitindo que o utilizador consiga olhar à sua volta com as imagens correspondentes a serem reproduzidas no HMD. Esta correlação entre o real e o virtual é conseguida graças aos três graus de liberdade (“*yaw*”, “*pitch*” e “*roll*”) de que o *tracker* é munido, possibilitando um levantamento real de quaisquer possíveis movimentos da cabeça do utilizador.

2.4.2 Posicionamento

Por forma a tornar a experiência intuitiva, foram concebidos quatro modos de navegação que permitem um comportamento o mais aproximado possível do real, tendo em conta os dispositivos utilizados.

O primeiro método caracteriza-se por o utilizador se deslocar apenas sobre um plano, e só em direcções paralelas aos eixos coordenados que o definem, independentemente da orientação do HMD. O segundo método permite que o utilizador se desloque na direcção (e segundo o sentido) de orientação do HMD, i.e., segundo a direcção de projecção. O terceiro método, tem as mesmas características do anterior, mas não contempla a componente da altura do vector que define a direcção de projecção.

Por último, foi concebido um método que agrega os modos de navegação acima. Assim, a direcção e sentido do movimento é definida pela orientação que o utilizador tem quando se encontra imóvel. Quando em movimento (sobre um plano), o utilizador poderá olhar à sua volta sem que a sua direcção e sentido de movimento sejam alterados.

Como já foi mencionado, o acelerómetro é o dispositivo utilizado para controlar o deslocamento do utilizador no mundo virtual. Para que se lhe pudesse dar uma utilização adequada, teve que ser concebida uma técnica que tornasse a navegação intuitiva, não propensa a cansaço prematuro nem a possíveis lesões provenientes do uso excessivo. Esta baseia-se fundamentalmente no utilizador ter o sensor numa das mãos e, simplesmente, movimentar o braço para a frente, para trás ou para os lados, consoante se queira deslocar no mundo virtual em frente, para trás, ou para os lados, respectivamente. A velocidade de deslocamento é definida pela altura (inclinação) a que o sensor estiver da posição de repouso do braço (esticado e em baixo).



Fig. 2: Sensor de aceleração

2.5 Resultados

Conjugando os equipamentos, *software* e técnicas já discutidos foi implementado um protótipo de uma aplicação de navegação em ambientes virtuais. O protótipo permite experimentar diferentes mundos virtuais, bastando ler o modelo pretendido. A implementação foi pensada de forma a ser facilmente extensível, o que possibilita a criação de diferentes aplicações a partir desta base. Estas podem implementar métodos alternativos de interacção e navegação, ou expandir as capacidades do protótipo no tocante ao sistema físico e às capacidades multimédia.

Nas figuras 3 e 4 é possível observar imagens da aplicação a ser utilizada em diferentes cenários: um modelo do interior da Biblioteca da Universidade de Aveiro, e um modelo da Sala *Delacroix*, no *Palais Bourbon*, obtido a partir de dados laser [Dias2003].



Fig. 3: Vista interior do modelo da Biblioteca da UA

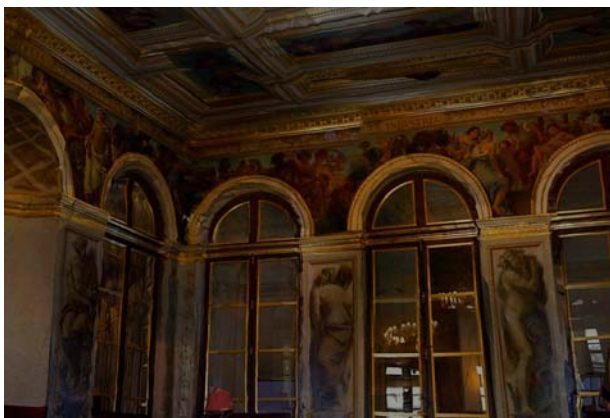


Fig. 4: Vista interior do modelo da Sala Delacroix

3. REALIDADE AUMENTADA

Partindo dos dispositivos de base e do protótipo de ambiente de Realidade Virtual já descritos, foi possível desenvolver um protótipo de uma aplicação de Realidade Aumentada com o simples acréscimo de uma câmara de vídeo. A utilização de um HMD convencional com uma câmara acoplada prefigura um sistema de Realidade Aumentada do tipo “*video see-through*”.

3.1 Equipamento

A câmara utilizada é uma micro-câmara sem fios analógica, com um sensor CMOS de aproximadamente 330 linhas, a funcionar na banda dos 1.2GHz, cujo receptor se encontra ligado a um conversor vídeo analógico-digital com saída USB 2.0. Este conversor permite utilizar qualquer câmara analógica como uma comum *webcam* USB, aumentando a flexibilidade da plataforma. As dimensões e consumo da câmara permitem que seja montada directamente no HMD, numa posição ao nível dos olhos do utilizador, e alimentada por uma pilha comum de 9 volt.

3.2 Software

Para o desenvolvimento do protótipo de Realidade Aumentada foi utilizado o *ARToolkit* [ARToolkit], um “*toolkit*” gráfico que incorpora capacidades de detecção da orientação da câmara e de “*rendering*” de objectos virtuais directamente na imagem capturada, posicionados de acordo com a orientação detectada. A orientação da câmara é calculada não relativamente ao espaço circundante, mas a marcadores pré-definidos colocados nesse espaço. Os marcadores são quadrados delineados a traço grosso negro, com um símbolo no seu interior.

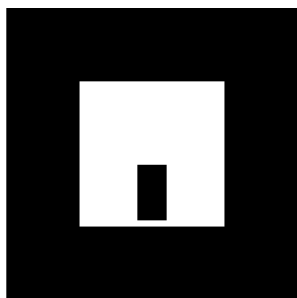


Fig. 5: Exemplo de marcador

A detecção dos marcadores começa pela conversão da imagem capturada num registo a preto e branco, por forma a maximizar o contraste entre o traço negro e o fundo branco. Este tipo de processamento permite fazer sobressair os marcadores em ambientes complexos e coloridos. A alteração do limiar, utilizado para a conversão das cores da imagem em preto ou branco, permite ainda adaptar o processo de detecção às características de iluminação e contraste das imagens obtidas pela câmara.

A deformação devida à perspectiva, a rotação do quadrado e a posição do símbolo no seu interior permitem calcular a orientação da câmara relativamente ao marcador. Atendendo a que a dimensão deste é conhecida, a determinação da distância também é possível.

Na posse destes dados, orientação e distância do observador relativamente aos marcadores, é possível desenhar objectos virtuais na cena, alinhados com estes. O “*rendering*” dos objectos virtuais sobre a imagem é feito utilizando as bibliotecas *OpenGL* [OpenGL] e *GLUT* [Glut].

3.3 Ambientes

Por definição, o ambiente em aplicações de Realidade Aumentada é o ambiente real que circunda o utilizador, o que no caso corresponde às imagens capturadas pela câmara. Os dados aumentados, sobrepostos no ambiente real são tipicamente objectos tridimensionais, mas podem no entanto ser apenas texto ou uma imagem num plano. O “*toolkit*” utilizado incorpora extensões que permitem ler objectos especificados em ficheiros VRML 2.0, tornando-o assim bastante flexível relativamente ao tipo

de conteúdos que podem ser adicionados a uma cena, permitindo incluir elementos com textura e animação.

3.4 Interacção

Os marcadores, utilizados no ambiente aumentado para determinar a posição do utilizador relativamente ao ambiente que o rodeia, podem também ser usados como instrumentos de interacção. O facto de ser possível segurar e manipular um marcador, dadas as suas características reais e tangíveis, junto com a possibilidade de determinar a posição da câmara no espaço, relativamente ao marcador, permite a interacção directa do utilizador com objectos tridimensionais virtuais, de forma orgânica e intuitiva. Esta interacção é especialmente útil na análise exterior de objectos, por forma a determinar pormenores da forma e volume dos mesmos, sendo uma ferramenta interessante em ambientes de criação de protótipos de objectos ou criação de maquetas à escala.

Como dispositivos adicionais de interacção, e dado o protótipo de Realidade Aumentada ser construído sobre o equipamento do ambiente de Realidade Virtual, é possível utilizar os sensores de aceleração e orientação, além dos tradicionais rato e teclado. Na implementação realizada, apenas o sensor de aceleração é utilizado, com o fim de assinalar a entrada e saída do modo de Realidade Virtual, e para navegação neste.

3.5 Resultados

Foi desenvolvido um protótipo de aplicação baseado num ambiente aumentado que contém um conjunto de itens virtuais, passíveis de serem seleccionados e manipulados pelo utilizador, que pode entrar num ambiente puramente virtual e navegar numa versão imersiva do item a ser observado. A componente de Realidade Virtual da aplicação é a já descrita na secção 2, pelo que a implementação se limitou ao desenvolvimento da componente de Realidade Aumentada e da sua interface com a componente de Realidade Virtual.

Como prova de conceito, foi criado um ambiente de teste, cujos elementos aumentados são modelos reduzidos de alguns dos ambientes virtuais disponíveis. Na figura 6 é possível observar o ambiente de Realidade Aumentada em funcionamento, com alguns destes elementos visíveis.

A selecção dos elementos é efectuada utilizando um “*avatar*”, associado ao marcador que o utilizador segura na mão, acoplado ao sensor de aceleração para maior comodidade. Um exemplo de selecção e manipulação de um dos modelos é observável na figura 7.

Finalmente, o utilizador pode imergir no ambiente virtual associado ao objecto seleccionado, alterando a posição do sensor de aceleração, rodando-o de 180°.

Os testes com a aplicação desenvolvida permitiram provar a validade do conceito, mas também fizeram emergir alguns problemas no ambiente de Realidade Aumentada. O mais pertinente prende-se com falhas na detecção dos marcadores, associadas às condições de luz e contraste do ambiente em que se está a operar.

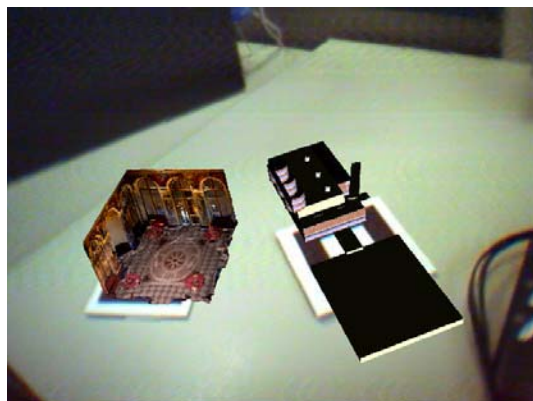


Fig. 6: Exemplo de elementos aumentados



Fig. 7: Exemplo de elemento seleccionado

O ambiente deve ter muita luminosidade, uniformemente distribuída, por forma a proporcionar a melhor taxa de detecção. Deficiências a este nível podem levar ainda a detecções pouco fiáveis, em que objectos existentes no campo visual do utilizador são confundidos com marcadores, gerando aparições esporádicas de modelos em locais e posições inesperados.

A iluminação dos modelos virtuais utilizados no ambiente de Realidade Aumentada também deve ser considerada. Estes são, por omissão, iluminados apenas por de um dos lados, fazendo com que o utilizador, ao manipular o marcador para visualizar o modelo de múltiplos ângulos, se depare com áreas totalmente escuras. Convém assim que os modelos incluam diversas fontes de luz, que possibilitem uma iluminação, se não uniforme, pelo menos suficiente em todos os modelos.

4. CONCLUSÃO

Com base na utilização de dispositivos de custo reduzido, foi desenvolvido um conjunto de funcionalidades que permite quer a concepção rápida de cenários de utilização de ambientes de Realidade Virtual ou de Realidade Aumentada, quer o teste de outras futuras técnicas de navegação e interacção nesses tipos cenários. No entanto, o trabalho efectuado constitui apenas o início da exploração das potencialidades das tecnologias da Realidade Virtual e Aumentada, devendo ser encarado como trabalho em e para desenvolvimento.

A utilização do sensor de aceleração permitiu alargar o trabalho ao estudo e implementação de métodos de interacção e navegação alternativos. Ficando assim demonstrada a possibilidade de construir uma plataforma leve, flexível e de custo reduzido para investigação na área, é agora possível evoluir sobre a mesma, acrescentando novas características no domínio da interacção e da imersividade, explorando áreas como a detecção posicional do utilizador, imersão sonora, e ambientes verdadeiramente aumentados.

5. AGRADECIMENTOS

Ao Tiago Vieira e João Santos, pela concepção e desenvolvimento do modelo em VRML 2.0 da Biblioteca da Universidade de Aveiro², utilizado para teste e demonstração neste trabalho.

6. REFERÊNCIAS

- [ARToolkit] ARToolkit, July 2005, <http://www.hitl.washington.edu/artoolkit/>
- [Dias03] Paulo Dias, "Three Dimensional Reconstruction Of Real World Scenes Using Laser and Intensity Data", *PhD Thesis*, University of Aveiro, 26 September 2003
- [Glut] GLUT, July 2005, <http://www.opengl.org/resources/libraries/glut.html>
- [ISense] InterSense, July 2005, <http://www.isense.com>
- [Microchip] Microchip Technology Inc., July 2005, <http://www.microchip.com>
- [OpenGL] OpenGL, July 2005, <http://www.opengl.org/>
- [SallesDias04] José Miguel Salles Dias, Rafael Bastos, Pedro Santos, Luis Monteiro, Joaquim Canhoto, "The Arena: An Indoor Mixed Reality Space", *Actas de Interacção 2004 - 1ª Conferência Nacional em Interacção Pessoa Máquina*, pp. 109-119, Lisbon, 12-14 July 2004
- [STMicroelectronics] STMicroelectronics, July 2005, <http://www.st.com>
- [vanDam00] Andries van Dam, Andrew S. Forsberg, David H. Laidlaw, Joseph J. LaViola, Jr., and Rosemary M. Simpson, "Immersive VR for Scientific Visualization: A Progress Report", *IEEE Computer Graphics and Applications*, V. 20, No. 6, Nov/Dec. 2000, pp. 26-52
- [VTK] Visualization Toolkit, July 2005, <http://public.kitware.com/VTK/>

² Este modelo foi desenvolvido, em 2003/2004, no âmbito da disciplina de Interfaces Humano-Computador, do 2º semestre do 3º ano da Lic. em Engenharia de Computadores e Telemática, da Universidade de Aveiro.

Using Essential Reality's P5 glove as a low cost tracking device for Virtual Reality applications

Bruno Miguel Oliveira^{a,b}

Pedro Miguel Ângelo^a

^aEscola de Engenharia
Universidade do Minho
Campus de Azurém, 4800-058 Guimarães
{bruno.oliveira, pangelo}@dcc.online.pt

^bCentro de Computação Gráfica
Rua Teixeira de Pascoais, 596, 4800-073 Guimarães
Bruno.oliveira@ccg.pt

Abstract

Nowadays we find on the market some low cost tracking devices that in some cases appear to provide the same level of functionality if compared with other more expensive but alike devices. The P5 glove is one of such low cost devices. It provides positional tracking with six degrees of freedom and finger bend information.

In the paper we give an insight into the project developed to explore the possibilities of using the P5 glove in Virtual Reality applications. For this purpose we developed an application using open source tools and libraries which are considered to be standard in Virtual Reality application development, namely the VR Juggler framework, the OpenSG scene graph and the Linux operating system.

The implementation of a test-bed application is presented, that allows the use of the P5 in Virtual Reality environments.

Finally we draw out some recommendations for the applications for this device and present future work lines which might help to better understand its possibilities and improve its shortcomings.

Keywords

Low-Cost Tracking, Open-Source, Virtual Reality, VR Juggler.

1. INTRODUCTION

The P5 glove [Figure 1] is a low cost tracking device manufactured by Essential Reality LLC. It integrates an optical position and orientation sensor for tracking the user's hand in 3D space. It also provides bend sensors for the fingers and three programmable digital buttons.



Figure 1. P5 glove tracking system

The tracking of the user's hand is performed by an optical sensor unit placed in front of the user, which tracks the relative position of seven LEDs placed on the glove's surface. This type of tracking demands that the glove must always be positioned in front of the sensor unit

(according to manufacturer's documentation at least three of those LEDs must be in sight). Both the finger bend angle data and the programmable button data are captured by specific sensors embedded in the glove and transmitted to the main sensor unit through a cable.

The main sensor unit is connected to the host computer by a standard USB connection.

The purpose of this project was to explore the possibility of using this tracking device in Virtual Reality applications, running on the Linux operating system, by developing a test-bed application, which explores VR Juggler and OpenSG, both standard open-source APIs for developing Virtual Reality systems.

VR Juggler is a high level framework for Virtual Reality applications [Bierbaum02]. It provides an extremely flexible API, especially for input/output device and scene graph handling.

2. IMPLEMENTATION

A virtual reality application, especially one that relies on external libraries, is formed by several layers of software. Each layer is responsible for a very specific function and needs nothing but the lower layer to work. To meet the standards, we have developed the application in a bottom-up strategy, thus building each new layer on top of the previous one.

As one can observe in Figure 2, the lower block provides communication with the hardware, namely the P5 glove. On top of the P5 glove we have the Gadgeteer (note: gadgeteer is VR Juggler's input device handling API) that at its core requires a wrapper for the device driver. When the Gadgeteer subsystem is ready, VR Juggler can collect tracking data from the hardware. At this point we have fulfilled the minimum level of requirements to build an application using the P5 glove as an input device.

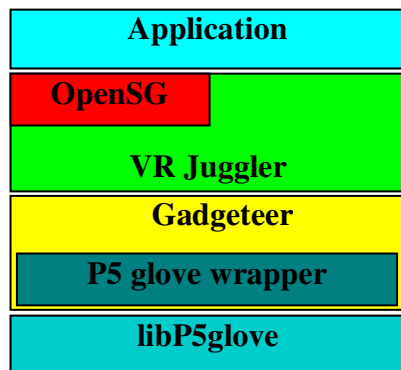


Figure 2. Overview of system architecture

As a proof of concept or test-bed we have implemented an application on top of VRJuggler using the P5 glove as input.

2.1 Low level device driver

As suggested in Gadgeteer's documentation, we first tried to create a low level device driver that would allow opening and closing the device, as well as sample acquisition and runtime configuration. To achieve this we had three possibilities: either develop our own device driver, or use the manufacturer's closed source drivers, or instead use libp5glove, a portable open-source device driver actively maintained by the P5 glove's user community. Given the short time span available for this project we decided not to write our own device driver.

Of the two remaining possibilities, we decided on using libp5glove. It provided us with a well designed open-source library that allowed us to peek into the lower level details of handling the device, should something go wrong. Further, our early tests showed us that libp5glove was more stable when accessing the device. The only caveat we found was that libp5glove would only allow one application to access the device at a time, something that was not troublesome for our purpose.

2.2 VR Juggler device proxy

The next step was to implement a Gadgeteer wrapper for the low level device driver. As we already mentioned, Gadgeteer is VR Juggler's device handling API. It provides a high level abstraction to the different types of controllers a device has, by using an elaborate device proxy system. This system allows Virtual Reality applications to be written taking into consideration only the kind of controllers needed, and not the controllers present on a specific device.

Even though VR Juggler comes bundled with Gadgeteer wrappers for a large number of tracking devices, there was none available for the P5. We had to develop a new

Gadgeteer wrapper from scratch in order to use the P5 with VR Juggler.

Creating a Gadgeteer wrapper is a rather straightforward process. After deciding which type of virtual controllers to support, we need to take over and implement a small set of interfaces. For our purpose, we decided to map the P5 glove onto a set of controllers that we felt best described the hardware:

- One PositionDevice controller that for each sample returns to the application a 4x4 matrix with the absolute position and orientation of the user's hand.
- Five AnalogDevice controllers that for each sample return a floating point number ranging from 0.0 to 1.0 that represents the bend angle of each individual finger.
- Three DigitalDevice controllers representing the glove's digital programmable buttons.

Having finished this process, it was necessary to configure the new Gadgeteer device in order for it to be used in applications.

To achieve this, we had to write two XML configuration files. The first one, called a device definition file, informs Gadgeteer about the necessary parameters to configure the device and about its position in VR Juggler's device hierarchy. The second file, called a device configuration file, differs from application to application, and maps the controls provided by the wrapper with the ones expected by the application. It defines the values for the device configuration parameters and sets the path to the device's low level driver.

2.3 OpenSG application

As stated before, OpenSG is a high level scene graph, developed in C++. The purpose of OpenSG is to build three dimensional worlds in a way very similar to VRML. However, the world is described in an object-oriented manner, as opposed to VRML's text based description [Reiners00].

OpenSG was designed only with the management and rendering of three-dimensional worlds in mind. It relies completely on external applications for device input and output. This is where VR Juggler is necessary, providing input and output device handling to OpenSG.

The greatest difficulty we found at this point was with the filtering of the sampled positional data coming from the glove. As with most tracking systems in use, the sampled positional data coming from the glove carries considerable noise, resulting in jittery behavior. This aspect is of particular importance due to the optical nature of the sensor unit. Filtering out this noise can be done at one of two levels: either at the low level device driver where the samples are collected, or at the Gadgeteer wrapper, where the samples are transformed to be used by the application. We chose the latter given that it was the approach suggested by VR Juggler's documentation, which specifies that device driver data should be sampled as raw as possible.

In our application, we decided to adapt a simple estimator provided by the manufacturer of the hardware. It consists of a simple weighted average of some predefined amount

of previous samples.

3. CONCLUSIONS

Our application shows a virtual hand in 3D space that responds realistically to the user's hand movements. This simple application allowed us to test both the developed software as well as the hardware.

Our tests have pointed out some limitations in the hardware, especially some very evident jittering and a short functional range of approximately 0.5 cubic meters centered in front of the optical sensor unit. The jittering was partially solved by the simple filter we implemented.

We can finally conclude that this device presents some promising features for developing Virtual Reality applications.

First and foremost, the P5 is a very low cost device, providing finger bend information and accurate tracking of the user's hand with six degrees of freedom inside a volume of approximately half a cubic meter.

Tracking devices for Virtual Reality applications are usually very expensive. Having a low cost device available can be rather useful for the development of Virtual Reality solutions or quick testing prototypes. If the system is implemented using VR Juggler, then our results are directly applicable. Such an application can be deployed on a more expensive tracking system with only a change of configuration.

Also, combining VR Juggler and this device can significantly reduce the entry cost for newcomers to the Virtual Reality industry.

Education in the Virtual Reality field is a very complex matter. Giving students access to very expensive equipment is a not feasible proposition for any institution. Usually students use simulation software to emulate tracking device input to their applications. Having a low-cost device that easily integrates together with standard libraries can provide a much richer experience to the students. We came to this conclusion out of our personal experience with this project.

The low cost of this device and its small form factor can help to bring Virtual Reality applications to people's homes.

The limited operational range of the device makes it especially suitable for new desktop applications. One can easily envisage applications that take advantage of a setup consisting of a user wearing the P5 glove sitting in front of a normal desktop workstation or carrying it in a portable or wearable computer.

The most evident applications for the P5 glove would be in electronic entertainment. Home users are used to the metaphors found in electronic entertainment applications and it would be rather easy for them to replace the keyboard and mouse by the glove.

Apart from that, one can imagine a new type of interaction between the user and the computer. The operating system could provide a three dimensional desktop in which users would move the windows with their own hands.

Moreover, some efforts are already being made to create

virtual reality musical instruments using the glove for performing [Burtner04].

Also, new interaction paradigms can be explored using a glove. For instance, gesture recognition as described in [Bimber99] can give applications a more intuitive and comfortable feel.

Modeling could also gain with the P5 glove, since the process of modeling three dimensional objects is analogous to the process of modeling clay. This sort of intuitive workflow is something that is clumsy to achieve with a mouse. The nowadays common use of the mouse for this task should be considered a rough approach that might be refined by the use of these input devices.

Finally, there is also some interesting future work that can be developed from our results, which we could not pursue due to the lack of available time:

- Further testing.
Our testing was not very extensive. It would be desirable to realize some further testing in order to pinpoint additional limitations of the hardware. Some variables like latency, resolution, and the effect of ambient luminosity are still not documented.
- Calibration.
The hardware provides calibration facilities, and the chosen device driver has functions to access them. However, these functions are not very well documented.
- Filtering.
The current filter used by the application is a simple, albeit effective, one. In future, other kinds of filters should be tested, among them, the well known Kalman filter [Grewal01], in order to improve the quality of the interactivity.

4. ACKNOWLEDGEMENTS

We would like to thank Leonel Valbom for all the support and help provided and also for lending us the equipment for this project.

5. REFERENCES

- [Bierbaum01] Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A. & Cruz-Neira, C. VR Juggler: A Virtual Platform for Virtual Reality Application Development, *Proceedings of IEEE Virtual Reality*, Yokohama, Japan, March 2001.
- [Reiners00] Reiners, D., Rettig, A., et. al. OpenSG - An OpenSource Scenograph, May 2000.
<<http://www.opensg.org/downloads/OpenSG-1.0-Design.html>>
- [Burtner04] Burtner, M., A Theory of Modulated Objects for New Shamanic Controller Design, *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04)*, Hamamatsu, Japan
- [Bimber99] Bimber, O., Continuous 6DOF Gesture Recognition: A Fuzzy Logic Approach *Proceedings of WSCG'99, vol. 1, pp. 24-30*, Plzen, Czech Republic, February 1999
- [Grewal01] Grewal, S. M., Andrews, P. A., Kalman Filtering: Theory and Practice, Wiley-Interscience, 2001

Sistema de Orientação para o Interior de Edifícios - Solução Visual de Localização

João Sena Marcos
joaosenamarcos@sapo.pt

José Miguel Salles Dias
Miguel.dias@iscte.pt

António Calado Lopes
Antonio.calado@iscte.pt

ADETTI, Edifício ISCTE, Av. Das Forças Armadas, 1600-082 Lisbon, Portugal,
<http://www.adetti.iscte.pt>

Abstract

Neste artigo apresentamos um sistema para computação móvel em cenários interiores, baseado em PDA, que consegue resolver com técnicas de visão por computador, a localização do utilizador e possibilita a sua orientação dentro de um edifício através de marcas dispostas em locais estratégicos, previamente identificados. As marcas, capturadas com a câmara de vídeo integrada no PDA, são conhecidas pelo sistema, que tendo identificado o local, faz aparecer no ecrã uma seta virtual com a direcção do caminho que se deve seguir, para chegar da forma mais rápida ao destino inicialmente seleccionado. É utilizada a biblioteca de Realidade Aumentada 'ARToolkit' adaptada à plataforma PocketPC, para as operações de reconhecimento de marcas e cálculo da pose da câmara real relativamente à marca. Para o desenho 3D é usada a biblioteca open-source 'Klimt'. O sistema é apresentado nos seus componentes principais, com ênfase no algoritmo de descoberta do caminho mais curto de Dijkstra e na disponibilização de Realidade Aumentada em ambiente de computação móvel com escasos recursos computacionais e com reutilização de marcas de visuais.

Keywords

Realidade Aumentada, Computação Móvel, PDA, Localização Visual, Marcas Visuais, Dijkstra, AR Toolkit

1. INTRODUÇÃO

Neste trabalho, propusemo-nos desenvolver um sistema de localização e orientação, utilizável no interior de edifícios em geral e, no ISCTE, em particular, que funcionasse num dispositivo de computação móvel simples, do tipo PDA (*Personal Digital Assistant*). Não podendo recorrer ao GPS (*Global Positioning System*) por não haver linha de vista para os satélites, nem a *pseudolites* (que correspondem à versão Terra dos satélites GPS, os quais foram descartados neste trabalho devido ao seu custo proibitivo), apresentavam-se-nos duas soluções alternativas: a colocação de uma multiplicidade de sensores no edifício, que comunicassem com o sistema via emissores/receptores infravermelhos, *bluetooth*, rede WI/FI, etc., ou a colocação na parede de marcas visuais conhecidas do sistema, que teriam que ser filmadas e adquiridas pelo dispositivo móvel do sistema, para o respectivo reconhecimento do local. Escolhemos a segunda solução, essencialmente por ser menos onerosa e, ainda, por permitir mais flexibilidade para o fim em vista. De facto, a tecnologia de seguimento da pose (posição e orientação) de um utilizador em movimento, por tecnologia de infra-

vermelhos, está limitada a uso em linha de vista entre dispositivos activos, o que implicaria a disponibilização de um número elevado dos referidos dispositivos de seguimento em áreas consideráveis do edifício (com o consequente aumento dos custos envolvidos). Por outro lado, as outras tecnologias de seguimento referidas apresentam os seus inconvenientes próprios: *bluetooth* está igualmente limitado geograficamente a pequenas áreas e, as redes WI/FI, ainda não se caracterizam pela ubiquidade da disponibilização do respectivo serviço de comunicação nos edifícios em geral e no ISCTE, em particular, onde os testes técnicos do sistema têm vindo a ser executados.

As marcas visuais usadas são identificadas pelo nível de Realidade Aumentada do nosso sistema, executada em modo local no dispositivo PDA e fazem parte de uma solução GPL de Realidade Aumentada, o ARToolkit, desenvolvida por Mark Billinghurst e Hirokazu Kato [Billinghurst03] [Kato03].

Para o desenvolvimento deste sistema, foram de grande utilidade os trabalhos publicados de duas equipas lideradas por Dieter Schmalstieg, da *Vienna University of Technology, Áustria* [Kalkusch02, Wagner03], o último

dos quais diz respeito à disponibilização do *ARToolkit* para ambientes de computação móvel em PDA (em modo local).

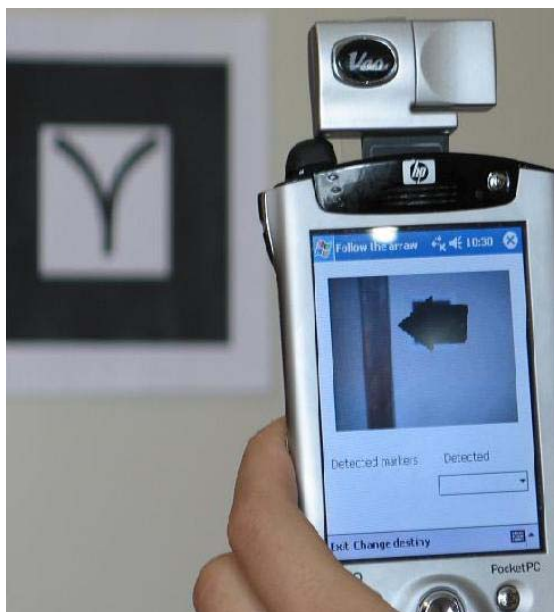


Figura 1. O Sistema reconhece a marca colocada num nó do percurso no interior do edifício e, conhecendo o caminho mais curto para chegar ao destino, sobrepõe uma seta na marca, a qual indica que o caminho devido, segue para a esquerda.

O artigo está organizado da seguinte forma: na secção 2, descrevemos a arquitectura modular e o funcionamento global do sistema. Na secção 3, abordamos o esquema de distribuição de marcas pelo interior do edifício, bem como a problemática da reutilização das marcas em ambientes computacionais de recursos limitados. A secção 5, foca a interface de utilização do sistema, bem como o equipamento utilizado. Finalmente, na secção 5, extraímos algumas conclusões e apontamos algumas direcções para trabalho futuro.

2. FUNCIONAMENTO GLOBAL DO SISTEMA

O nosso sistema desenvolveu-se a partir de um requisito fundamental: a necessidade de indicar ao utilizador o caminho a seguir para um dado destino, no interior de um edifício. A satisfação deste requisito, exige, em primeiro lugar, o conhecimento da localização do mesmo. Depois, terá que descobrir qual o caminho mais curto desde o local actual até ao destino, com base em informação geométrica e topológica do respectivo edifício (disponível a partir da informação em planta) e, finalmente, indicar a sua direcção.

No esquema que se mostra na Figura 2, podemos observar os diferentes módulos do sistema e o respectivo fluxo de informação.

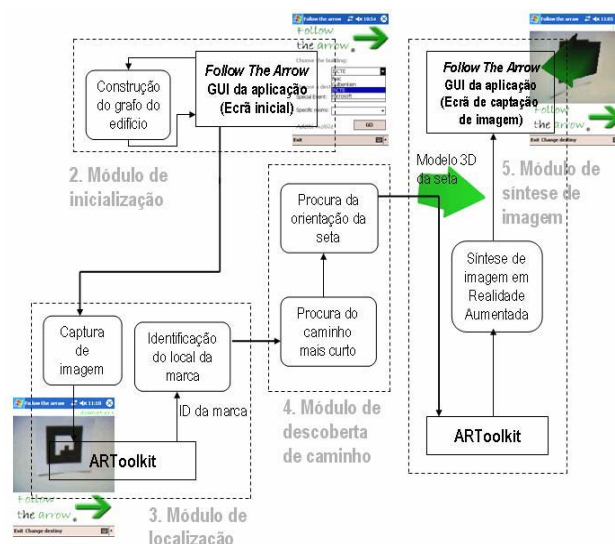


Figura 2. Módulos do sistema e fluxo de operações

2.1 MÓDULO DE INICIALIZAÇÃO

O módulo de inicialização do sistema, constrói um grafo a partir da informação geométrica e topológica do edifício seleccionada pelo utilizador, a qual está guardada num ficheiro de entrada 'adjacentVertices.txt'.

Uma forma prática que encontramos para representar a geometria e topologia da planta de edifícios foi a construção de grafos, em que os nós do grafo são os pontos de interesse nas planta e o custo entre nós, a distância estimada ou arbitrada entre pontos adjacentes. Cada grafo corresponde a uma ou mais plantas do edifício e é construído a partir de um ficheiro de pontos de interesse, onde estão registadas as distâncias e as relações espaciais entre pontos adjacentes.

Os pontos de interesse tanto podem ser divisões do edifício como locais onde é necessário escolher uma direcção a seguir, em virtude de serem zonas de quebra de linha de vista de um utilizador em andamento, relativamente ao percurso a seguir. Nesses pontos de interesse estarão colocadas marcas como a que se vê na Figura 1. A associação entre os identificadores das marcas (IDs) e os locais onde elas se encontram é guardada num ficheiro denominado 'assignedMarkers.txt'.

A estrutura de uma ou várias plantas de um edifício representada como um grafo, torna-se útil na altura de descobrir o caminho mais "curto" entre dois pontos da mesma. De facto, existem alguns algoritmos clássicos de pesquisa do caminho mais curto em grafos. Um deles, o algoritmo de Dijkstra [Cormen90], foi o utilizado no nosso sistema.

Para mais fácil edição e manutenção dos ficheiros acima referidos, desenvolveu-se um editor gráfico simples, denominado *Adjacent Points Editor – for Buildings Representation*.

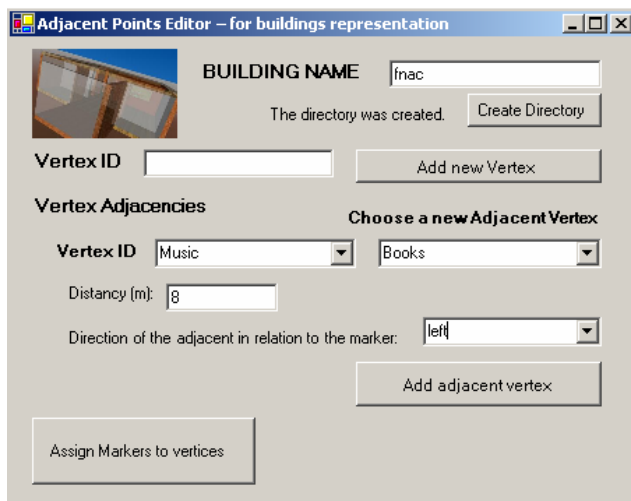


Figura 3. Exemplo de execução da aplicação ‘Adjacent Points Editor – for building representation’

No módulo de inicialização constrói-se assim a estrutura de dados que representa o grafo que representa o edifício. As duas formas mais comuns de representar computacionalmente um grafo são, ou como uma matriz de nós adjacentes, ou como um conjunto de listas de nós adjacentes. Seja o grafo $G = (V, L)$, em que V são os vértices (nós) do grafo e L o conjunto de ligações entre vértices. $|V|$ é o número de vértices e $|L|$ o número de ligações entre vértices. A representação do grafo como uma matriz de nós adjacentes é normalmente usada quando o grafo é *denso* - $|L|$ está próximo de $|V|^2$ - ou quando se pretende saber de uma forma rápida se existe uma ligação entre quaisquer dois vértices. Contudo, é mais usada a representação de um grafo como um conjunto de listas de nós adjacentes, pois consegue-se representar grafos *esparsos* - aqueles nos quais $|L|$ é muito menor que $|V|^2$ - ocupando muito menos memória¹. É este o tipo de grafo que representa as ligações entre zonas de um edifício. Por esse motivo a representação de grafo que usámos foi a última que se referiu.

A representação do grafo $G = (V, L)$ como lista de adjacências consiste numa matriz de *Adjacências* com $|V|$ listas, uma para cada vértice em V . Para cada $u \in V$, a lista de adjacências *Adjacências*[u] contém todos os vértices v para os quais há uma ligação $(u, v) \in L$, isto é todos os vértices adjacentes a u em G .¹

No exemplo de utilização da Figura 3, acima, na preparação da planta do “espaço fnac”, foi inserida a distância (8 metros) entre a zona da Música (Music) e a zona dos livros (Books) e está a ser inserida a relação espacial entre as duas zonas: a zona dos livros fica à esquerda (left) da zona musical. Obviamente esta relação espacial tem haver com o local onde se vai colocar a marca na zona da música.

¹ Cfr. [Cormen90], capítulo 23.1 *Representation of graphs*.

2.2 MÓDULO DE LOCALIZAÇÃO - Uma Solução Visual de Localização

A cada nó do grafo que representa um edifício, está associada uma marca visual (*fiducial marker*) que será interpretada e reconhecida pelo nível de Realidade Aumentada do nosso sistema.

Reconhecida uma marca, o sistema descobre o local onde o utilizador se encontra. Este sistema de localização é consideravelmente menos oneroso, quando comparado com o recurso a sensores activos, pois apenas são necessárias marcas impressas e uma câmara comercial integrada no PDA.

O módulo de reconhecimento de marcas, estimação da pose (posição e orientação) da marca relativamente à câmara de vídeo e de desenho 3D em Realidade Aumentada, corresponde a uma *dll* realizada em *Embedded C++*. Esta biblioteca dinâmica, a que chamámos *ARModule.dll*, encapsula uma versão modificada do *ARToolkit* especificamente adaptada para plataformas móveis (PocketPC e Symbian) [Wagner04]. Para o desenho 3D foi utilizada a biblioteca GPL Klimt, igualmente criada por Daniel Wagner [Klimt03]. A comunicação entre a aplicação principal e esta *dll* é feita através de uma técnica da plataforma .NET, o *Interop*. Resolvemos dividir o sistema nestas duas componentes, uma vez que desta forma conseguimos aliar a facilidade de desenvolvimento e programação característicos da linguagem C#, ao superior desempenho do *Embedded C++*.

O *ARToolkit* constituiu, pois, a base da nossa solução visual de localização do utilizador. O seu funcionamento processa-se da seguinte forma [Billinghurst03]: Primeiro, a imagem vídeo captada pela câmara é convertida numa imagem binária (a preto e branco), com base em valores limiares de luminância. Nesta imagem são procuradas regiões quadradas. O *ARToolkit* encontra todos os quadrados na imagem binária, muitos dos quais não são as marcas de localização. Para cada quadrado é capturado o padrão nele desenhado, produzida uma assinatura DCT que é comparada com os padrões conhecidos do sistema. Se houver uma correspondência, então o *ARToolkit* encontrou uma das marcas de localização. Esta biblioteca determina ainda a pose da câmara relativamente à marca. Envia então o ID unívoco da marca encontrada a uma função que identifica o local a que pertence a marca, quer seja por associação directa ou pelo conhecimento que tem do percurso já percorrido, conforme o esquema de distribuição de marcas que tenha sido usado, como se explicará na secção 3.

2.3 MÓDULO DE DESCOBERTA DO CAMINHO

Como já fizera Kalkusch no seu sistema de navegação para interiores, o *SignPost* [Kalkusch02], usámos igualmente o algoritmo de Dijkstra para descobrir, em tempo real, o caminho mais “curto” entre os locais de origem e destino do utilizador.

Não foi utilizado o algoritmo *breadth-first search*, um dos mais simples algoritmos de pesquisa em grafos, pois este algoritmo trabalha em grafos sem *pesos*, isto é grafos nos quais se pode considerar que cada ligação tem *peso*

(*custo*) unitário². No caso dos grafos que representam edifícios todas as ligações têm um *custo*: a sua distância (real ou arbitrária).

Necessitávamos de um algoritmo que, dado um nó de origem e outro de destino, encontrasse a sequência de nós cuja soma dos *custos* das ligações que os unisse fosse a menor. Este é um problema do tipo *single-source shortest path problem*, que tem como solução clássica, para o caso em que nenhuma ligação tenha *custo* negativo, o algoritmo de Dijkstra. O algoritmo de Bellman-Ford é usado no caso mais geral em que as ligações podem ter *custos* negativos³. Os procedimentos do algoritmo de Dijkstra descritos no livro *Introduction to Algorithms* [Cormen90], foram por nós concretizados na plataforma .NET, em C#, como aliás, grande parte do nosso sistema.

Para testar o algoritmo, desenvolvemos uma aplicação que constrói o grafo com os nós que vai importar a um ficheiro. A aplicação desenha os nós em posições relativas, que têm que ser especificadas nos ficheiros de nós, e permite ao utilizador escolher um nó de origem e outro de destino a fim de se procurar a sequência de nós com custo mais baixo entre esses dois pontos, correspondente ao caminho de “custo” mais baixo, ou caminho mais “curto”. Na Figura 4, podemos observar um exemplo de execução do algoritmo de Dijkstra, a determinar o caminho mais “curto” entre dois nós (b e f) de um grafo.

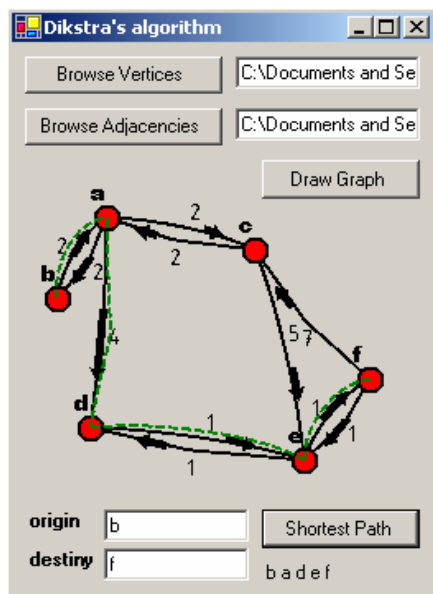


Figura 4. Aplicação de teste do algoritmo de Dijkstra.

2.4 MÓDULO DE SÍNTESE DE IMAGEM

Uma vez descoberto o caminho mais curto para o destino, o módulo de procura do caminho vai descobrir qual a relação espacial entre o *nó* onde o utilizador actualmente se encontra e o próximo *nó* desse caminho. Sabida essa

² Cfr. [Cormen90], capítulo 23.2 *Breadth-first search*.

³ Cfr. [Cormen90], capítulo 25 *Single-Source Shortest Paths*.

relação, a orientação do modelo virtual da seta é enviada para o módulo de síntese de imagem.

Na figura 5 pode ver-se dois exemplos de síntese de imagem de objectos virtuais na imagem real capturada pela câmara. Na imagem da esquerda o modelo indica que se devem descer as escadas e na imagem da direita que se deve seguir em frente.

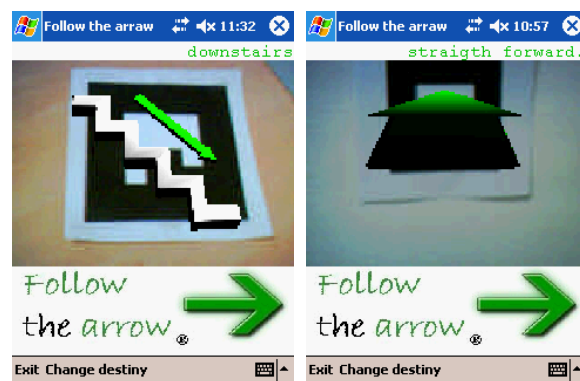


Figura 5. Dois exemplos de síntese de imagem de objectos virtuais na imagem real. Os modelos virtuais indicam o caminho que se deve seguir

3. ESQUEMA DE DISTRIBUIÇÃO DAS MARCAS PELO EDIFÍCIO

O número possível de marcas visuais a utilizar foi limitado pela versão do *ARToolkit* para PDA disponível à data da realização deste projecto. A nova versão entretanto publicada, o *ARToolkitPlus* [Wagner05], consegue analisar mais de 512 marcas, a uma taxa de 150 imagens por segundo. Como a nova versão ainda não estava disponível na altura da concretização do sistema, a solução usada foi reutilizar as marcas disponíveis, visto que para muitos edifícios haverá mais locais do que marcas disponíveis.

A vantagem de reutilizar marcas é que quanto menor for o número de marcas, menor será o número de comparações entre a assinatura DCT (*Discrete Cosine Transform*) da marca detectada pela câmara e as marcas registadas no sistema e, por outro lado (e mais importante), também será menor a semelhança e correlação entre as marcas. Estes factores diminuem o tempo e os erros na identificação das marcas e no registo de informação gráfica 3D nas mesmas.

A utilização de marcas únicas em cada local colmata por sua vez a grande limitação da reutilização de marcas: neste caso, o utilizador tem obrigatoriamente de filmar, senão todas, pelo menos algumas marcas que identificam zonas, consoante o esquema de reutilização usado, pois caso contrário o sistema não conseguirá discernir onde é que aquele se encontra. O sistema precisa de conhecer o contexto da utilização.

Na nossa opinião esta é uma limitação muito grande, visto que o utilizador com muita facilidade pode passar por algumas marcas sem as filmar, quer inconscientemente

quer conscientemente, quando já conhece parte do edifício ou por qualquer outro motivo presume um dado caminho.

Apesar desta limitação, nos casos em que as marcas disponíveis são insuficientes é necessário colocar marcas iguais em diferentes sítios, sendo nesse caso um dos requisitos da utilização do sistema que todas as marcas do caminho têm que ser filmadas.

Caso isso não aconteça o sistema perde a posição actual do utilizador dando-lhe a possibilidade de seleccionar o sítio onde se encontra, caso o saiba.

O nosso sistema está preparado para funcionar com os dois esquemas de distribuição de marcas: a reutilização de marcas e a utilização de marcas únicas, no caso de menos de 100 marcas serem necessárias.

A associação das marcas aos locais é feita automaticamente, através do já referido editor *Adjacent Points Editor – for Buildings Representation*, ficando registada no também referido ficheiro “assignedMarkers.txt”, dentro da pasta do respectivo edifício.

A primeira coisa que o algoritmo de distribuição de marcas faz é contar o número de nós introduzidos no editor. Se esse número for superior a 100, é seguido um esquema de reutilização de marcas e, na primeira linha do ficheiro “assignedMarkers.txt”, é introduzida a expressão ‘MARKERS_SCHEME reusable’. Se o número de nós for inferior a 100 será usado um esquema mais simples, para atribuição de marcas únicas a cada nó. A primeira linha do ficheiro “assignedMarkers.txt” será então ‘MARKERS_SCHEME single_markers’.

Esta expressão, que identifica o tipo de marcas usado, será usada pela função que procura o local a que uma marca pertence. No caso de marcas únicas a identificação é directa.

3.1 REUTILIZAÇÃO DE MARCAS

A estratégia de reutilização de marcas usada foi uma adaptação de um algoritmo que resolve o problema da coloração de grafos. O conhecido *graph coloring problem* [Jensen95, Dirac51], lança o desafio de se conseguir um conjunto mínimo de cores para colorir os nós de um grafo, sem que quaisquer dois nós adjacentes tenham a mesma cor. O mesmo problema se apresenta para colorir o mapa-mundo sem ter dois países adjacentes com a mesma cor. À solução apresentada por P. Gopala Krishna [Krishna98], realizámos as necessárias alterações para que a mesma marca seja usada não apenas com a intercalação de um nó com uma marca diferente, mas sendo também intercalada por dois nós com marcas distintas. Isto é necessário, porque se tivéssemos por exemplo dois nós ‘A’ separados apenas por um nó ‘B’, nunca saberíamos em que sentido se tinha ido de ‘B’ para ‘A’.

Inicialmente tem-se um conjunto de 5 marcas para reutilizar. Se este não for suficiente, o sistema vai acrescentando marcas até que se consiga o resultado pretendido.

O algoritmo assume que os nós do grafo estão ordenados por ordem decrescente de grau do nó (número de nós

adjacentes). Para que isto aconteça utilizámos o algoritmo de ordenação *Selection Sort*.

3.2 ESTRATÉGIA DE RECONHECIMENTO DO SÍTIO A QUE PERTENCE A MARCA

Como a mesma marca está presente em vários locais do edifício, coloca-se a questão de saber a qual deles pertence quando for reconhecida pelo sistema. Pelos motivos já explicitados, assume-se que neste caso são filmadas todas as marcas ao longo do caminho indicado pelo sistema.

Quando uma marca é detectada na imagem, o sistema vai buscar todos os locais onde ela está presente ao ficheiro “assignedMarkers.txt”. Depois, se se trata da primeira marca filmada, confirma-se que é uma marca única correspondente a uma das entradas do edifício. Caso não seja, é pedido ao utilizador que indique onde se encontra.

Se a marca reconhecida pelo sistema já não é a primeira, sabe-se que o caminho para o destino já foi descoberto, pois o caminho mais curto para o destino escolhido pelo utilizador, é calculado assim que o sistema sabe onde é que este se encontra, ou seja, quando a primeira marca é identificada. Deste modo, pode-se procurar entre os locais trazidos do ficheiro “assignedMarkers.txt” se há algum que esteja presente no caminho mais curto calculado e pelo qual ainda não se tenha passado. Caso haja, assume-se que é esse o local onde o utilizador se encontra.

4. APLICAÇÃO FOLLOW THE ARROW

Follow The Arrow é a aplicação que corresponde ao nosso sistema de localização em espaços interiores. Pode-se ver uma imagem do ecrã inicial da mesma aplicação na Figura 6.



Figura 6. Ecrã inicial da aplicação *Follow The Arrow*. O utilizador escolheu o edifício ISCTE e o destino Adetti(HP).

Inicialmente é pedido ao utilizador que seleccione o edifício onde se encontra e depois o destino para o qual se pretende deslocar. Quando o utilizador carrega no botão ‘GO’ aparece um novo ecrã já com a imagem captada pela câmara do PDA. A partir desse momento o sistema analisa todas as imagens (*frames*) provenientes da câma-

ra de vídeo, através de um módulo do *ARToolkit*, o qual tenta identificar marcas conhecidas.

Quando uma marca é detectada no ecrã, uma seta com a direcção que o utilizador deve seguir, a partir desse ponto físico do edifício, para chegar da forma mais rápida ao destino (figura 5). Na Figura 7 podemos observar um utente a utilizar o sistema.

4.1 HARDWARE UTILIZADO

O PDA que utilizámos para testar a aplicação *Follow The Arrow* apresentada abaixo é um HP iPAQ equipado com um processador Xscale a 400 MHz. A câmara de filmar é uma VEO Photo Traveler 130S.



Figura 7. Utilizando a aplicação *Follow the Arrow*.

5. CONCLUSÕES E TRABALHO FUTURO

A aplicação *Follow The Arrow* revelou-se funcional, apesar da limitação de todas as marcas do caminho terem de ser filmadas. Em breve utilizaremos a nova versão do *ARToolkit*, que permitirá a utilização de um universo extenso de marcas únicas (até 512 marcas). Além disso, nos casos em que mais marcas sejam necessárias, colocaremos duas marcas lado a lado, o que aumentará em muito o número de ‘marcas’ disponíveis.

Mesmo não sendo uma ideia completamente nova [Kalkusch02], a concretização de um sistema de localização para o interior de edifícios pareceu-nos de grande utilidade, tendo em vista a crescente utilização de PDAs e a já consumada integração dos PDAs com os telemóveis.

Um trabalho futuro, mas já em andamento, será a passagem da aplicação *Follow The Arrow* para telemóveis do tipo SmartPhone, as quais suportam o ambiente de desenvolvimento que existe no equipamento PDA utilizado no nosso Projecto. O sistema está ainda a ser generalizado por forma a poder comunicar com um servidor remoto de plantas de edifícios, com a geração automática de grafos topológicos de cada edifício. Prevêem-se ainda, no curto prazo, a realização de estudos de avaliação usabilidade do sistema no apoio a eventos científicos a serem organizados no ISCTE e, ainda, uma demonstração nas instalações da *Microsoft* no *TagusPark*.

6. REFERÊNCIAS

- [Billingham03] Billingham, M., Shared Space Homepage, Junho 2003
www.hitl.washington.edu/research/shared_space/download/, pesquisa efectuada em Janeiro de 2005.
- [Cormen90] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; *Introduction to Algorithms*, 2nd edition, 1990, MIT Press and McGraw-Hill Book Company.
- [Dirac51] Dirac, G., "Note on the colouring of graphs", Math. Z. 54, 1951.
- [Jensen95] Jensen, R., TOFT, B., *Graph Coloring Problems*, Wiley-Interscience Series in Discrete Mathematics and Optimization, 1995.
- [Kalkusch02] Kalkusch, M., Lidy, T., Knapp, M., Reitmayr, G., Kaufman, H., and Schmalstieg, D. "Structured Visual Markers for Indoor Pathfinding". Proceedings of the First IEEE International Workshop on ARToolkit (ART02), 2002.
- [Kato03] Kato, H.; Billingham, P.; Poupyrev, I., *ARToolkit version 2.3.3*, November 2002.
- [Klimt03] <http://studierstube.org/klimt/>, pesquisa efectuada em Janeiro de 2005.
- [Krishna98] <http://www.geocities.com/krishnapg/graphcoloring.html>, pesquisa efectuada em Março de 2005.
- [Wagner03] Wagner, D., Schmalstieg, D. "First Steps Towards Handheld Augmented Reality", To appear in: *Proceedings of the 7th International Conference on Wearable Computers*, White Plains, NY, USA, Oct.21-23, 2003.
- [Wagner04] Wagner, D., Schmalstieg, D. "ARToolkit on the PocketPC Platform".
- [Wagner05] http://studierstube.org/handheld_ar/artoolkitplus.php, pesquisa efectuada em Junho de 2005.

A Graph-Matching Approach to Sketch-Based Modelling

Filipe Marques Dias Joaquim A. Jorge
Department of Information Systems and Computer Science
INESC-ID/IST/Technical University of Lisbon
R. Alves Redol, 9, 1000-029 Lisboa, Portugal
fil@immi.inesc-id.pt, jorgej@acm.org

Abstract

Automobile designers, when creatively designing a vehicle, have to bind by several shape constraints while also exploring new concepts. A natural and unhindered modelling environment for a designer to start modelling in is thus desired. We present our ongoing modelling system, AutoMake, that enables calligraphic modelling of a simple 3D object that resembles the body shape of a car. Our system provides an innovative way to recognize 3D objects by matching sketches against templates stored in a database, indexed by multiple edge graphs based on various views. Our approach enables users to specify a bounding box providing perspective and size information about model. The user next sketches a model that is matched against a template, using a multidimensional indexing structure.

Keywords

Sketch-based Modelling, Graph Matching.

1. INTRODUCTION

Modelling three-dimensional objects from sketch input has gained importance in the modelling field recently. This fact was motivated by the lack of user-centred concerns in modelling systems, preventing users from pouring their creative power directly upon 3D models without fiddling with bureaucratic modelling interaction concerns. This hampering of creative use forces the separation of tasks. In the industry, this translates into modelling experts working together with designers, causing a need for comprehensive and delaying communication due to misinterpretation and other physical impossibility issues [Dias03]. In order to solve this, the designer needs to build an early stage 3D model that can afterwards be refined and worked from by the modeller. Furthermore, in those early stages of product design, designers sketch creative drawings to explore ideas and produce rough sketches of objects that are usually bound to some general constrained specifications. This led us to the problem of providing a natural and unhindered modelling environment for a designer to start modelling in. We propose a calligraphic approach, based on 3DSketch [Mitani00, Mitani02, Varlet04], where the user sketches a visible-edge representation of one of several possible template models, and is able to manipulate, and edit it by re-sketching over edges as if correcting their curvature. Our approach allows the designer to draw with stylus and tablet or directly on screen, with a digitizing monitor, in a similar fashion as sketching on paper. Our approach applies fewer restrictions to models since unknown information is assumed to be specified by the template stored on the database.

The following section takes a look at other approaches in this area and points out their difficulties in completely solving this problem. The next section describes our approach, explaining how our approach sees the template drawing as a graph, and retrieves it from a data structure. The remaining section refers final considerations, ongoing work aspects, and directions to take for future development.

2. RELATED WORK

The topic of modelling 3D objects interactively from 2D calligraphic input is becoming a more common in the modelling context.

In 1994, IdeS [Branco94] appeared with a modelling scheme based on reconstruction and gesture recognition, for use in a mechanical engineering context. The user was able to draw the solids' visible edges and issue some modelling commands by means of gestures, like extrusions, cuts, and other operations involving more objects. Later, this work saw vast improvements, when context-based modelling aid was introduced in its successor, GIdeS [Pereira00]. GIdeS improved on the previous work in many key areas, the main of which: the user interface, and the addition of context-based functionality to cope with the ambiguity natural to reconstruction. The latter is achieved by an ingenious expectation list that presents several previews of the final result depending on the applicability of operations. The modelling approach present in that system reflects the concerns with user interaction, corresponding to a constructive way of modelling.

SKETCH [Zelesnik96] introduced another gesture-based interface for approximate 3D modelling, with some added

support for animation. With this system, the user follows a few rules to draw specific features of supported primitives. The system then instantiates them in 3D space appropriately. Its modelling methodology is slightly different from the previous GIdE work, as the model is constructed from separate objects (parts) that are grouped together automatically.

Chateau [Igarashi01] is a system that enables the definition of polygonal shapes and objects, and suggests results of candidate operations based on hints – a similar approach to the aforementioned expectation lists. Chateau highlights entities drawn, to indicate the hints that will be used for suggesting operations. It is up to the user to toggle the highlight status of the entities, to appropriately tailor the scope of possible operations to his desire. Suggestions are generated each time entities change highlight status.

RIBALD [Varley00, Varley04] is a system that is able to produce a boundary representation model from an imperfect line sketch of a manifold polyhedral object. Its approach is based on interpretation, to create solid objects. It starts by labelling lines (concave, convex, or occluding) and junctions; later produces visible geometry, assigning provisory depth values to vertices; and identifies local symmetry features. Additionally, the system tries to classify objects to attempt at speeding the reconstruction using specific case methods. Objects produced are a variety of solid models that can be made from non-perfect line input.

3DSketch [Mitani00, Mitani02, Varley04] reconstructs and renders particular models based on sketch input using non-photorealistic rendering techniques maintaining a sketchy appearance. It works by generating an edge graph from a set of input sketches, reconstructing a model from that graph and some assumptions, and finally rendering it. The system is focused on reconstructing only objects that have 3 perpendicular faces visible, have a flat based and back faces, and possess mirror symmetry. Some of these constraints are necessary for estimating camera parameters due to drawing in perspective. This system also enables de definition of curved edges and surfaces.

These approaches tackle the problem of constructing relatively simple 3D geometry from 2D sketch input, either using a set of commands the user must issue by means of a gesture, or by interpreting the model from a starting set of constraints or existing template. The gesture approach has the drawback of forcing the user to know the commands to issue, and often these do not correspond to the most obvious pen movements. This way of creating faces and surfaces is not appropriate for sketching creative drawings, as interruption is undesired. On one hand, interpretation provides a higher sketched fidelity for visually representing objects. On the other hand, templates, allow restricting the domain of objects to a controllable scope, simplifying reconstruction, and enabling the addition of curvature to an otherwise difficult to reconstruct

drawing. However, in Mitani's 3DSketch approach, this scope is limited to a single template.

Our system is based on 3DSketch, in a way that the starting point for modelling is well defined by a template, but allows the existence of a vast number of different templates. Each templates is stored in a database using edge graphs produced from several views of it, so it can be retrieved when a users draws it.

In a context of image-based similarity of models, [Chen03] calculate several views of objects in a database. They use these views to do image-based indexing using Zernike moments and retrieve similar objects. That approach is based in that if 3D models are similar, then they have similar views. They use the vertices' positions of a dodecahedron around the object for placing a camera pointing at it, and render the various views. In our approach we also position a "camera" on a dodecahedron's vertices, but with the difference: we build an edge graph instead of rendering an image. A single template model is stored in an NBTtree data structure [Fonseca04] using those graphs as indexes.

3. OUR APPROACH

Our system is based on work by [Mitani00, Mitani02, Varley04] for matching perspective, and the model to graph. However, that approach takes into account a single fixed object template. This drastically limits the usefulness of a modelling application to cope only with an object of a single topology. Among car body shapes, the number of visible vertices on a template can vary depending on the type of vehicle, and this led us to consider more templates in our approach. Our system also has an increased simplicity dealing with more complex templates.

3.1 Overview

A template is a simplified linear-edged model that not only represents the connectivity of vertices on the desired object, but also specifies its rough desired shape, dimensions and some other constraints. A template can be seen as a simplified model, with its key characteristic features specified. Figure 2 shows a template, on the left.

Using our system, the user starts by sketching, in perspective, a bounding box of the object he wishes to draw. This simple step has many uses, as mentioned in the next subsection.

The user then draws the object he intends to represent, by sketching its visible edges. The system sees these edges as edges on a 2D graph, and builds that representation internally. This graph corresponds to the 3D object as viewed from the viewpoint it was drawn in.

This graph is then used for building a numerical vector to query a database, through its adjacency matrix representation [Fonseca04], as described next. This query will be executed on a database to retrieve a set of templates that most resemble the drawn graph. Then the right one is chosen and the user can take part in that choice.

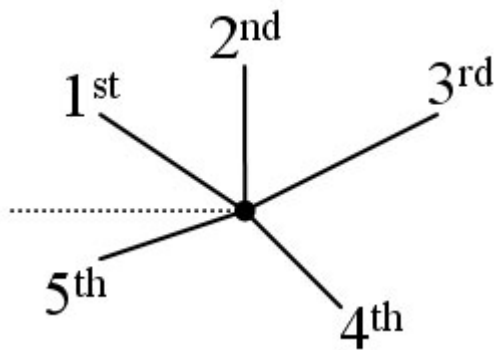


Figure 1 - Order of a node's edges

From this point onwards the user should be able to edit and refine the model interactively.

3.1.1 Bounding box

The simple step of drawing a bounding box has many uses. Its key function is based on the designers' own methodology of drawing, making the user confident and comfortable. Not only will the user be specifying a self guiding method for more accurately express his intent, he will also be less prone to fall into disproportion errors, non-parallel lines, and other unwanted distortions.

As a second advantage, the user will also be revealing certain aspects of the object that the system may use, in a posterior phase, to help cope with ambiguity. Such clues are perspective information and rough proportions. To determine these clues, the system calculates the intersections of line segments for bounding box's edges and their lengths. For perspective information, the linear extensions of the bounding box's edges are used. The intersection of these enables estimation of vanishing points. With these points determined, camera parameters are calculated. The estimation of vanishing points and camera parameters is similar to that described in [Varley00].

A third use is extracted from this bounding box, as well, which is the expected size of the graph, useful for knowing when the drawing is nearing completion. At a later phase, the system must build the graph representation and issue a query with it to a database, to retrieve the template for posterior modelling. The system must either be explicitly issued by the user to do this, or it must estimate the correct time for it. In order to estimate it the system looks for a closed graph, without degenerate edges, that roughly spans the extension visible faces of the bounding box.

3.1.2 Graph Representation

As mentioned, the sketch drawn by the user is stored as an edge graph representing the 2D projection of the 3D object for the given view. Each graph node corresponds to a visible vertex in the object, and each edge to an edge of an object's visible face. The list of edges a graph node has is ordered in a way that edges appear in a clockwise order, as shown in Figure 1.

The purpose of ordering edges is twofold: to make sure the graph is traversed simply in a coherent manner, and to make edges belonging to a same face adjacent in the rep-

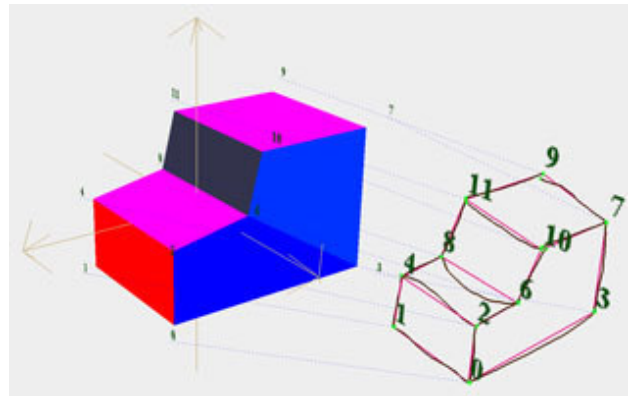


Figure 2 - Vertex numbering and correct match between object graph and template

resentation. Consequently, the graph needs only a pre-processing phase to order the edges, enabling all subsequent traversals to be made simpler and faster. However, this also means the graph may require partial re-ordering if the view is changed substantially, in such a way that new edges become visible or invisible, or when a new starting node needs to be calculated for the graph.

3.1.3 Template Indexation and Retrieval

In order to store and retrieve template objects, a numerical vector needs to be computed. These vectors index template models, stored along with camera position information, in a multidimensional indexing structure (NBTree [Fonseca04]). A vector is composed of the eigenvalues for the adjacency matrix representation of the drawn edge graph, and the retrieval of such vectors, is a stable problem [Fonseca04]. This means that small variations in the graph used for searching will retrieve results in a vicinity of the original graph.

After a retrieving a set of templates, the system has to choose the best one among them. Our approach allows something similar to an expectation list, like GIdES [Pereira00] or Chateau [Igarashi01], to be used, enabling the user to take part in more specifically opt for the desired template. Nonetheless, rating possibilities should be made possible using proportion measures based on the drawn edge graph, and perspective hints from the bounding box as well. Though, it is possible that the graph drawn does not correspond to a template stored. In such a case the template returned is the one that is most closely matched. At present, our system assumes that only one best fitting template is returned by the query made to the indexing structure, and that it topologically resembles the drawn graph.

3.1.4 Graph Matching

With a candidate graph selected, the system traverses it, and the graph is built from the user's drawn input, in the same order establishing correspondences between vertices, edges, and faces.

To start traversal, a leading vertex must be chosen for the graph's starting node. Because we assume the object is drawn from an "informative" point of view, and that 3

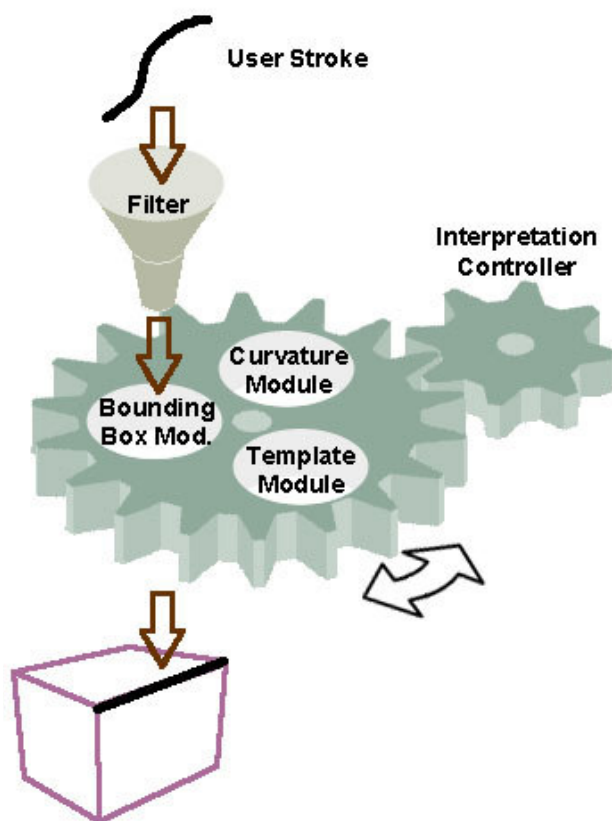


Figure 3 - Modular system architecture for interpreting a stroke depending on modelling context

faces of the bounding box are visible, we simply chose the first vertex to be the leftmost lower vertex (vertex 0 in Figure 2). The system pays more attention to non-dihedral vertices, as these vertices only have two edges. This gives higher flexibility to deal with graphs of slightly different topology, weakening the constraint on both graphs having the same starting node. This means the starting node must be on the same face, instead.

Having picked the starting node, the matching traversal progresses normally, comparing the number of edges for each non-dihedral node and each face. In the end the dihedral nodes are matched. If that match is not possible, the vertex information on the template prevails, if both graphs are successfully matched.

We do not reconstruct an object in the same way 3DSketch or RIBALD do. Instead we use the information stored in the database for specifying the edges. In the future our system will allow the user to change them on the next modelling phase.

3.2 Sketched entities

RIBALD [Varley00, Varley04] assumes the object is drawn from “most informative viewpoint”. In our approach, this assumption is replaced by a less restrictive one: a viewpoint that produces an edge graph of the same topology. This is because the user is able to change viewing position after drawing the bounding box. Though, like them, we also assume no through holes are present, and

that edges form a single graph. Like [Mitani00] for determining perspective, we assume the bounding box is drawn with 3 perpendicular faces visible. 3DSketch must assume the back face is planar because no invisible vertex or edge is represented. However, with our approach that information comes from the retrieved template object. The user can also navigate around the object and refine it.

Regarding curvature, our system does not currently support it. However, we are planning to follow the same principles described in [Varley04]. As mentioned there, the information needed to define curvature from a single view involves specification of symmetry or other constraints. That information is accounted for inside our template. This would mean curve edges would be estimated at the time of matching, similarly to [Mitani00]. In such cases when curvature estimation is not possible due to lack of information (a symmetrical edge is not visible) our approach will revert to the default edge information saved on the template.

3.3 System Architecture

Our system is organized in modules that wait their turn before acting upon a stroke drawn by the user, depending on context. Figure 3 illustrates this organization.

When the user draws a stroke, it goes through a basic filtering (re-sampling) process to reduce the high number of points. Drawn strokes are then stored in memory to allow posterior access to them, for changes to be made.

Next the filtered stroke is passed to an “interpretation” module assigned by a controller based on context. The module that processes the first strokes is the one responsible for creating the bounding box. This module interprets strokes as linear edges forming a box. This module also estimates de camera parameters mentioned earlier.

When the bounding box is completed and perspective information has been determined, the controller switches the active stroke interpretation module to the Template Module, responsible for building the graph.

After a template has been retrieved from the database, the system can then switch interpretation control over to another module that can interpret strokes differently; for instance, a module for changing curvature of edges.

4. CONCLUSIONS AND FUTURE WORK

In this paper we presented our ongoing calligraphic object modelling approach that matches a drawn graph against a template stored on a database. The user first draws a bounding box for the model, and then sketched the object inside it. Matching these allows the generation of a 3D model that can later be refined.

We showed that the use of the information extracted from the bounding box at the beginning of modelling can be useful for the modelling approach, as it is to the user drawing it and the modelled result. The modelled object is no longer confined to being of a single “boxy” nature, and the user sketches his sketch using his learned drawing techniques.

Regarding the ability to deal with several templates – provided by the use of an indexing structure – we mentioned the stability of retrieving vectors computed from eigenvalues of adjacency matrixes. This encourages us to continue pursuing our approach, improving its modelling capabilities.

We should also find rating solutions for selecting graph similarity based on edge length, dimensions and other features in order to better sort the possibilities returned by a query to the indexing structure.

Our approach allows the user to interact with the model between estimation of camera parameters and definition of the object. This allows us to cope with “not-so-informative” points of view, which RIBALD and 3DSketch are unable to model in. This is an advantage. Having the number of possibilities stored in the database as the limiting factor, our approach has the potential for being more robust. However, this still requires substantial practical testing to confirm.

Another topic to research is the possibility of users specifying invisible edges by using pressure or a lower number of overlapping strokes. Extra information provided by hidden-curve specification may override the default behaviour of depending on the template, and allows for more freedom in choosing a point of view to draw the model – as some invisible faces may be specified.

The curvature in edges is ignored in the current implementation of our system, but it can be added in a similar fashion as in 3DSketch.

When the database is filled with more templates, we believe that we will achieve our goal of allowing a designer to explore his creativity unhindered, using the calligraphic techniques natural to his skills.

5. REFERENCES

- [Branco94] Branco, V; Ferreira, F N; Costa, A: "Towards an Intuitive 3D Interaction", *Third Luso-German Computer Graphics Meeting (Conference Proceedings)*, Coimbra, Portugal, 1994.
- [Chen03] Chen, Ding; Tian, Xiao; Shen, Yu; Ouhyoung, Ming; "On Visual Similarity Based 3D Model Retrieval", In *Proc. Eurographics*, 2003.
- [Dias03] Dias, Filipe; Jorge, Joaquim A. "Task Analysis and Scenario-Based Design of Calligraphic Interfaces", *12^o EPCG*, Porto, Portugal, Oct 2003.
- [Fonseca04] Manuel J. Fonseca. "Sketch-Based Retrieval in Large Sets of Drawings". *PhD thesis*, IST/UTL, Lisboa, Portugal, July 2004.
- [Igarashi01] Takeo Igarashi and John F. Hughes, "A suggestive interface for 3D drawing", *UIST*, 173-181, 2001.
- [Mitani00] Jun Mitani, Hiromasa Suzuki, Fumihiko Kimura, "3D Sketch: Sketch based Model Reconstruction and Rendering", *IFIP Workshop Series on Geometric Modeling: Fundamentals and Applications, IFIP Working Group 5.2, Seventh Workshop GEO-7*, University of Parma, Parma, Italy, October 2-4, pp.85-112 (2000).
- [Mitani02] Jun Mitani, Hiromasa Suzuki, Fumihiko Kimura, "3D sketch: sketch-based model reconstruction and rendering", *From geometric modeling to shape modeling*, Kluwer Academic Publishers, Norwell, MA, 2002.
- [Pereira00] Pereira, J. P.; Jorge, J. A.; Branco, V. A.; Ferreira, F. N.; "Towards Calligraphic Interfaces: Sketching 3D Scenes with Gestures and Context Icons", *WSCG - International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, Feb. 2000.
- [Varley00] P. A. C. Varley, H. Suzuki, J. Mitani, R.R. Martin: "Interpretation of Single Sketch Input for Mesh and Solid Models", *International Journal of Shape Modelling* 6(2), 207-241, 2000.
- [Varley04] P. A. C. Varley, Y. Takahashi, J. Mitani, H. Suzuki: "A Two-Stage Approach for Interpreting Line Drawings of Curved Objects", *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modelling 2004*, Grenoble, France, 2004.
- [Zelesnik96] Zelesnik, Robert C., Kenneth P. Herndon and John F. Hughes.; "Sketch: An interface for sketching 3d scenes," *Computer Graphics, SIGGRAPH 96 Proceedings*, August 1996, pp. 163-170.

Direct Manipulation of feature models using Handles

Daniel Lourenço

DEI / IST

Lisbon, Portugal

daniel.antunes.lourenco@gmail.com

Pedro Oliveira

DEI / IST

Lisbon, Portugal

pedro.vcm.oliveira@gmail.com

Rafael Bidarra

EEMCS / TU Delft

Delft, The Netherlands

r.bidarra@ewi.tudelft.nl

Abstract

In feature based systems support for direct manipulation is not commonly available. This is partly due to the strong reliance of feature modelling systems on constraints and on the lack of speed of current constraint solvers. In this paper an approach to the optimization of the geometric constraint solving for the specific situation of direct manipulation is described. Also a solution for a direct manipulation interface was designed that brings to feature modelling the advantages of direct manipulation while taking into account the main feature modelling paradigm concepts. Details are provided on how it was implemented successfully in the SPIFF feature modelling system.

Keywords

Feature-modelling, Direct manipulation, CAD/CAM

1. INTRODUCTION

1.1 Feature Modelling

Feature modelling is a design paradigm that aims to bring the modelling task to a higher level in a way that it is easier for the designer to specify and understand elements and their relations in a model. This is done by associating functional information to the shape information in the product model.

The feature modelling process is done based on features. [Bidarra99] defines a feature as a representation of shape aspects of a product that are mappable to a generic shape and are functionally significant for some product life-cycle phase. This way, unlike when the design is based in geometry alone, in feature modelling the designer builds a model out of features that have a well-defined semantics.

A feature class library is a collection of feature classes that are made available to the designer. Each feature class has a specified semantics (e.g. a specific machining operation) and can be instantiated as a feature in the model by providing a set of instance parameters.

Constraints are a crucial concept for feature modelling systems and can be used to express high-level characteristics of the model (e.g. restrict the volume of a product to a certain maximum). Constraints can also be used as internal elements of features that express the feature semantics. Systems that force the maintenance of each feature's semantics in a feature model throughout the modelling process are called semantic feature modelling systems [Bidarra99]. Because of this intensive usage of constraints, feature modelling systems have to make an extensive usage of constraint solving techniques.

1.2 Limitations of Traditional Feature Manipulation

In the SPIFF system the specification of feature parameters is done solely through the input of values in dialog boxes. When a user edits a feature, the model is updated with the new user values and then fully solved to determine the new geometry.

The main disadvantages of this approach are:

- inefficient feedback, making the design task much slower. Each time the designer changes the parameters of a feature he has to wait for the whole system to be solved and only then can he see the effect of his changes and check the validity of the model.
- lack of insight on the consequences of the modelling operation. When changing a parameter the user can only see the original and resulting model of the operation. This means that there is no explicit feedback on which features were affected and how.
- non-intuitiveness due to the fact that the user is simply editing dialog boxes that do not express how the feature is affected by the parameter.

1.3 Project Goals

The main goal of this project was the development of a direct manipulation interface for form-features in the system. In particular feature handles were added as a way of specifying feature parameters. As the modelling operation occurs real-time feedback has to be provided to the user¹. With this improvement the design process will be improved so that the user may interactively change

¹ by real-time we mean fast enough for the user to have a direct manipulation experience and not the strict meaning of the real-time expression

the value of the parameters of features in the model while he is being provided with feedback of the changes. When the user is satisfied with the model he can choose to make the changes final and check the model validity.

Two critical issues had to be tackled in order to provide the real-time interaction required -- efficient model constraint handling and efficient display of the model geometry being manipulated by the user².

One other orthogonal problem that had to be solved was to determine which should be the characteristics of the feature handles available for each feature (where they are positioned; how they are specified; how they behave; how does their manipulation affect the parameter being changed).

1.4 Paper Overview

On this paper we will first start by giving an overall insight of the what the SPIFF system is, its architecture, main concepts and how they relate and are implemented internally. This will be done in section 2.

In section 3, after a brief introduction on how constraint solving works in SPIFF, we will present a solution for constraint solving that copes with the efficiency requirements of real-time feedback for direct manipulation.

In section 4 we will state the designed target manipulation requirements and how these were materialized in the SPIFF system.

In section 5 we give a brief overview of what was done while concluding whether the main objectives were achieved and identifying the limitations of the solutions found.

2. THE SPIFF SYSTEM

The SPIFF system [SPIFF] is an advanced feature modelling system developed at the Delft University of Technology, the Netherlands. Development on this system started in 1994 and work on it has been going on since then leading to a system which incorporates many different ideas and concepts on feature modelling and which is therefore quite complex.

The underlying geometric kernel of this system is the ACIS modelling kernel.

This system supports many advanced feature modelling concepts including the concept of semantic feature modelling [Bidarra00].

2.1 Feature Classes in the SPIFF System

A feature class in the SPIFF system can be seen as a specification of its shape and the set of parameters needed to instantiate it.

The shape of a feature class is determined based on the concept of basic shape. A basic shape works as a meta-feature that only specifies the geometry it represents (e.g. cylinder). This representation is a set of model elements (e.g. faces) and constraints that specify their relations. Features acquire their geometry by deriving from a specific basic shape.

Besides its shape, a feature class also contains a number of constraints and variables which further define its shape and semantics and the way it relates with other features in the model.

2.2 Feature Model

The feature model in the SPIFF system has multiple internal representations. One of these, on which we will focus, consists of the High Level Constraint Graph. Its nodes contain the variables and the constraints applied upon them and their relationships are established through the edges. This graph aggregates the definitions of every feature in the model and the relations between them. Each feature is present as a set of shape constraints, variables (e.g. faces and parameters) and other constraints. In figure 1 we can see a simplified version of a High Level Constraint Graph.

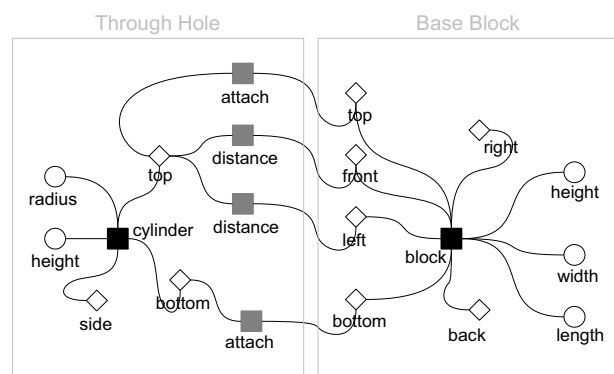


Figure 1: Sample constraint graph

Only two types of variables exist:

- geometric variables which represent geometric entities of the model. These variables are defined by a position and an orientation. From the point of view of our problem, we can assume that only two types of geometric variables exist: faces and references. References are model elements like lines, points or surfaces that can be added to the model.
- value variables which contain several types of numeric values.

As for the constraints they can be of several types but, for this context, it suffices to explain the following three:

- geometric constraints which restrict the way geometric variables are positioned (e.g. distance between two faces).
- shape constraints which are similar to geometric constraints but specify the relation between all the faces of a basic shape and between its value variables.
- algebraic constraints which specify algebraic relations between value variables (e.g. equal, division).

Other constraints that exist that are not relevant for this paper are the boundary constraints and dimension constraints. These constraints are not used for the definition of model values but only to further enforce the semantics of each specific feature. One example of a boundary con-

² we will not approach the latter issue on this paper

straint is that the top face of the through hole in Figure 1 must not be on the boundary of the model.

Each time a feature is added to the model, its corresponding variables and constraints are added to the High Level Constraint Graph. It is by solving this graph that the position and orientation of each geometric variable is determined and, thereafter, the full geometry of the model is known and can be shown to the user.

One other internal model of the SPIFF system is the Cellular model. This model is a geometric model which is used both for display and for the validity checking of boundary constraints. It is implemented on top of the geometric kernel using its data structures and operations.

3.REAL-TIME CONSTRAINT SOLVING FOR DIRECT MANIPULATION OF FEATURES

As was seen in the introducing section, a feature model in the SPIFF system is internally represented as a constraint graph. To know the geometry and to check the semantics of the model, the whole graph has to be solved. After the solving process, if the model is valid, the value of all the variables in the system is determined.

In order for the system to solve the graph it follows a number of steps of which the most important are the solving of specific mappings of the High Level Constraint Graph by both an algebraic and a geometric constraint solver. These solvers are, respectively, the SkyBlue solver [Sannella92,Sannella93] and the Kramer Solver [Kramer92].

A performance analysis has shown that the two most important bottleneck of this solving process are the execution time of the geometric solving and the evaluation of the cellular model.

The evaluation of the cellular model can be skipped by displaying only a simplified model. As the boundary constraints validation depends on the cellular model it cannot be performed.

Since the cellular model evaluation was avoided the optimization efforts were focused on the Kramer Solver. We next briefly describe how it works and what was changed in order to make the system faster for the direct manipulation situation.

3.1The Kramer Graph

The specific mapping of the High Level Constraint Graph which is used by the Kramer Solver is called Kramer Graph. It is composed of constraints and variables. In this graph the variables are called geoms and represent a coordinate system which is in a specific position and orientation in space. This position and orientation are not fixed³ but are restrained by the constraints in the graph. It is by determining which is the position and orientation for each geom that satisfies all constraints that the final values for the variables are determined.

Each constraint reduces the relative degrees of freedom between geoms by specifying restrictions to their position and orientation.

To each geometric variable of the High Level Constraint Graph will correspond exactly one geom and to each constraint will correspond one or more constraints in the Kramer Graph.

3.2Solving Strategy

To explain how the solving actually works we first need to define the concept of joint. A joint is no more than the set of constraints between two geoms. This means a joint is what restricts how two geoms are positioned in relation to each other. We call a joint rigid if it leaves no relative degrees of freedom between the two geoms it relates i.e. their position and orientation is fixed in relation to each other.

The Kramer Solver works by iteratively determining which joints are rigid and joining them into new geoms called macro-geoms. By progressively joining geoms a point is reached when there is only one geom. At this point the system is solved.

One final remark is that the Kramer Solver is able to handle models which are not fully constrained. In this situation it merges geoms up to the point where it can no longer find rigid joints. The result will be a graph with more than one remaining macro-geom. This fact is crucial for this work as will be explained in the next section.

3.3Geometric Constraint Solving for Direct Manipulation

The key remark to understand the optimization proposed is that, when the user is directly manipulating a parameter, only its corresponding variable is being changed during the direct manipulation session⁴. As a result, all its solving operations are highly repetitive because most of the graph is exactly the same. From here we can devise a solution that attempts to avoid this repetitiveness.

The idea is to put the ability of the solver to handle under-constrained models at the service of the repeated solving for the same parameter. Our solution will be to add a preprocessing step to the direct manipulation session that executes the computations that otherwise would have been repeated and encapsulates their result in the constraint graph. For each interaction step only the operations that depend on the parameter are executed.

This solution will be composed of two steps: 1) the preprocessing step solving and the 2) interaction step solving.

In the preprocessing step the High Level Constraint Graph is mapped without the constraints which are determined by the parameter being changed by the user. The removal of these constraints makes the mapped Kramer Graph under-constrained and therefore, after solving, we will have a solution with a set of macro-geoms. We call this solution the preprocessed graph. This graph is precisely the structure that encapsulates the repeated part of the solving operations. It will be used as the starting point for solving in subsequent changes of the parameter.

³ apart from a small number of geoms which have an absolute position and orientation in space

⁴ by direct manipulation session we mean the period of time while the user directly manipulates one parameter

For each of the subsequent solving operations the interaction step solving is executed. In this step the constraints determined by the parameter being changed will be mapped to the preprocessed graph. This graph is now fully constrained and can be solved to give us the solution for the current value of the parameter. This solving step is much faster than solving the whole model as most of the repeated solving is skipped.

4. MANIPULATION OF FEATURES USING FEATURE HANDLES

4.1 Target Manipulation Interface

The direct manipulation facilities designed must deal with the three disadvantages of traditional feature manipulation mentioned in section 1.2.

For efficient feedback every modelling operation has to lead to an immediate preview of the result of the operation to the user. This preview will be provided through a transparent overlay display of the resulting model. The user will have insight on the consequences of the operation through the comparison of the original model with the preview which are displayed simultaneously. The manipulation will be intuitive because of the feature handles' characteristics.

The deployed feature handles must take into account that each feature parameter has a specific semantics. This semantics will be expressed through the handle's behaviour, positioning and graphical representation.

4.2 Types of Feature Handles

In this work it was decided to deal only with parameters which contain numerical values which generally have a simple geometric meaning.

In SPIFF the following four types of handles are enough to cover the direct manipulation needs: linear handles; angular handles; planar handles; slider handles.

In this proposal all handles of a feature are specified by their relation with reference elements (e.g. reference points and reference lines) of the feature. With them the system will be able to know the placement and behaviour of the handles. One reference that is needed for the specification of any type of handle is the point that specifies its position in the model. These references are specified at the feature definition like any other references - using constraints. One very important consequence of the definition using references is that the position and orientation of the handles will be determined by the constraint solver alone.

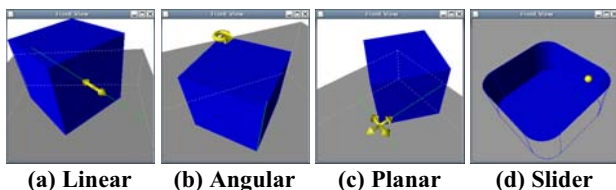


Figure 2: Types of Handles

We will now look with more detail into each of the proposed handles:

- A linear handle is a handle that moves along a straight line and reacts linearly with the mouse movement. It contains a reference line which represents the line along which the handle moves. See figure 2(a).
- An angular handle is a handle that moves around a rotation axis in a way that the user is able to specify an angle parameter. This handle has a reference line which represents the axis of rotation around which the angular handle revolves. See figure 2(b).
- A planar handle behaves somewhat like a linear handle with the difference that, instead of having its movement restricted to a line, its movement is restricted to a plane (having two linear degrees of freedom). For this type of handle two reference lines will be needed to determine the plane on which it can move. One thing that differs from this handle to all the others is that when manipulated it affects two parameters of the feature instead of one. The motivation for the existence of this handle is to enable the user to change the position of features that have it specified by two distances to external faces. See figure 2(c).
- A slider handle is also similar to a linear handle but, in this case, the line will simply be a vertical line on the viewport with no connection to the actual feature geometry. Therefore this handle will contain no reference lines. The slider handle adjusts nicely to parameters that have no simple geometric meaning but should be directly manipulatable. See figure 2(d).

4.3 Handle Manipulation Flow of Events

Two important moments exist in the handle manipulation flow of events: the moment when the user selects a feature to be edited and each time a handle is dragged.

When the user selects a feature to be edited this leads to the activation of all the handles of the feature -- the handle is displayed and registered in the Operator (a control entity). The feature is also highlighted with a transparent overlay.

When a single drag event of the user occurs the drag event is reported to the Operator with the information of the new mouse position. The operator sends a message to the Handle which leads to the computation of the new value of the parameter and to its update in the model. This computation is obviously done taking into account the handle behaviour (see section 4.2). After the new value is set, the Operator orders the Constraint Manager to solve the model. With the new model values resulting from the constraint solving process a transparent preview of the result of the operation is rendered and displayed.

5. CONCLUSIONS

An approach for 3D geometric constraint solving for direct manipulation was developed. This approach has been successful at increasing the speed for the direct manipulation situation and, therefore, improving the user experience. The average improvement on the speed of geometric solving for the tests performed was of 18 times. With full frames being generated in 290ms for medium sized

models (13 features) we believe the objective of providing the user with real-time manipulation has been achieved. Of these 290ms only 50ms are for geometric solving therefore further optimization work mostly has to be performed on other parts of system⁵.

One problem concerns the large amount of time required for preprocessing before the first frame of the manipulation session can be shown. This happens because of the necessary underconstrained solving process which is slower. Unfortunately, at this point, we have no promising idea on how to improve this time.

In spite of the problems that still stand, the overall outcome regarding the work on geometric constraint solving was clearly positive with impressive gains at the geometric constraint solving performance.

A whole new architecture for the direct manipulation with handles for a feature modelling system was designed and implemented. Four types of feature handles were designed and no situation was found in which these four types of handles were not enough. For parameters with a clear geometric meaning this solution was fully successful in making the manipulation process intuitive⁶.

For an effective and insightful feedback on the modelling operations, a transparent preview of the model is overlaid as the user directly manipulates a feature. This way he can clearly see the effect of the modelling operations and compare the result with the original situation.

Through out this project separate parts were developed. They build on top of each other to provide a system which allows the user to perform direct manipulation of features in real-time. Although many improvements can

still be made the product of this project provides a sound foundation for further developments.

6. REFERENCES

- [Bidarra99] Bidarra, R. (1999), Validity Maintenance in Semantic Feature Modelling, PhD Thesis, Delft University of Technology, Delft, The Netherlands
- [Bidarra00] Bidarra, R. and Bronsvort, W.F. (2000) Semantic feature modelling. *Computer-Aided Design* 32(3): 201-225
- [Bunnik00] Bunnik, A. (2004), Web-based direct manipulation of feature models, MSc Thesis, Delft University of Technology, Delft, The Netherlands
- [Dohmen97] Dohmen, M. (1997), Constraint-based feature validation, PhD Thesis, Delft University of Technology, Delft, The Netherlands
- [Kramer92] Kramer, G. A. (1992), Solving geometric constraint systems: a case study in kinematics, The MIT Press.
- [Noort02] Noort, A. (2002), Multiple-View Feature Modelling with Model Adjustment, PhD Thesis, Delft University of Technology, Delft, The Netherlands
- [Sannella92] Sannella, M. (1992), The SkyBlue constraint solver, Technical Report 92-07-02, Dept. of Computer Science and Engineering, University of Washington, USA.
- [Sannella93] Sannella, M. (1993), The SkyBlue constraint solver and its applications, in "First Workshop on Principles and Practice of Constraint Programming".
- [SPIFF] The SPIFF system homepage,
<<http://www.cg.its.tudelft.nl/~spiff/spiff.html>>
- [Wikipedia] Direct manipulation interface, Wikipedia,
<http://en.wikipedia.org/wiki/Direct_manipulation_interface>

⁵ tests performed on an Intel Centrino 1.3Ghz with 512Mb of physical memory and an ATI Radeon Mobility 9000 graphics card

⁶ in truth this conclusion could only be assured based on usability tests with real users

Malhas Poligonais Multiresolução

Frutuoso G. M. Silva

IT – Grupo de Redes e Multimedia

Departamento de Informática, Universidade da Beira Interior

Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã

fsilva@di.ubi.pt

Resumo

Este artigo apresenta o estado da arte sobre a análise multiresolução na área da Computação Gráfica e descreve um esquema multiresolução desenvolvido para malhas poligonais. O esquema multiresolução baseia-se num novo algoritmo de simplificação de malhas (NSA) e num algoritmo de refinamento de malhas. Assim, apresenta-se o algoritmo de simplificação de malhas NSA e os principais resultados obtidos. O algoritmo NSA consegue reduzir em cerca de 50% o número de faces de um modelo poligonal numa única simplificação, mantendo a forma global do modelo e a sua fronteira. Este algoritmo de simplificação é o mais rápido dentro da sua classe.

Palavras chave

Análise multiresolução, simplificação de malhas, malhas multiresolução.

1. INTRODUÇÃO

A complexidade dos modelos tridimensionais tem aumentado bastante devido principalmente às tecnologias de aquisição de dados tridimensionais como os *scanners* 3D. Este tipo de tecnologias produzem hoje modelos poligonais a partir de objectos reais, normalmente designados por malhas poligonais que são usadas nas mais variadas aplicações como, por exemplo, na visualização, nos ambientes virtuais e nos jogos de computador. De facto, muitos trabalhos de investigação têm surgido nos últimos anos no que se refere a malhas poligonais, particularmente na edição de malhas [Kobbelt99, Suzuki00, Sorkine04], na compressão geométrica e transmissão pela Internet [Taubin98, Touma98, Karni00], e na análise multiresolução aplicada a malhas [Hoppe96, Zorin97, Garland99, Kobbelt00, Botsch03].

No entanto, as malhas geradas pelos *scanners* 3D possuem muitas vezes informação redundante devido à elevada resolução destes dispositivos, a qual é aplicada de forma uniforme e independentemente das características de forma do objecto.

A manipulação eficiente de malhas com grande número de células é conseguida através do uso de múltiplas representações com diferente resolução para cada objecto geométrico. Isto é, usando modelos com diferentes níveis de detalhe, ou seja, modelos LOD (*Level-Of-Detail*). No entanto, a análise multiresolução veio introduzir uma representação alternativa aos modelos LOD. A análise multiresolução permite representar um modelo com diferentes níveis de detalhe

com apenas uma única malha em memória em cada instante. Depois, com o auxílio de algoritmos de simplificação e refinamento faz-se a passagem de um nível de detalhe para o seguinte e para o anterior, consoante as necessidades da aplicação.

Este artigo faz um resumo da análise multiresolução na área da Computação Gráfica e apresenta um esquema multiresolução para malhas poligonais. Para isso foi desenvolvido um novo algoritmo de simplificação de malhas, designado de *Normal-based Simplification Algorithm* (NSA).

Este artigo está organizado em várias secções: A secção 2 apresenta as diferentes abordagens da análise multiresolução na Computação Gráfica. A secção 3 apresenta um esquema multiresolução para malhas. A secção 4 apresenta o algoritmo NSA de simplificação de malhas desenvolvido e que serve de base ao esquema multiresolução apresentado. Na secção 5 apresenta-se alguns resultados sobre malhas multiresolução e sobre o algoritmo NSA. Por último, na secção 6 apresenta-se algumas conclusões e o trabalho futuro.

2. A MULTIRESOLUÇÃO NA COMPUTAÇÃO GRÁFICA

A análise multiresolução está inevitavelmente associada às onduletas (*wavelets*) na Matemática e no processamento de imagem.

A Tabela 1 apresenta uma classificação das diferentes abordagens da análise multiresolução na Computação Gráfica. Esta classificação baseia-se na representação hierárquica da granularidade dos modelos. Por exemplo,

num algoritmo de subdivisão é gerado um modelo mais detalhado de um objecto geométrico do que o modelo original. Ao invés, no caso dos algoritmos baseados em ondulas e dos algoritmos de simplificação está associada uma granularidade mais grosseira ao modelo gerado do que a do objecto original. As diferentes abordagens são apresentadas de seguida.

Abordagens	Classificação
Ondulas	Detalhado → Grosseiro
Algoritmos de Subdivisão	Grosseiro → Detalhado
Algoritmos de Simplificação	Detalhado → Grosseiro

Tabela 1: Classificação das diferentes abordagens da multiresolução na Computação Gráfica.

2.1. Ondulas

As ondulas são uma ferramenta matemática da análise multiresolução. As ondulas permitem decompor hierarquicamente uma função em duas partes: uma função aproximada mais simples (ou com menos resolução) e os coeficientes de detalhe necessários para recuperar a função original.

Em Computação Gráfica, pode usar-se também a teoria das ondulas para simplificar malhas, através da aplicação de uma transformação **A** que cria uma malha grosseira e ao mesmo tempo aplicando outra transformação **B** para captar o detalhe que é perdido devido à primeira transformação (Figura 1). Mais informações sobre ondulas em Computação Gráfica podem ser encontradas em [Fournier95, Stollnitz96].

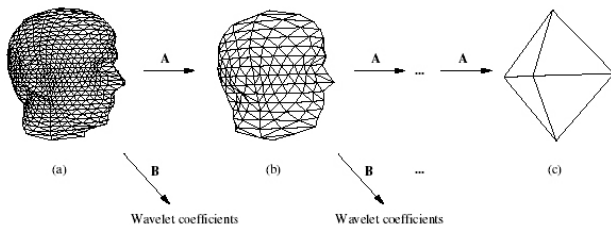


Figura 1: Decomposição de uma malha poligonal (retirada de [Lounsbery97]).

2.2. Algoritmos de Subdivisão

Em Computação Gráfica, os algoritmos de subdivisão são usados para criar superfícies por subdivisão ou malhas poligonais com diversos níveis de detalhe. A partir de um modelo grosseiro gera-se um modelo mais detalhado por subdivisão sucessiva das células da malha, como ilustra a Figura 2 para o caso de uma malha triangular.

Na Figura 2 vê-se a malha inicial M_0 e dois passos do algoritmo de subdivisão que geram M_1 e M_2 , respectivamente. Claro que a subdivisão obedece a regras precisas que permitem distinguir um esquema de subdivisão de outro.

As superfícies obtidas por subdivisão têm importantes propriedades de modelação: suportam malhas com topologias arbitrárias, garantem uma certa continuidade da superfície (continuidade C^0) e permitem o

desenvolvimento de algoritmos eficientes que são simples de codificar.

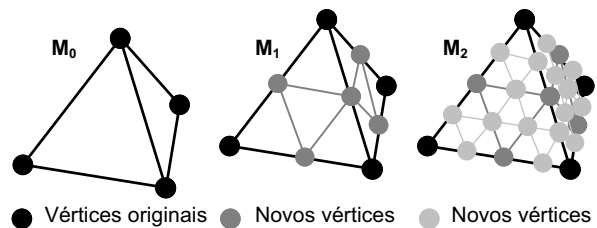


Figura 2: Subdivisão de uma malha poligonal.

A subdivisão permite definir uma curva ou superfície como o limite de uma sequência de refinamentos (subdivisões) sucessivos de um modelo inicial grosseiro. Existem vários esquemas de subdivisão para superfícies (e também para curvas). Os primeiros esquemas de subdivisão para malhas irregulares foram os esquemas de Catmull-Clark [Catmull78] e de Doo-Sabin [Doo78]. Mais tarde surgiu o esquema de Loop [Loop87] para malhas triangulares arbitrárias. Dyn *et al.* [Dyn90] apresentaram um novo esquema designado de *butterfly* para interpolar superfícies obtidas por subdivisão triangulares com continuidade C^1 . Kobbelt [Kobbelt96] apresentou também um esquema com continuidade C^1 mas para malhas quadrangulares com topologia arbitrária. Note-se que todos estes esquemas permitem criar diversas aproximações à malha. A diferença entre eles reside no cálculo da geometria dos novos vértices. Esquemas alternativos têm surgido na literatura e estão descritos em [Zorin00, Velho01, Warren02].

A Figura 3 evidencia as diferenças práticas entre os diversos esquemas de subdivisão referenciados anteriormente. Como se pode constatar, o resultado de cada esquema de subdivisão faz-se notar na forma final do objecto, podendo este ser maior ou menor e mais ou menos suave.

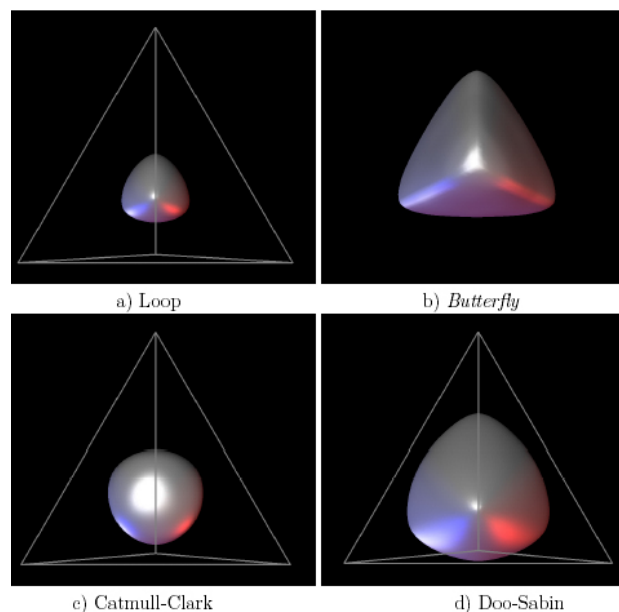


Figura 3: Comparação dos esquemas de subdivisão apresentados para um tetraedro (retirada de [Zorin00]).

Um resumo mais alargado dos esquemas de subdivisão e dos últimos desenvolvimentos na área pode ser consultado em [Sabin04].

2.3. Algoritmos de Simplificação

Os algoritmos de simplificação são outra das formas usadas na criação de representações multiresolução, ou seja, representações com uma estrutura hierárquica de modelos com diversos níveis de detalhe. Neste caso parte-se do modelo detalhado para obter um modelo mais grosseiro após vários passos de simplificação.

Simplificar uma malha poligonal M_i consiste na geração de outra malha M_j com menor número de células, onde a malha M_j obedece a um critério que estabelece um erro máximo admissível. Em alguns casos, o erro é imposto implicitamente pelo número de células da malha M_j .

Os algoritmos de simplificação podem ser classificados em três categorias, consoante o tipo de operação usada para a simplificação da malha. Assim, temos as seguintes categorias:

Remoção de células (*Cell decimation*).

Basicamente, estes algoritmos seleccionam uma célula (vértice, aresta ou face) para remover, após o que removem também todas as suas células adjacentes, o que implica a reconstrução da malha na zona de onde foram removidas as células. Embora estes algoritmos reduzam o número de células da malha, a topologia do modelo é preservada. A sua principal vantagem é que os vértices que permanecem na malha pertencem à malha original, o que de alguma forma permite ter a informação da malha original. Este facto permite ainda a utilização dos atributos (cor, textura, etc.) dos vértices que se mantêm inalterados.

Confluência de vértices (*Vertex clustering*).

Os algoritmos desta classe utilizam um volume que envolve a malha, que depois é subdividido em pequenos sub-volumes. A subdivisão do volume envolvente pode ser do tipo *octree*. Na subdivisão *octree*, um volume é recursiva e hierarquicamente subdividido em oito sub-volumes. Os vértices que se encontrarem dentro de um mesmo sub-volume passam a ser representados apenas pelo vértice representativo (ou *cluster*), sendo então criada uma nova malha com base apenas nos vértices representativos. O vértice representativo pode ser o centro de massa do sub-volume ou outro ponto qualquer do sub-volume escolhido por um critério pré-definido. Normalmente, estes algoritmos produzem aproximações muito grosseiras do modelo porque não preservam nem a topologia nem a geometria do modelo, e a qualidade das malhas depende sempre da dimensão do sub-volume criado.

Contracção de arestas (*Edge collapsing*).

Estes algoritmos simplificam a malha iterativamente pela contracção de arestas em vértices. A escolha da aresta a contrair é uma das características intrínsecas de cada algoritmo que determina como as simplificações são geradas. Estes algoritmos mantêm a topologia do modelo e os vértices do modelo simplificado também podem pertencer ao modelo original. A operação de

contracção de arestas tem como desvantagem poder causar localmente auto-intersecções ou inflexões na malha, pelo que se torna necessário identificar cuidadosamente estas situações. O algoritmo NSA descrito na Secção 4 pertence a esta categoria.

3. MALHAS MULTIRESOLUÇÃO

Uma malha multiresolução permite representar um modelo para diferentes níveis de detalhe mas com apenas uma única malha em memória em cada instante. Assim, a passagem de um nível de detalhe para o seguinte e para o anterior faz-se por aplicação de algoritmos de simplificação e refinamento.

Neste caso, o esquema multiresolução desenvolvido para malhas baseia-se no algoritmo de simplificação NSA que é descrito na secção seguinte.

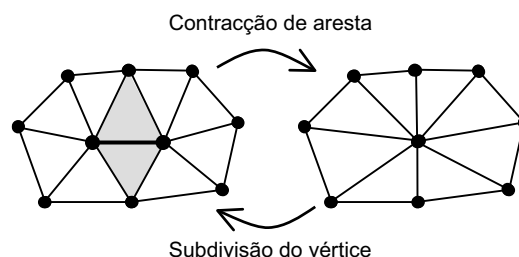


Figura 4: Esquema multiresolução.

O algoritmo NSA é baseado na operação de contracção de arestas enquanto que o algoritmo de refinamento é baseado na operação inversa, a subdivisão de vértices. O refinamento da malha é efectuado recorrendo a um algoritmo similar ao proposto por Hoppe para as malhas progressivas [Hoppe96]. A Figura 4 apresenta o esquema multiresolução desenvolvido com base na contracção de arestas e na subdivisão de vértices.

A estrutura de dados utilizada no desenvolvimento das malhas multiresolução e dos respectivos algoritmos de simplificação e refinamento, foi a estrutura de dados AIF. A estrutura de dados AIF permite o acesso rápido à informação topológica sobre adjacências e incidências. Além disso, a estrutura de dados AIF é uma estrutura de dados B-rep que permite representar malhas poligonais genéricas, ou seja, malhas não necessariamente triangulares nem *manifold*. A estrutura de dados AIF permite representar malhas bidimensionais onde:

- um *vértice* é definido em termos das arestas que lhe são incidentes;
- uma *aresta* é definida em termos dos seus vértices adjacentes e das faces que lhe são incidentes;
- uma *face* é definida em termos das suas arestas adjacentes.

Mais detalhes sobre a estrutura de dados AIF podem ser vistos em [Silva04a].

4. ALGORITMO DE SIMPLIFICAÇÃO – NSA

A simplificação de uma malha é feita por contracção de arestas em vértices. No caso do algoritmo NSA o vértice escolhido para a contracção da aresta é o vértice correspondente ao seu ponto médio. No entanto, nada impede que seja outro o ponto escolhido.

Este algoritmo tem como particularidade usar o mesmo critério para a escolha e validação da aresta a contrair, contrariamente aos outros algoritmos da sua classe.

No algoritmo NSA a escolha da aresta a contrair baseia-se na variação das normais às faces em redor da aresta. Assim, a contracção de uma aresta no algoritmo NSA só acontece quando a variação das normais às faces em redor da aresta a contrair estiver contida numa dada tolerância ϵ . O valor de ϵ define a variação admissível para o ângulo entre as normais às faces antes e depois da contracção da aresta. Este valor é especificado pelo utilizador num intervalo pré-definido. Quanto maior for o valor de ϵ maior será a simplificação da malha. Além disso, para um mesmo valor de ϵ o algoritmo cria sempre um número finito de simplificações. Isto é, existe uma simplificação a partir do qual já não é possível simplificar mais a malha para o mesmo valor de ϵ .

Este é um critério geométrico que requer que a região à volta da aresta a contrair seja aproximadamente complanar. Mas o contrário já não é verdade, ou seja, a complanaridade não assegura que exista uma variação mínima das normais às faces. Por exemplo, a Figura 5 mostra dois casos de faces complanares, mas só as faces da Figura 5(a) respeitam o critério visto que temos duas faces com a mesma orientação, enquanto que na Figura 5(b) temos duas faces com orientações opostas, apesar de complanares.

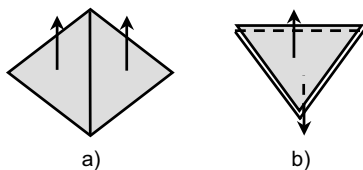


Figura 5: Faces complanares.

Note-se que o algoritmo NSA requer a quase-complanaridade da região a simplificar, o que acaba por validar simultaneamente as operações de escolha e contracção de arestas, pois a operação de contracção só é permitida quando a variação das normais às faces for menor que um dado ϵ .

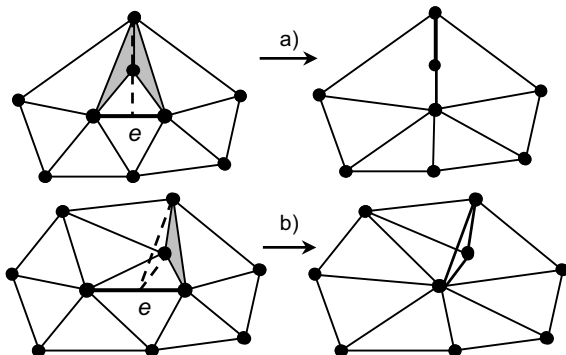


Figura 6: Validação da operação de contracção de arestas.

De facto, a operação de contracção de arestas pode causar inversões locais na superfície da malha, a menos que haja uma forma de controlar a sobreposição de faces (Figura 6(a)) ou a criação de concavidades (Figura 6(b)) na malha, que podem aparecer depois da contracção de uma aresta.

Por exemplo, na Figura 6(a) a contracção da aresta e provoca a sobreposição das faces sombreadas como se indica pelo segmento a traço interrompido. Similarmente, a face sombreada na Figura 6(b) passa a estar na posição indicada pela face a traço interrompido, o que significa que muda de orientação depois da contracção da aresta e . Em ambos os casos o algoritmo NSA não permite a contracção da aresta e porque a variação das normais seria maior do que ϵ , e assim sendo faz simultaneamente a validação da operação de contracção de arestas.

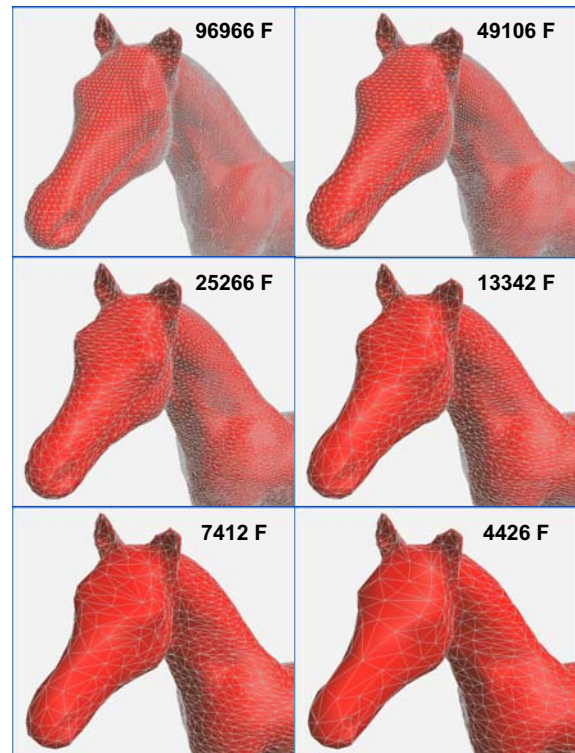


Figura 7: Malha de um cavalo (original e cinco simplificações).

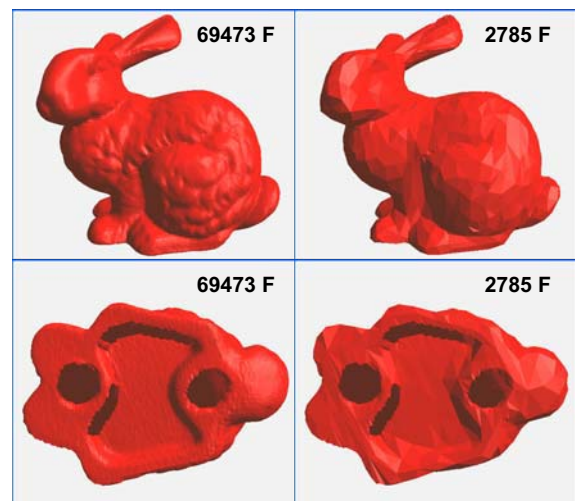


Figura 8: Malha de um coelho (original e a última simplificação).

O algoritmo NSA consegue reduzir o número de faces de uma malha até cerca de 50%, numa única simplificação. Por exemplo, na Figura 7 pode ver-se a

malha de um cavalo com 96966 faces (na malha original) e cinco simplificações criadas com o algoritmo NSA. Neste caso, para a primeira simplificação a redução do número de faces atingiu os 49,35%.

Como se pode ver também na Figura 7, o algoritmo NSA mantém a forma global do objecto onde apenas alguns detalhes de forma se vão perdendo em cada simplificação. O algoritmo NSA mantém ainda a fronteira da malha pois não é permitida a contracção de arestas da fronteira, ou de arestas que possam causar alterações na fronteira. Como se pode constatar na Figura 8 para o caso de um coelho a sua fronteira mantém-se inalterada durante todas as simplificações.

Além disso, os teste comparativos entre o algoritmo NSA e Qslim [Garland97] mostraram que é o algoritmo mais rápido dentro da sua classe. Pode dizer-se que isto se deve principalmente a três factores: (i) a usar um critério meramente geométrico e simples de calcular, (ii) de se contrair as arestas sempre no seu ponto médio, evitando cálculos adicionais, e (iii) devido à estrutura de dados AIF de suporte que permite o acesso rápido à informação topológica. Note-se que o algoritmo Qslim é de acordo com a literatura [Cignoni98] o mais rápido.

5. RESULTADOS

O hardware de suporte ao desenvolvimento das malhas multiresolução foi um PC equipado com um processador Pentium 4, com 768MB de memória, uma placa gráfica GeForce 4 com 64MB, e o sistema operativo Windows 2000.

O desempenho dos algoritmos de simplificação e refinamento de malhas depende sempre da dimensão da malha, ou seja, quantas mais células a malha contiver maior é o tempo necessário à sua simplificação e ao seu refinamento. Por exemplo, a Figura 9 apresenta os tempos de simplificação e refinamento para a malha do coelho, ou seja, apresenta os tempos necessários à passagem de um nível de detalhe para o seguinte e para o anterior. Como se pode ver os tempos de refinamento são sempre inferiores aos tempos de simplificação o que advém do facto de o algoritmo de refinamento saber *a priori* quais são os vértices a subdividir, enquanto que o algoritmo de simplificação tem de escolher as aresta a contrair de acordo com o critério estabelecido. A Figura 10 compara os tempos de simplificação para a malha do coelho entre os algoritmos Qslim e NSA. Como se pode ver o algoritmo NSA é o mais rápido especialmente nas primeiras simplificações, sendo a diferença já pouco significativa para as últimas simplificações.

Quanto à qualidade das malhas geradas pelo algoritmo NSA pode ver-se na Figura 11 por comparação com o algoritmo Qslim para a malha do coelho. De facto, o algoritmo NSA para as últimas simplificações gera malhas de menor qualidade em relação às do algoritmo Qslim. Mais detalhes sobre o algoritmo NSA podem ser vistos em [Silva04], incluindo mais resultados comparativos sobre tempos e qualidade das malhas geradas.

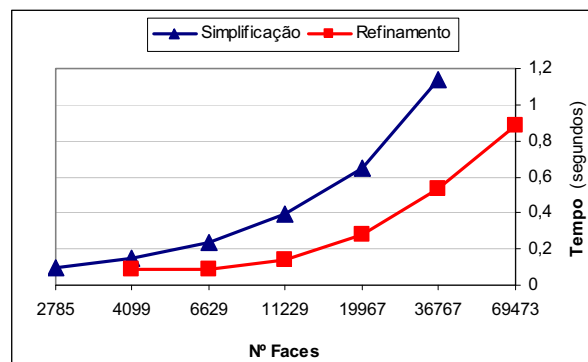


Figura 9: Tempos de simplificação e refinamento para a malha do coelho.

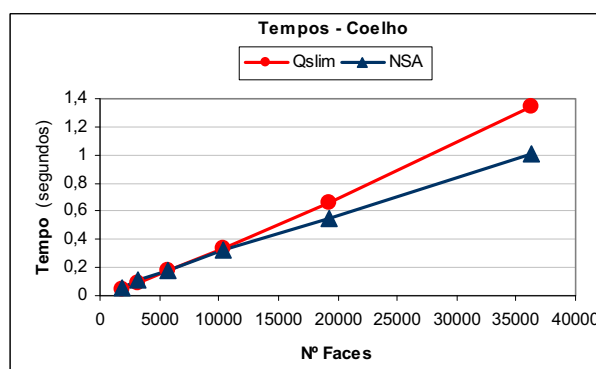


Figura 10: Tempos de simplificação para os algoritmos Qslim e NSA para a malha do coelho.

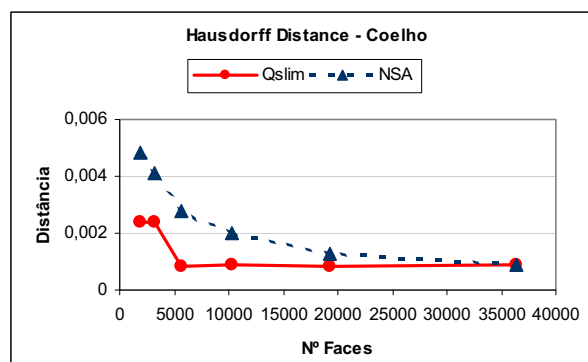


Figura 11: Qualidade da malha para os algoritmos Qslim e NSA para a malha do coelho.

6. CONCLUSÃO E TRABALHO FUTURO

Este artigo faz um resumo da análise multiresolução na área da Computação Gráfica. Neste contexto, apresentou-se um esquema multiresolução para malhas poligonais baseado num novo algoritmo de simplificação de malhas (NSA). Note-se que o algoritmo NSA de simplificação de malhas consegue reduzir a cerca de metade o número de faces de uma malha numa única simplificação, mantendo a forma global do objecto e a sua fronteira. Além disso, é o algoritmo mais rápido dentro da sua classe.

O desempenho dos algoritmos de simplificação e refinamento ainda limita a utilização de modelos multiresolução para alguns tipos de aplicações. No entanto, estes modelos já são hoje usados em algumas

áreas com sucesso como, por exemplo, na compressão e transmissão de modelos geométricos pela Internet, na visualização dependente do contexto e na edição interactiva de malhas. No futuro os modelos multiresolução poderão vir mesmo a substituir os modelos LOD. Por exemplo, quando pudermos ter os algoritmos de simplificação e refinamento codificados em hardware. Isto porque, os modelos LOD necessitam de armazenar diversas representações de um mesmo objecto em memória, contrariamente aos modelos multiresolução que mantêm apenas uma única representação em memória.

No futuro espera-se desenvolver um protótipo de um sistema de animação que possa integrar o esquema multiresolução descrito neste artigo. Assim, os objectos do sistema de animação terão uma representação multiresolução o que permitirá representá-los para diferentes níveis de detalhe utilizando algoritmos de simplificação e refinamento.

AGRADECIMENTOS

Os malhas utilizadas neste trabalho são cortesia de: Cyberware - Cavallo e Stanford University - Coelho.

REFERÊNCIAS

- [Botsch03] Mario Botsch and Leif Kobbelt. Multiresolution Surface Representation Based on Displacement Volumes. *In Proceedings of Eurographics*, pp.483-491, 2003.
- [Catmull78] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, Vol 10, pp.350-355, 1978.
- [Cignoni98] P. Cignoni, C. Montani and R. Scopigno. A comparison of mesh simplification algorithms. *Computers and Graphics*, Vol 22, no.1, pp.37-54, 1998.
- [Doo78] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer Aided Design*, Vol 10, no.6, pp.356-360, 1978.
- [Dyn90] Nira Dyn, David Levin and John A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, Vol 9, no.2, pp.160-169, 1990.
- [Fournier95] Alain Fournier. Wavelets and their Applications in Computer Graphics. *In Siggraph Course Notes*, 1995.
- [Garland97] Michael Garland and Paul S. Heckbert. Surface Simplification Using Quadric Error Metrics. *In Proceeding of Siggraph*, pp.209-216, 1997.
- [Garland99] Michael Garland. Multiresolution modeling: Survey & future opportunities. *In Eurographics - State of the Art Reports*, pp.111-131, 1999.
- [Hoppe96] Hugues Hoppe. Progressive Meshes. *In Proceedings of Siggraph*, pp.99-108, 1996.
- [Karni00] Zachi Karni and Craig Gotsman. Spectral Coding of Mesh Geometry. *In Proceedings of Siggraph*, pp.279-286, 2000.
- [Kobbelt96] Leif Kobbelt. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. *Computer Graphics Forum*, Vol 15, no.3, pp.409-420, 1996.
- [Kobbelt99] Leif Kobbelt, Jens Vorsatz and Hans-Peter Seidel. Multiresolution hierarchies on unstructured triangle meshes. *Computational Geometry Journal: Theory and Applications*, Vol 14, pp.5-24, 1999.
- [Kobbelt00] Leif Kobbelt, Thilo Bareuther and Hans-Peter Seidel. Multiresolution Shape Deformations for Meshes with Dynamic Vertex Connectivity. *In Proceedings of Eurographics*, pp.C249-C260, 2000.
- [Loop87] Charles Loop. *Smooth Subdivision Surfaces Based on Triangles*. Master's Thesis, Utah University, 1987.
- [Lounsbery97] Michael Lounsbery, Tony D. DeRose and Joe Warren. Multiresolution Analysis for Surfaces of Arbitrary Topological Type. *ACM Transactions on Graphics*, Vol 16, no.1, pp.34-73, 1997.
- [Sabin04] Malcolm Sabin. Recent Progress in Subdivision: a Survey. *In Advanced in Multiresolution for Geometric Modelling*, Dodgson, Floater, Sabin (Eds.), Springer, pp.203-230, 2004.
- [Silva04] Frutuoso G. M. Silva and Abel J. P. Gomes. Normal-based simplification algorithm for meshes. *In Proceedings of Theory and Practice of Computer Graphics (TPCG'04)*, pp.211-218, 2004.
- [Silva04a] Frutuoso G. M. Silva and Abel J. P. Gomes. A B-rep data structure for polygonal meshes. *VIRTUAL - The Portuguese Journal of Computer Graphics*, Special edition - Advances in Computer Graphics in Portugal, Adérito Marcos and Miguel Salles Dias (eds.), Novembro, 2004.
- [Sorkine04] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl and Hans-Peter Seidel. Laplacian Surface Editing. *In Proceedings of Eurographics Symposium on Geometry Processing*, pp.179-188, 2004.
- [Stollnitz96] Eric J. Stollnitz, Tony D. DeRose, David H. Salesin and Anthony D. DeRose. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann, 1996.
- [Suzuki00] Hiromasa Suzuki, Yusuke Sakurai, Takashi Kanai and Fumihiko Kimura. Interactive mesh dragging with an adaptive remeshing technique. *The Visual Computer*, Vol.16, no.3/4, pp.159-176, 2000.
- [Taubin98] G. Taubin and J. Rossignac. Geometric compression through topological surgery. *ACM Transactions on Graphics*, Vol.17, no.2, pp.84-115, 1998.
- [Touma98] Costa Touma and Craig Gotsman. Triangle Mesh Compression. *In Proceedings of Graphics Interface*, pp.26-34, 1998.
- [Velho01] Luis Velho. Quasi 4-8 Subdivision. *Computer-Aided Geometric Design*, 2001.
- [Warren02] Joe Warren and Henrik Weimer. *Subdivision Methods For Geometric Design - A constructive approach*, Morgan Kaufman, 2002.
- [Zorin97] D. Zorin, P. Schröder and W. Sweldens. Interactive Multiresolution Mesh Editing. *In Proceedings of Siggraph*, pp.259-268, 1997.
- [Zorin00] D. Zorin and P. Schröder. Subdivision for modeling and animation. *In Siggraph Course Notes*, 2000.

Improved Line/Edge Detection and Visual Reconstruction

João Rodrigues
Escola Superior de Tecnologia

University of Algarve - Faro, Portugal
{jrodrig, dubuf}@ualg.pt

J.M.H. du Buf
Vision Laboratory - FCT

Abstract

Lines and edges provide important information for object categorization and recognition. In addition, one brightness model is based on a symbolic interpretation of the cortical multi-scale line/edge representation. In this paper we present an improved scheme for line/edge extraction from simple and complex cells and we illustrate the multi-scale representation. This representation can be used for visual reconstruction, but also for non-photorealistic rendering. Together with keypoints and a new model of disparity estimation, a 3D wireframe representation of e.g. faces can be obtained in the future.

Keywords

Visual cortex; line and edge detection; multi-scale; visual reconstruction; non-photorealistic rendering.

1. INTRODUCTION

Computer graphics, for example (non-)photorealistic rendering (NPR), is closely related to visual perception. Trying to understand perception requires insight into low-level processes in the eyes and visual cortex. In addition, artistic rendering in the style of a certain painter requires insight into the painter's techniques (brushes, strokes, color palettes) and high-level cognitive processes [Zeki00, Livingstone00]. The latter relate to image interpretation and certain effects, for example illusions like brightness and color induction, i.e. the opposite effects of simultaneous contrast (a high surround brightness pushes the center towards a lower level) and assimilation (a high surround may also pull up the center brightness), see [du Buf95].

Because such opposite center-surround effects are not yet well understood, there is a need to develop e.g. brightness models from which we can learn. Such models must be based on state-of-the-art models of visual representations in the primary cortex: area V1 and beyond.

In this paper we present an improved model for line and edge detection, suitable for modeling brightness and NPR, concentrating on the multi-scale representation, curvature and visual reconstruction. This model is based on simple and complex cells in V1, and complements existing models for end-stopped cells [Heitger92], keypoints [Wütz00, Rodrigues04b], saliency for focus-of-attention [Deco04, Rodrigues05a], inhibition [Grigorescu03, Petkov93] and disparity [Fleet91, Rodrigues04a]. Together, these models provide input for object and face detection [Rodrigues05a, Rodrigues05b] if embedded into an architecture with ventral and dorsal data streams: the where and what systems [Deco04, Rensink00].

2. CELL MODELS AND NCRF INHIBITION

Gabor quadrature filters provide a model of cortical simple cells [Lee96]. In the spatial domain (x, y) they consist of a real cosine and an imaginary sine, both with a Gaussian envelope. A receptive field (RF) is denoted by (see e.g. [Grigorescu03]):

$$G_{\lambda, \sigma, \theta, \varphi}(x, y) = \exp\left(-\frac{\tilde{x}^2 + \tilde{y}^2}{2\sigma^2}\right) \cos\left(2\pi \frac{\tilde{x}}{\lambda} + \varphi\right), \quad (1)$$

$\tilde{x} = x \cos \theta + y \sin \theta$; $\tilde{y} = x \sin \theta - y \cos \theta$, where the aspect ratio $\gamma = 0.5$ and σ determines the size of the RF. The spatial frequency is $1/\lambda$, λ being the wavelength. For the bandwidth σ/λ we use 0.56, which yields a half-response width of one octave. The angle θ determines the orientation (we use 8 orientations), and φ the phase symmetry (0 or $\pi/2$). We can apply a linear scaling between $f_{\min}$ and $f_{\max}$ with a few discrete scales or hundreds of contiguous scales.

The responses of even (Fig. 1(A)) and odd (Fig. 1(B)) simple cells, which correspond to the real and imaginary parts of a Gabor filter, are obtained by convolving the input image with the RF, and are denoted by $R_{s,i}^E$ and $R_{s,i}^O$, s being the scale, i the orientation ($\theta_i = i\pi/(N_\theta - 1)$) and N_θ the number of orientations. The responses of complex cells (Fig. 1(C)) are modelled by the modulus

$$C_{s,i}(x, y) = \left[\left\{ R_{s,i}^E(x, y) \right\}^2 + \left\{ R_{s,i}^O(x, y) \right\}^2 \right]^{\frac{1}{2}}. \quad (2)$$

Non-classical receptive field (NCRF) inhibition (Fig. 1(D)) can be used to suppress information in textured regions [Grigorescu03]. There are two types: (a) ani-

sotropic, in which only responses obtained for the same preferred RF orientation contribute to the suppression, and (b) isotropic, in which all responses over all orientations equally contribute to the suppression.

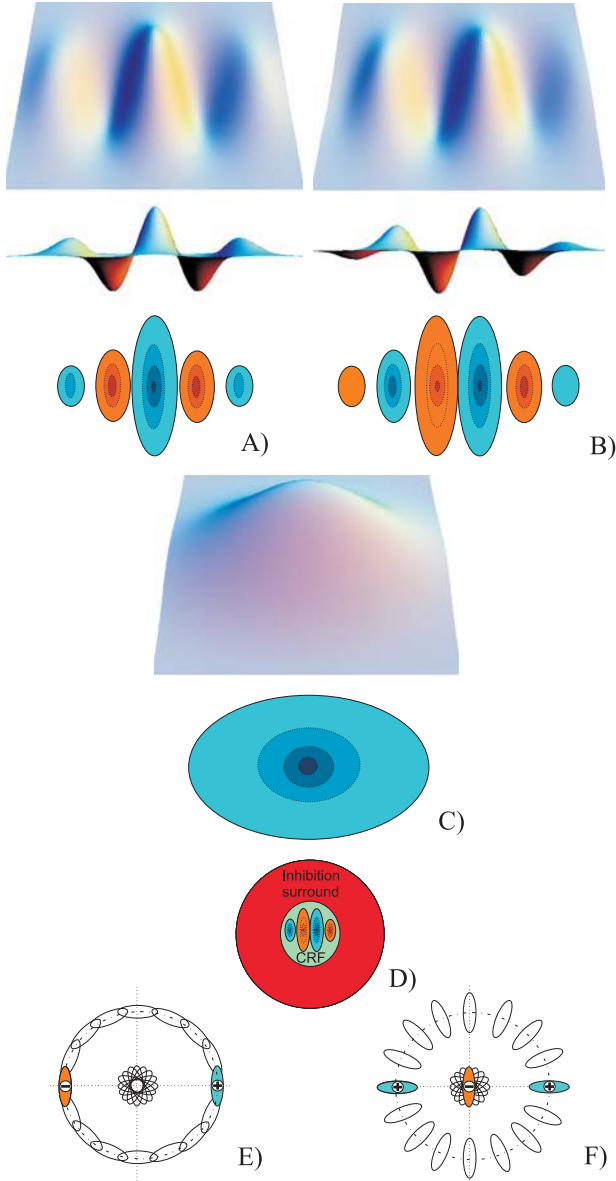


Figure 1: 3D and 2D cell representations (see text).

The anisotropic non-classical receptive field inhibition (A-NCRF) model is computed by an inhibition term $t_{s,\sigma,i}^A$ for each orientation i ,

$$t_{s,\sigma,i}^A = C_{s,i} * w_{\sigma}, \quad (3)$$

as a convolution of the complex cell response $C_{s,i}$ with the weighting function w_{σ} , with $\|\cdot\|_1$ being the L_1 norm and $[\cdot]^+$ denotes the suppression of negative values:

$$w_{\sigma}(x, y) = [DoG_{\sigma}(x, y)]^+ / \|[DoG_{\sigma}]^+\|_1, \quad (4)$$

$$DoG_{\sigma}(x, y) = \frac{1}{2\pi(4\sigma)^2} \exp\left(-\frac{x^2 + y^2}{2(4\sigma)^2}\right) - \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (5)$$

The operator $b_{s,\sigma,i}^A$ corresponds to the inhibition of $C_{s,i}$, i.e. $b_{s,\sigma,i}^A = [C_{s,i} - \alpha t_{s,\sigma,i}^A]^+$, with α controlling the strength of the inhibition.

The isotropic NCRF (I-NCRF) model is obtained by computing the inhibition term $t_{s,\sigma}^I$ which does not depend on orientation i . For this we construct the maximum response map of the complex cells $\tilde{C}_s = \max\{C_{s,i}\}$, with $i = 0, \dots, N_{\theta} - 1$. The isotropic inhibition term $t_{s,\sigma}^I$ is computed as a convolution of the maximum response map $\tilde{C}_s$ with the weighting function w_{σ} , and the isotropic operator is $b_{s,\sigma,i}^I = [C_s - \alpha t_{s,\sigma}^I]^+$.

3. LINE/EDGE DETECTION METHOD

Van Deemter and du Buf [Deemter96] presented a scheme for line and edge detection based on the responses of simple cells. A positive line is detected where R^E shows a local maximum in the orthogonal filter orientation and R^O shows a zero crossing. In the case of an edge the even and odd responses must be swapped. This gives 4 possibilities for positive and negative events: local maxima/minima plus zero crossings. Rodrigues and du Buf [Rodrigues04b] combined the responses of simple and complex cells, i.e. simple cells serve to detect positions and event types, whereas complex cells are used to increase the confidence. Since the use of Gabor modulus (complex cells) implies some loss of precision at vertices [du Buf93], increased precision was obtained by considering multiple scales.

The two algorithms described above work reasonably well but there are still a few problems: (a) either one scale is used or only a very few scales for increasing confidence, (b) some parameters must be optimized for specific input images or even as a function of scale, (c) detection precision can be improved, and (d) detection continuity at highly curved lines/edges must be guaranteed.

We present an improved algorithm with no free parameters, truly multi-scale, and with new solutions for problems (c) and (d). With respect to precision, simple and complex cells respond beyond line and edge ends, for example beyond the corners of a rectangle. In addition, at line or edge crossings, detection leads to continuity of the dominant events and gaps in the sub-dominant events. These gaps must be reduced in order to reconstruct continuity. Both problems can be solved by introducing new inhibition schemes, like the radial and tangential ones used in the case of keypoint operators [Rodrigues04b].

Here we use lateral (L) and cross-orientation (C) inhibition, see Fig. 1(E,F), defined as:

$$I_{s,i}^L(x,y) = \left[C_{s,i}(x + d\dot{C}_{s,i}, y + d\dot{S}_{s,i}) - C_{s,i}(x - d\dot{C}_{s,i}, y - d\dot{S}_{s,i}) \right]^+ + \left[C_{s,i}(x - d\dot{C}_{s,i}, y - d\dot{S}_{s,i}) - C_{s,i}(x + d\dot{C}_{s,i}, y + d\dot{S}_{s,i}) \right]^+; \quad (6)$$

$$I_{s,i}^E(x,y) = \left[C_{s,(i+N_\theta/2)}(x + 2d\dot{C}_{s,i}, y + 2d\dot{S}_{s,i}) - 2C_{s,i}(x,y) + C_{s,(i+N_\theta/2)}(x - 2d\dot{C}_{s,i}, y - 2d\dot{S}_{s,i}) \right]^+, \quad (7)$$

where $(i + N_\theta/2) \perp i$, with $\dot{C}_{s,i} = \cos \theta_{s,i}$, $\dot{S}_{s,i} = \sin \theta_{s,i}$ and $d = 0.6s$. We can apply the inhibition to the complex cell responses, with β controls the strength of the inhibition (normally we use $\beta = 1.0$),

$$\hat{C}_{s,i} = \left[C_{s,i}(x,y) - \beta(I_{s,i}^L(x,y) + I_{s,i}^E(x,y)) \right]^+. \quad (8)$$

Line/edge (event) detection is achieved by constructing a few cell layers on top of simple and complex cells.

The first layer serves to select active regions and dominant orientations. At each position responses of complex cells are summed

$$\hat{C}_s = \sum_{i=0}^{N_\theta-1} \hat{C}_{s,i}. \quad (9)$$

and only at positions with $\hat{C}_s$ not equal to zero an output cell is activated. At active output cells, the dominant orientation is selected by gating one complex cell on the basis of non-maximum suppression of $\hat{C}_{s,i}$. The gating is confirmed or corrected by an excitation/inhibition process of dominant orientations in a local neighborhood.

In the second layer, event type and position is determined on the basis of active output cells (1st layer) and gated simple and complex cells. A first cell complex checks simple cells $R_{s,i}^E$ and $R_{s,i}^O$ for a local maximum (or minimum by rectification) using a dendritic field size of $\pm\lambda/4$, λ being the wavelength of the simple cells (Gabor filter).

The active output cell is inhibited if there is no maximum or minimum. A second cell complex does exactly the same on the basis of complex cells. A third cell complex gates four types of zero-crossing cells on the basis of simple cells, again on $\pm\lambda/4$. If there is no zero-crossing, the output cell is inhibited. If there is a zero-crossing, the active cell at the position of the zero-crossing cell determines event position and the active zero-crossing cell determines event type.

In the third layer, the small loss of accuracy due to the use of complex cells in the second layer is compensated. This is done by correcting local event continuity, considering the information available in the second layer, but by using excitation of output cells by means of grouping cells that combine simple and complex cells tuned to the same and adjacent orientations, see Fig. 2. The latter

process is an extension of linear grouping [Deemter96] and a simplification of using banana wavelets [Krüger97]. In the same layer event type is corrected in small neighborhoods, restoring type continuity, because the cell responses may be distorted by interference effects when two events are very close [du Buf93].

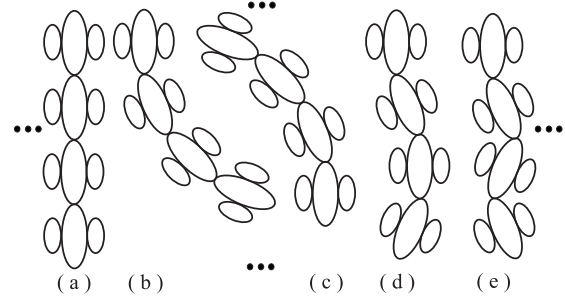


Figure 2: Curvature continuity.

Figure 3 shows four input images together with detection results (positive and negative lines and edges are coded by different gray levels). Detection accuracy is very good and there are many small events due to low-contrast textures (Fig. 3 does not show event amplitudes!) and the fact that there is no threshold value in the detection scheme.

Figure 4 shows results obtained by standard edge-only (!) detection algorithms from computer vision in the case of the orange image with, from left to right: Canny, Nalwa, Bergholm and Iverson [Heath97] (for more results see http://marathon.csee.usf.edu/edge/edge_detection.html).

Most events in textured regions can be suppressed by NCRF inhibition. Figure 5 shows results obtained by I-NCRF at the finest filter scale. For more results obtained with NCRF we refer to [Grigorescu03], but we note that they developed contour (edge) detection algorithms, whereas we can distinguish between edges and lines with positive and negative polarities. This is necessary for visual reconstruction; see below.

4. MULTI-SCALE REPRESENTATION

Here we focus on the multi-scale representation. Although NCRF inhibition can be applied at each scale, we will not do this for two reasons: (a) we want to illustrate line/edge behavior in scale space for applications like face detection and recognition, also visual reconstruction, and (b) in many cases a coarser scale, i.e. increased RF size, will automatically eliminate texture detail.

For illustrating scale space we can create an almost continuous, linear scaling space with 320 scales between $4 \leq \lambda \leq 40$, but here we will present only a few scales in order to show complications. Figure 6 shows events detected at five scales in the case of ideal, solid square and star objects. At fine scales (at left) the edges of the square are detected, as are most parts of the star, but not at the very tips of the star. This illustrates an important difference between normal image processing and developing cortical models. The latter must be able to construct brightness maps, and at the tips of the star, where two edges converge, there are very fine lines. The same

effect occurs at coarser scales, until entire triangles are detected as lines and even pairs of opposite triangles (right image). In the case of the square, we show medium-scale results with little cross-orientation inhibition (edges will continue at the corners). Lines will be detected at the diagonals, which will vanish at very coarse scales (also in case of the star object).

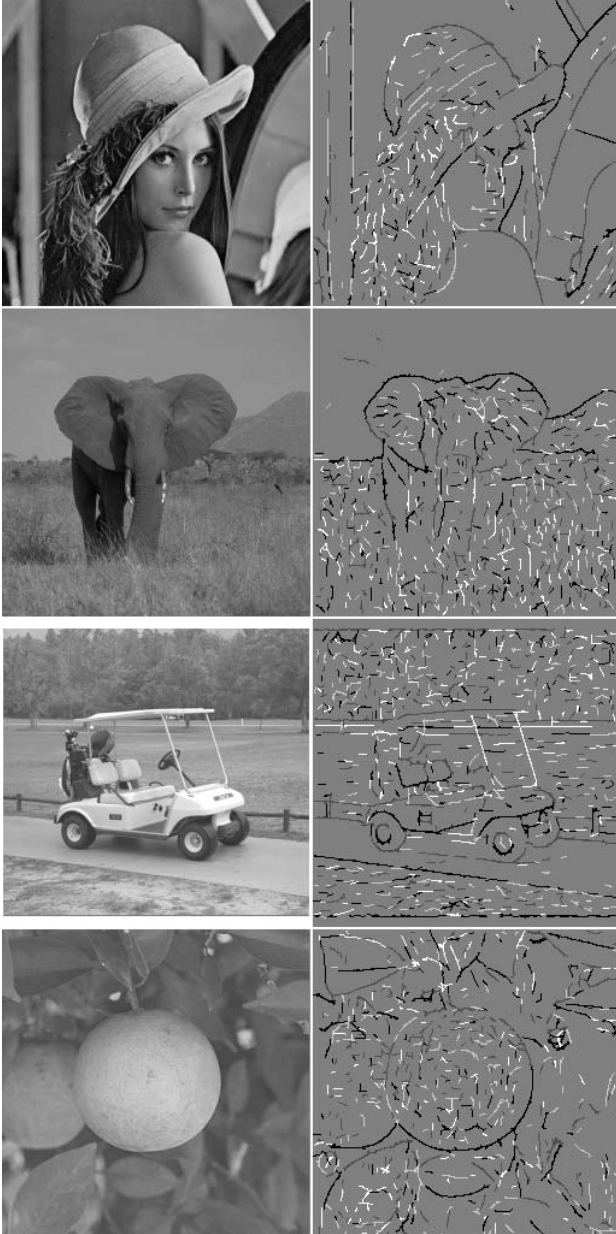


Figure 3: Fine-scale line/edge detection (see text).

Figure 7 shows the Fiona image with events detected at six scales, with decreasing amount of detail at coarser scales, and the result of a face detection algorithm that only exploits keypoint scale space [Rodrigues05b]. In the future, keypoints and lines/edges will be combined in order to also recognize faces in a database.

Figure 8 shows image reconstructions. These are not based on a complete set of (bandpass) wavelets complemented by one lowpass filter, a technique common in image coding, but on a virtual interpretation of lines and

edges: our visual system does not reconstruct in the normal sense, i.e. there is no cortical layer in which cell activity corresponds to input luminance. An active “line cell” is interpreted as a Gaussian intensity profile with a certain orientation, amplitude and scale, the size of the profile being coupled to the scale of the underlying simple and complex cells. In the same way an active “edge cell” is interpreted, but with a bipolar, Gaussian-truncated error-function profile. As for image coding, this interpretation must be complemented with a lowpass filter, a process that happens to exist by means of retinal ganglion cells with light-sensitive dendritic fields NOT connected to rods and cones [Berson03].

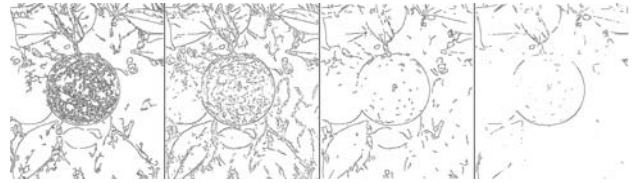


Figure 4: Edges by Canny, Nalwa, Bergholm and Iverson algorithms.

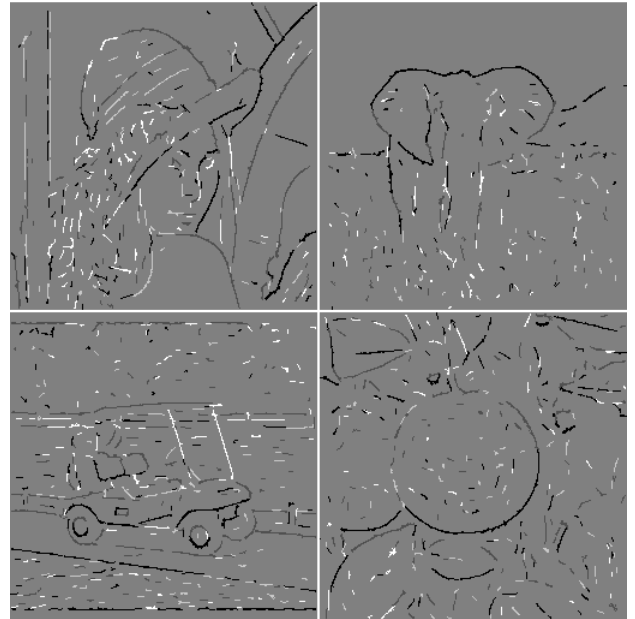


Figure 5: Line/edge detection with NCRF.

In order to show reconstructions, we must sum the lowpass (LP_σ) plus line (L_s) and edge (E_s) representations, using

$$R = \gamma LP_\sigma + (1 - \gamma) \cdot \sum_{s=1}^{N_s} W_s \cdot (L_s + E_s), \quad (10)$$

in which γ controls the balance between lowpass and other information, and W_s the relative weighting of the different scales. In Fig. 8 we used $\gamma = 0.5$, $W_s = 1/N_s$ and 16 equally-spaced scales $4 \leq \lambda \leq 20$. As can be seen, reconstructions are quite good. Better results are

obtained when using more filter scales, but optimization requires extensive experiments with different values for γ and W_s .

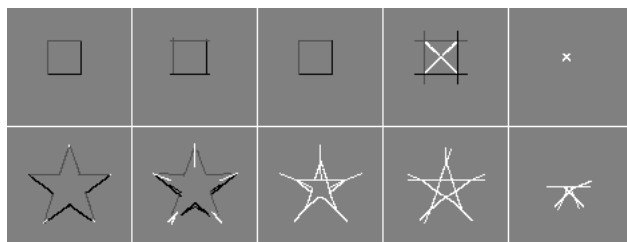


Figure 6: Multi-scale representation of a square and a star, left to right $\lambda=\{4, 12, 18, 24, 40\}$.

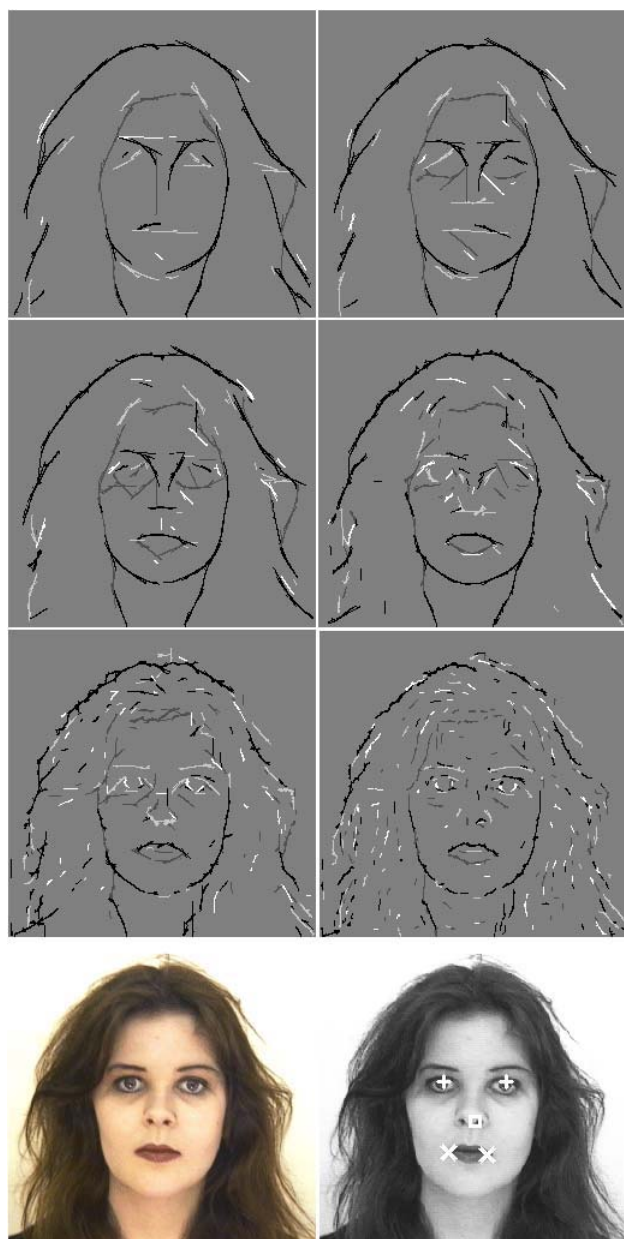


Figure 7: Multi-scale event detection on Fiona.



Figure 8: Reconstruction (see text).

5. DISCUSSION

In this paper we presented an improved scheme for line and edge detection, a scheme that can be used for creating a truly multi-scale image representation suitable for object detection/recognition and visual reconstruction.

Faces can already be detected on the basis of the multi-scale keypoint representation, i.e. by grouping keypoints at eyes, nose and mouth [Rodrigues05b], but lines and edges can (must!) complement keypoint information for developing robust face recognition algorithms, for which a cortical architecture of the what and where subsystems was explored by Deco and Rolls [Deco04].

For example, eyes, nose and mouth detectors can be complemented by a detector of the quasi-oval face outline, which is quite stable over many scales (Fig. 7). More information is available of course: eyebrows and quasi-elliptical mouth and eyes.

With respect to visual reconstruction, the symbolic representation with Gaussian line profiles and Gaussian-truncated error-function edge profiles, in combination with a lowpass filter, yields a "rendering" that is suitable for extending our brightness model [du Buf95] from 1D to 2D, for example for modeling brightness illusions.

Apart from "realistic" rendering in the case of brightness perception, non-photorealistic rendering can be obtained by selecting a few scales in combination with picking a few colors of the input image [Lam05]. Figure 9 shows an example of NPR in which the typical crayon effect was added by Corel's Photo-Paint standard "impressionist" filter. In the future we plan to develop real brushes (instead of using the line and edge profiles) and use these along simulated strokes (detected line and edge positions) in order to simulate real oil paintings and watercolors in certain styles, like Van Gogh and Turner.

In the future we can develop 3D wireframe models of objects, including faces, because we already have an optimized scheme for keypoint extraction [Rodrigues05a], which link lines and edges at vertices, and an initial model for disparity (stereo) [Rodrigues04a] that can be used to attribute depth to keypoints and therefore also to lines and edges. Such extensions are very useful for analyzing automatically facial expressions and to simulate realistic speaking avatars.



Figure 9: NPR rendering of Fiona

6. ACKNOWLEDGEMENTS

This investigation is partly financed by PRODEP III Medida 5, Action 5.3, and by the FCT program POSI, framework QCA III.

7. REFERENCES

- [Berson03] D. Berson. Strange vision: ganglion cells as circadian photoreceptors. *Trends in Neurosciences*, 26(6): 314–320, 2003.
- [Deco04] G. Deco and E.T. Rolls. A neurodynamical cortical model of visual attention and invariant object recognition. *Vision Res.*, (44):621–642, 2004.
- [Deemter96] J.H. van Deemter and J.M.H. du Buf. Simultaneous detection of lines and edges using compound Gabor filters. *Int. J. Patt. Rec. Artif. Intell.*, 14(6):757–777, 1996.
- [du Buf93] J.M.H. du Buf. Responses of simple cells: events, interferences, and ambiguities. *Biological Cybernetics*, 68:321–333, 1993.
- [du Buf95] J.M.H. du Buf and S. Fischer. Modeling brightness perception and syntactical image coding. *Optical Eng.*, 34(7):1900–1911, 1995.
- [Fleet91] D.J. Fleet, A.D. Jepson and M.R.M. Jenkin. Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210, 1991.
- [Grigorescu03] C. Grigorescu, N. Petkov and M.A. Westenberg. Contour detection based on nonclassical receptive field inhibition. *IEEE Trans. IP*, 12(7):729–739, 2003.
- [Heath97] M. Heath, et. al., A robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE Tr. PAMI*, 19(12):1338–1359, 1997.
- [Heitger92] F. Heitger et al. Simulation of neural contour mechanisms: from simple to endstopped cells. *Vision Res.*, 32(5):963–981, 1992.
- [Krüger97] N. Krüger and G. Peters. Object recognition with banana wavelets. *Proc. 5th Europ. Symp. Artificial Neural Networks*, 1997.
- [Lam05] R. Lam, J. Rodrigues and J.M.H. du Buf. Artistic rendering of the visual cortex. Accepted for: 2nd Workshop Luso-Galaico de Artes Digitais, V. N. de Cerveira (Portugal), 27 August, 2005.
- [Lee96] T.S. Lee. Image representation using 2D Gabor wavelets. *IEEE Tr. PAMI*, 18(10):pp. 13, 1996.
- [Livingstone00] M. Livingstone. *Vision and art: the biology of seeing*. Abrams, New York (NY), 2000.
- [Petkov93] N. Petkov, T. Lourens and P. Kruizinga. Lateral inhibition in cortical filters. *Proc. Int. Conf. Dig. Sig. Proc. and Inter. Conf. on Comp. Appl. to Eng. Sys.*, Cyprus:122–129, July 14–16 1993.
- [Rensink00] R. Rensink. The dynamic representation of scenes. *Visual Cognition*, 7(1–3):17–42, 2000.
- [Rodrigues04a] J. Rodrigues and J.M.H. du Buf. Vision frontend with a new disparity model. *Early Cogn. Vision Workshop*, Isle of Skye, Scotland, 28 May - 1 June 2004.
- [Rodrigues04b] J. Rodrigues and J.M.H. du Buf. Visual cortex frontend: integrating lines, edges, keypoints and disparity. *Proc. Int. Conf. Image Anal. Recogn.*, Springer LNCS 3211(1):664–671, 2004.
- [Rodrigues05a] J. Rodrigues and J.M.H. du Buf. Multi-scale cortical keypoint representation for attention and object detection. In J.S. Marques et al., editors, *2nd Iberian Conf. on Patt. Recogn. and Image Anal.*, Springer LNCS 3523: 255–262, 2005.
- [Rodrigues05b] J. Rodrigues and J.M.H. du Buf. Multi-scale keypoints in V1 and face detection. Accepted for: 1st Int. Symp. Brain, Vision and Artif. Intell., Naples (Italy), 19–21 October, 2005.
- [Würtl00] R.P. Würtl and T. Lourens. Corner detection in color images by multiscale combination of end-stopped cortical cells. *Image and Vision Computing*, 18(6–7):531–541, 2000.
- [Zeki00] S. Zeki. *Inner vision: an exploration of art and the brain*. Oxford Univ. Press, 2000.

Web LOG File Information Analysis and Visualization Inside Organizations

Florin Zamfir
Dep. of Electronics and Telecommunications
Dep. of Communication and Art
University of Aveiro
florin@ca.ua.pt

Óscar Mealha
Dep. of Communication and Art
University of Aveiro
oem@ca.ua.pt

José Nunes
Dep. of Communication and Art
University of Aveiro
jnunes@ca.ua.pt

Beatriz Sousa Santos
IEETA – Instituto de Engenharia Electrónica e Telemática de Aveiro
University of Aveiro
3810-193 Aveiro, Portugal
bss@ieeta.pt

Abstract

In the context of this work, organizations can be understood as organic systems composed of people, processes and procedures. The intranets are incorporated in organizations to support their mission and guarantee effective process delivery within an efficient framework. These technological platforms, with a web driven user front-end, are able to log several user interaction details. The main goal of this paper is to report how the analysis and visualization of interaction patterns inside an organization's internal web based system can be achieved, as mentioned by [Eick01] in a more general context and by [Healey01] in a specific e-commerce scenario.

In order to validate the research on conceptual models, information architecture and several visualization methods, a prototype is being developed, tested and permanently upgraded. At this stage of iterative development, the prototype can be considered as an experimental research platform [Nunes03], capable of integrating and testing specific visualization schemes [Mealha04] and visual correlation procedures. Besides this, the prototype incorporates features that can visually support answers for questions, such as:

- i How is the site used?*
- ii Who is using the site?*
- iii What are the site areas / sectors / pages of interest?*
- iv What statistical information can be obtained from the log files?*
- v What are the problematic areas?*
- vi Which usability problems can be identified?*

Some of the methodological stages, inherent to our research, will be described and exemplified in this paper, specifically, the conceptual model of the prototyped system, overall prototype design and unified user interface, and visual inspection methods/schemes. Examples will be presented as prototype screenshot images depicting visual inspection settings and results.

Keywords

Information visualization, information and communication management, usability, web logs.

1. INTRODUCTION

On the behalf of evolution of nowadays information systems, worldwide institutions developed and supported important informational frameworks, based on the latest hypermedia presentation systems. An emerging problem

occurred, related to the management and understanding of these vast informational workspaces.

The latest evolution of Internet based technologies gave the institutions the means to develop large websites, meant mainly to support their strategic and operational

decisions. However, the management of such systems became more and more difficult, due to the unpredictable growth facilitated by these technologies. Starting with the role of technological support, the Internet became an efficient instrument to support business decisions, the prediction and understanding of its potential represents advantage for every institution.

In this context, partially reported by this paper, good analysis tools are needed, to support the managers in decision making, based on analysis results of their organizational information workspaces. However, this is not an easy task, from the computational point of view, specifically the development of applications that have quantitative and qualitative correlated representation and inspection features of the informational workspace.

The following sections describe methodological stages and usage examples of the prototyped system that represents a current research framework which provides answers for some strategic questions in the context of organizational web-based information and communication management.

2. SYSTEM MODEL

The prototype system model is organized to achieve two main goals focused on intranet usage. One is related to the site structure, the other to the interface design (visual workspace organization). It provides solutions to detect structural problems revealed by visual inspection of usage patterns with the possibility of having simultaneously qualitative and quantitative visual inspection methods. Another vital area, where severe system problems occur, is at interface level. The visual workspace hosts the visual component of the interface, i.e. an information area where the prototype user decides on major navigational issues. The registered interaction activity log is processed and the usage pattern is represented and analyzed.

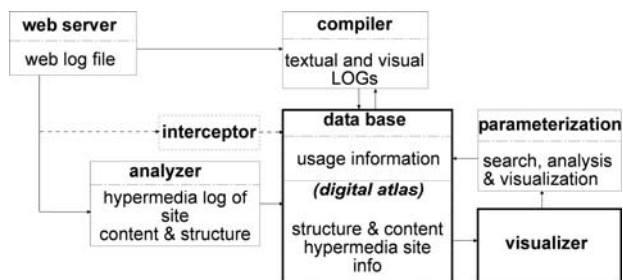


Figure 1 System Model

The global system architecture, as represented in figure 1, is user goal oriented and gathers raw information [Shneiderman96] from system log files, from controlled experience environment data (compiler) and from a copy of the pages loaded from the server and presented to the user (analyzer). A database is used to store the data and the visualizer to display it. The system conceptualization exercise and internal information structure uses, as an overall reference, the model proposed by [Card01, p. 554]. As a consequence of a state of the art exercise, it also complies with other raw information sources such as

eye movement tracking technologies for a possible future system version.

3. UNIFIED INTERFACE-VISUAL CORRELATION STRATEGIES

Our system uses an interface strategy that integrates several visualization schemes, dialogues and feedback information units into one unified interface. It delivers an articulated and contextualized visual inspection method sustained on multiple simultaneously present views. The visual correlation of qualitative and quantitative network information properties is considered as shown in Figure 5.

An inspection exercise starts with the specification of a goal that may consider the homepage (or some other page) as a starting point. It ends with the specification of a final page, where the information or communication service entry is located, and is specified as a usability evaluation goal. An articulated and tightly coupled set of views are generated for deeper inspection analysis of inter-page related statistics and interface design parameters. All these schemes are triggered and redrawn by the same event and use the same raw information space to represent their visual features.

4. DATA VISUALIZATION SCHEMES

Visualization functions can be subdivided according to three main areas of concern, at this stage of conceptualization, development, use and evaluation:

- *Visual inspection of intranet structure and efficiency;*
- *Visual inspection of interface design coherence, specifically for Hot-spot and usage path overlapped representation;*
- *Representation of tree-structured information traversing in time.*

The following sections describe some of the authors' recent work in these areas. Some examples are presented based on real data to highlight the features inherent to the conceptualized and developed visualization schemes.

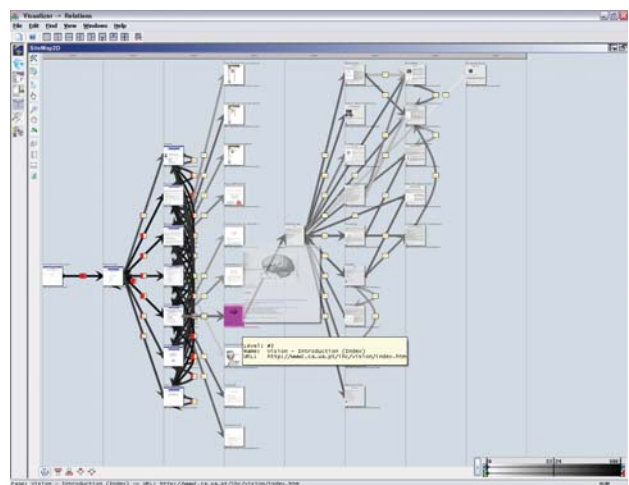


Figure 2 Tree representation with quantitative and qualitative usage data

4.1 Tree map based visual inspection features

Hierarchical tree based representation constitutes one of the visualization schemes. This scheme, Figure 2 and upper left quadrant of Figure 5, represents site page relational tree map [Card99], superimposed with usage data in qualitative and quantitative form with dynamic visual inspection capabilities.

The Colour Look Up Table (LUT), is used in this scheme, in greyscale for intensity representation purposes. It is a dynamic tool with multiple possibilities in its slider functions similar to slider applications as reported by [Eick99]. The following list describes some of the LUTs dynamic visual inspection capabilities, figure 3 and 4:

- Remaps or masks usage behaviour into colours;
- Change between different colour schemes to increase contrast/visibility;
- Filter out representation objects to highlight specific usage information;
- Dynamically synchronize with the selected object's usage properties and remap them;
- Mask or hide various levels of details to allow high-lighting others.

This visualization scheme answers the first, third and fifth questions presented on the abstract, the site inter-connections, pages and areas of interest being highlighted as the user explores the scheme, filters are helping distinguish between statistical usage levels.

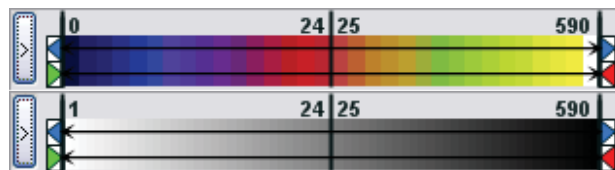


Figure 3 Colour LUT and greyscale LUT

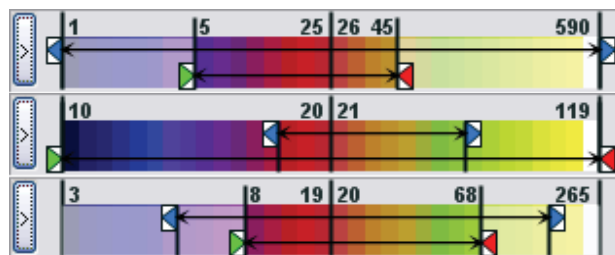


Figure 4 LUT dynamic filtering, masking and remapping of usage data

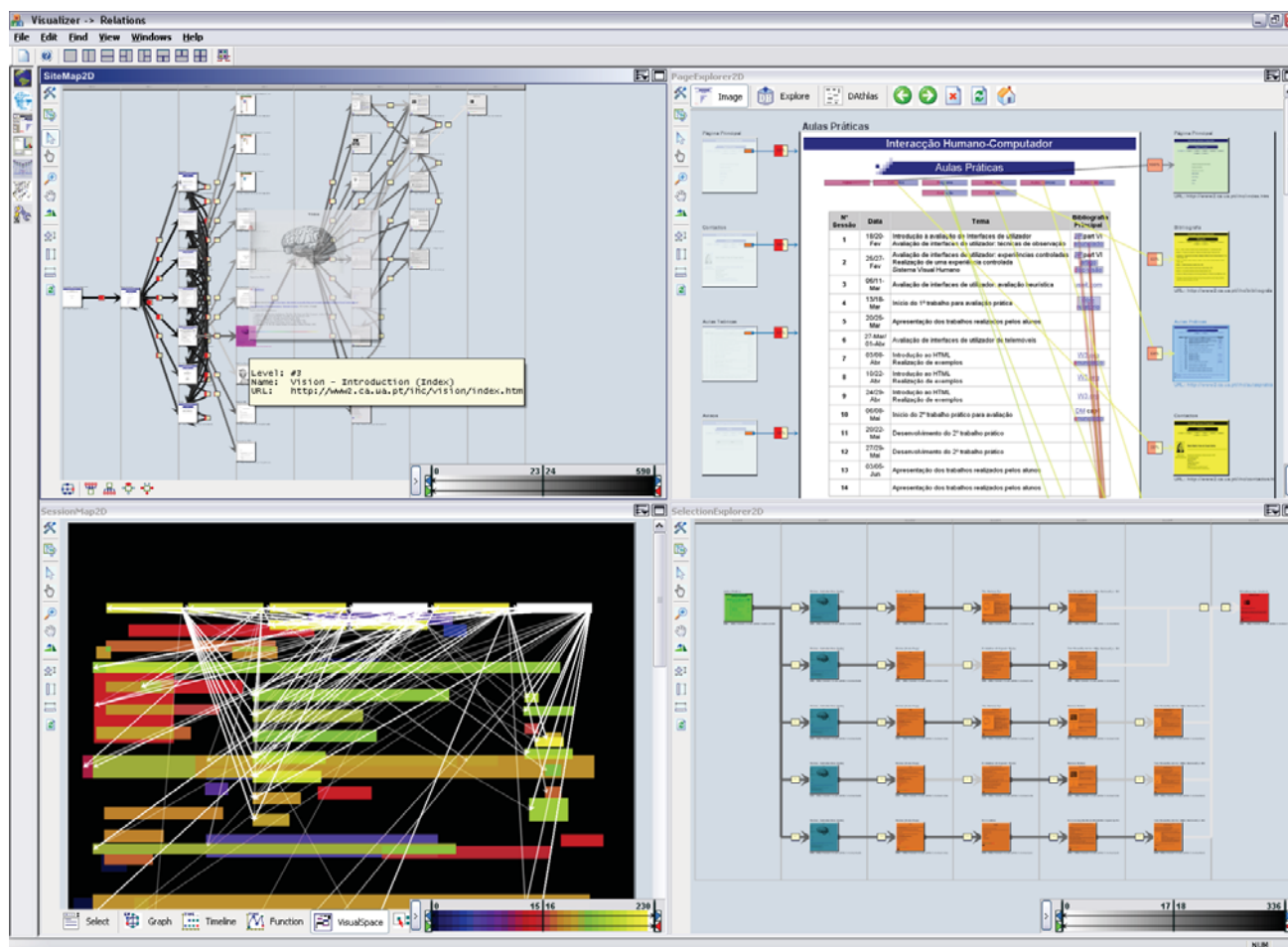


Figure 5 Unified interface with tightly coupled visual inspection methods

4.2 Page inspection tool

The site page inspection tool has a major goal of representing, in detail, all the hypermedia related information of a specific chosen site page, as seen in upper right quadrant of Figure 5 and Figure 6.

This scheme can:

- Highlight statistical hotspots usage information and map it to the actual page position;
- Bind the selected page referrers using the physical position of their relational hotspots;
- Use the physical site interconnections mapped to the visual usage function for Referrer → Page, as for Page → Child page representations;
- Use masking and filters to highlight the connections Referrer → Page → Child page
- Interactive scheme exploration allowing page to page navigation, driven by statistical usage data and the spatial hotspot location on the visual field.

This representation of the page helps the detection of usability problems by analysing the usage statistical information extracted from the log files applied to the structural UI of each page of the site. It answers to first, fourth and fifth questions presented on the abstract.

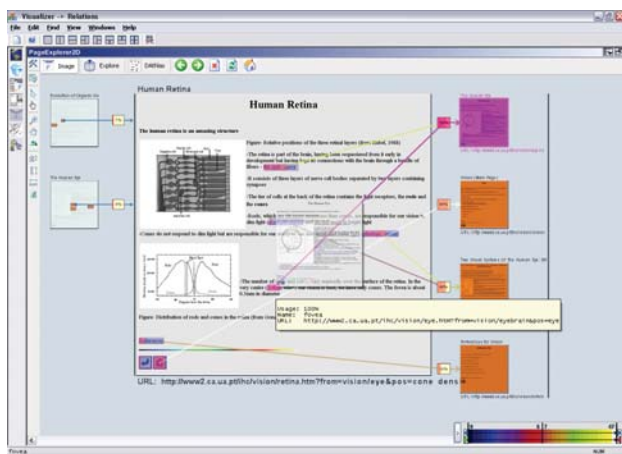


Figure 6 Hypermedia information superimposed to usage data of a specific site page

4.3 Visual workspace

The visual workspace (interface area on computer screen) inspection scheme delivers information on how the user interface of the site is being used. The scheme overlaps colour coded hotspot localization with usage data and hypermedia relations. As can be seen in Figure 7 and lower left quadrant of Figure 5, features can be reported as:

- Overlap of hotspots of a session's usage to understand/validate the visual workspace organization;
- Direct mapping of session jumps to explore visual/interface relations between pages;
- Visual inspection of usage patterns and interface design layouts used to detect UI design errors.

This visualization scheme helps viewing the hottest areas of the user interface, the visual workspace organization of the selected usage sessions being represented. It improves the detection of UI design errors by analyzing the concentration of hits on a specific visual area, one that does not comply with specific usability patterns. It answers the last three questions presented on the abstract.

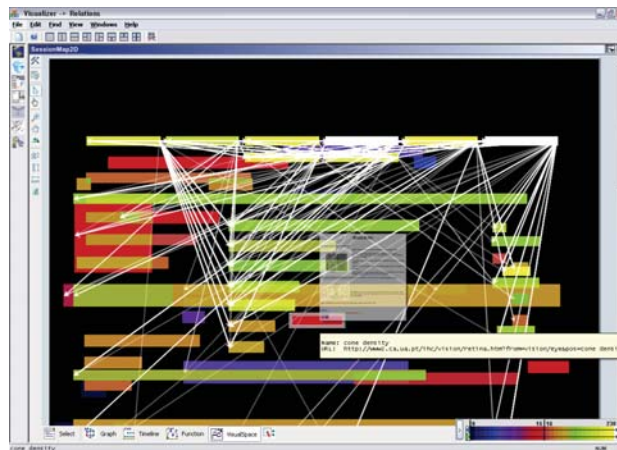


Figure 7 Overlap of hotspot representation superimposed to usage data on visual workspace

4.4 Behavioural function

The behavioural function $B(x)$ represents the basis to calculate and identify a potential problem in the traversing of site data. Based on real usage data, this discrete function is defined with page position in hierarchical site tree (Y axis) and the time at which it was visited (X axis), Figure 8. The differential $\Delta B(x)/\Delta x$ of this function, specifically its high differential values, may depict a serious problem in page relation. Such a situation can be a synonym of information that is out of context in a specific page cluster, or simply just a bad page link established from a deep tree level to a near home page level. This function's major goal is to constitute an awareness feature, an instrument that marks a specific page hotspot/relation as needing inspection due to suspicious hyper-jump in the tree map representation.

The behavioural function answers the first three questions presented on the abstract. It allows the user to observe the site usage sessions, evaluate their comprehension of the inter-site relations and the mapping of the site structural content to the site navigation features.

The colour-codes and shape-codes used by the scheme are interpreted as follow:

- Green and red shapes (filled circle or filled square) are used to highlight the beginning and, respectively, the end of a usage session;
- The blue filled circles are used to code regular jumps between site pages, forward exploration of neighbour site levels being considered regular jumps;
- Yellow filled squares represent user jumps out of context or jumps backward several levels, Figure 9;

All these features help the automated detection of incoherent user jumps to backward located pages; the jump

distance between site levels giving a good approximation of the user's effort in terms of site exploration.

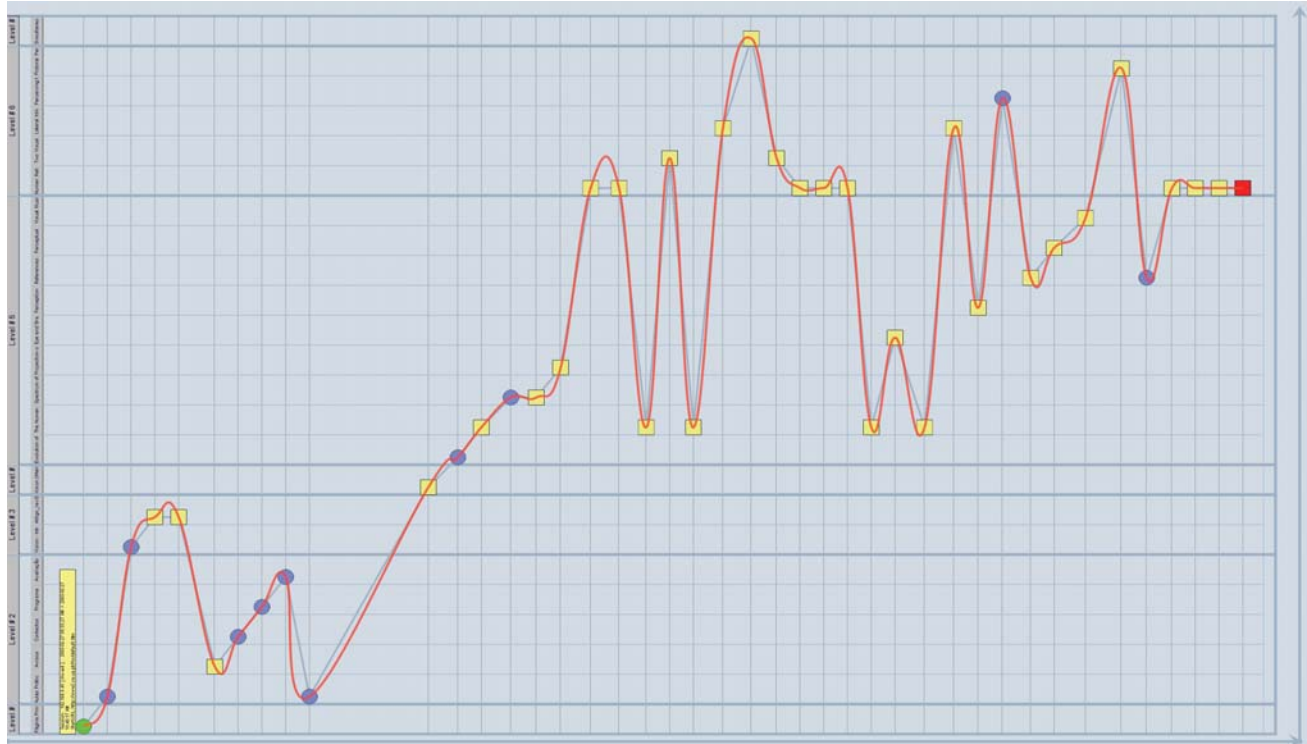


Figure 8 Behavioural function based on page traversing, session data

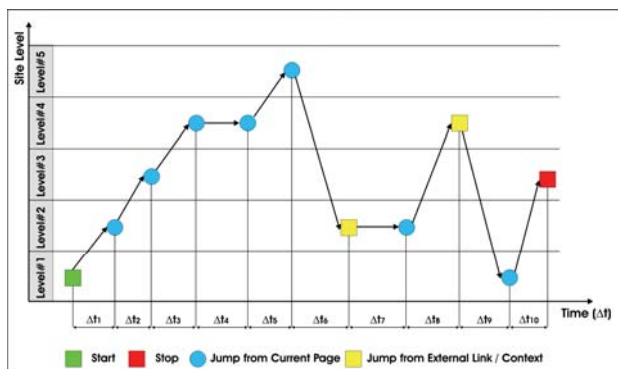


Figure 9 Behavioural function concept

5. CONCLUSIONS

The work reported in this paper represents a coherent research framework integrating visualization concerns in the analysis and representation of interaction / usage patterns inside web based technology / information repositories, specifically organization's intranets.

Besides the conceptual system model, specific details on the information architecture are presented alongside the visualization schemes that are used and the best visual correlation strategies to achieve efficient analysis procedures. Results are presented in the form of prototype screenshots depicting inspection conclusions applied to real data obtained inside University of Aveiro's (Portugal) web repositories. As a synthesis:

- i* Common visualization schemes, such as tree representations, are augmented with dynamic visual inspection tools based on LUTs and their specific slider functions to enhance inspection and analysis;
- ii* The visual workspace, or visual interface area on the screen, is inspected with a specific scheme that overlaps interaction hotspots and their hyper-relations between pages;
- iii* The behavioural function is introduced as an awareness feature that is used to capture form usage data, a potential information context problem in relations with deep tree pages with near home page level pages.

Future work is planned to enhance data visualization schemes directly related with alternative or augmented new website analysis and capture features.

Although ongoing evaluation is expected, the latest version of our system prototype has undergone an empirical and analytical evaluation reported at [Santos04]. The prototype's next evaluation phase will be developed as a real case study, considering common and expert evaluation subjects, and a specific organizational website. The purpose of this evaluation session will be to ask evaluation subjects to perform regular website management tasks, for an objective analysis of the proposed website. The evaluation's starting point is the capture of the website structure followed by the analysis of its usage data and website user's behaviour.

6. REFERENCES

- [Card99] Card, S. K., Mackinlay, M. J. and Shneiderman, B., (editors). (1999). *Information Visualization: Using Vision to Think*. San Diego, CA, Morgan Kaufman, pp 493-502.
- [Card01] Card, S. (2001). *Information Visualization*, in *The Human-Computer Interaction Handbook Fundamentals, Evolving Technologies and Emerging Applications*, ed. Jacko, J., Sears, A., Lawrence Erlbaum Associates, pp. 544-582.
- [Eick01] Eick, S.G. (2001). *Visualizing Online Activity*. in *Communications of the ACM*, Vol. 44, No. 8, pp. 45-50.
- [Eick99] Eick, S.G. (1999). *Data Visualization Sliders*. In *Readings in Information Visualization: Using Vision to Think*, (editors) Stuart K. Card, Jock D. Mackinlay and Ben Shneiderman., San Diego, CA, Morgan Kaufman, pp 251-252.
- [Healey01] Healey, C.G., Amant, R. St. and Chang, J. (2001). *Assisted Visualization of E-commerce Auction Agent*, presented at Graphics Interface 2001 – Canadian Human-Computer Communications Society, pp. 201-208.
- [Mealha04] Mealha, O., Santos, B.S., Nunes, J., Zamfir, F. (2004). *Integrated Visualization Schemes for an Information and Communication Web Log Based Management System*. 8th International Conference Information Visualization, (Eds.) Ebad Banissi, Katy Börner, Chaomei Chen, Muhammad Dastbaz, Gordon Clapworthy, Anthony Faiola, [Izquierdo] Ebroul Izquierdo, Carsten Maple, Jonathan Roberts, Chris Moore, Anna Ursyn and Jian Zhang, Published by the IEEE Computer Society, London, England, ISBN 0-7695-2177-0, ISSN 1093-9547, pp 295-301.
- [Nunes03] Nunes, J., Zamfir, F., Mealha, Ó., & Santos, B. S. (2003). Web LogVisualizer: A Tool for Communication and Information Management. In *Proceedings of the 10th International Conference on Human-Computer Interaction – HCI International 2003*, Heraklion – Crete – Greece, Vol. 3, pp. 824-828.
- [Santos04] Santos, B.S., Zamfir, F., Ferreira, C., Mealha, O., Nunes, J. (2004). *Visual Application for the Analysis of Web-Based Information System Usage: A Preliminary Usability Evaluation*, 8th International Conference Information Visualization, (Eds.) Ebad Banissi, Katy Börner, Chaomei Chen, Muhammad Dastbaz, Gordon Clapworthy, Anthony Faiola, Ebroul Izquierdo, Carsten Maple, Jonathan Roberts, Chris Moore, Anna Ursyn and Jian Zhang, Published by the IEEE Computer Society, London, England, ISBN 0-7695-2177-0, ISSN 1093-9547, pp 812-818.
- [Shneiderman96] Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations, In *Proceedings of Visual Languages*, (Boulder, CO, Sept, 3-6), IEEE Computer Science Press, Los Alamitos, CA, pp 336-343.

LEMe Wall: Desenvolvendo um Sistema de Multi-Projecção

Bruno Rodrigues de Araújo, Tiago Guerreiro, Ricardo Jota
IMMI INESC-ID/ IST, Universidade Técnica de Lisboa
Rua Alves Redol, 1000-029 Lisboa
{brar,tjvg,rjc}@immi.inesc-id.pt

Joaquim A. Jorge, João M. Pereira
Dep. Eng^a. Informática, IST/UTL
Av. Rovisco Pais, 1000 Lisboa
jorgej@acm.org

Abstract

Este artigo apresenta a LEMe Wall, um ambiente inteligente organizado em torno de um sistema de multi-projecção. Este é composto por três componentes essenciais: uma grelha de 4x3 projectores devidamente suportados, uma tela flexível que oferece o suporte para a visualização e um aglomerado de máquinas que controlam a projecção. O ambiente é complementado por um conjunto de sensores e actuadores que aumentam a imersão e naturalidade de interacção. Neste artigo são descritos os maiores desafios na construção deste sistema de multi-projecção bem como as opções tomadas ao longo do desenvolvimento do ambiente inteligente.

Keywords

Ecrãs de larga escala, Multi-projecção, Retro-projecção, Calibração, Ambiente Inteligente.

1. INTRODUÇÃO

Os ecrãs de larga escala são uma tecnologia emergente que tem oferecido várias aplicações inovadoras na área de Computação Gráfica nos últimos cinco anos, sendo objecto de investigação em várias instituições [IGD-A4, Hereld00, Li00, PennState]. As potencialidades de visualização e interacção deste género de dispositivos apresentam-nos como sucessores à metáfora do tampo de secretária, que se mostra insuficiente, quando se procura visualizar imagens de alta resolução ou interagir de forma mais natural (semelhante à interacção entre humanos).

A realização de um ambiente imersivo sugere, automaticamente, um conjunto de mecanismos adicionais que potenciam as capacidades do mesmo, transformando a sala onde este se encontra num verdadeiro ambiente inteligente. Este ambiente, contempla, para além do ecrã, a presença de sensores e actuadores que rodeiam os utilizadores extrapolando a imersão para todos os níveis sensoriais, facilitando a interacção e gestão de dados. Este artigo apresenta os conceitos e tecnologias utilizadas, bem como os vários paradigmas existentes e decisões tomadas durante o processo de lançamento da sala de multi-projecção. Esta iniciativa apresenta grandes potencialidades e será fruto de inúmeros trabalhos de investigação, sendo este o seu enquadramento no projecto LEMe, o qual é descrito sumariamente em seguida.

1.1 Contexto

O LEMe é uma iniciativa conjunta do IST e de várias empresas do sector para a criação de uma massa crítica científica importante e de uma capacidade de formação de excelência dos alunos do Instituto Superior Técnico, campus do Tagus Park. O LEMe constitui um centro de projectos inovadores no Ensino e na Investigação e Desenvolvimento, enquadrando estudantes de licenciatura, pós-graduação e professores do IST em colaboração

com investigadores e profissionais da indústria. É neste contexto que nasce o projecto da sala de multi-projecção, situada no IST-Tagus Park, onde se procura investigar ao nível mais avançado na área de ambientes imersivos inteligentes.

1.2 Abordagem Global

A dimensão e qualidade da imagem é o ponto inicial e essencial de discussão no lançamento de um ecrã de larga escala. O aumento da dimensão da imagem obtida através de um projector traduz-se na redução de qualidade da mesma, e num ecrã de grandes dimensões este problema aumenta, impossibilitando a utilização de uma projecção única. Os avanços na multi-projecção tornaram exequível a realização de imagens compostas, sendo o resultado final um mosaico de imagens. Adoptámos esta solução,



Figura 1: Evolução da sala do LEMe

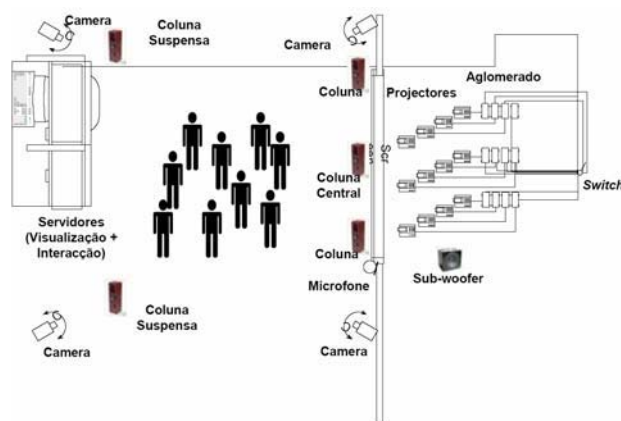


Figura 2: Dispositivos sala do LEMe

tendo investido numa matriz de quatro por três (12 projectores), onde salvaguardamos a elevada qualidade da imagem, com uma relativamente reduzida área de projecção por cada projector. Esta solução apresenta contudo um conjunto de detalhes técnicos de calibração relacionados com a uniformidade da imagem final que tiveram que ser considerados. O sistema é de retro-projecção sendo esta característica obrigatória se queremos dispor de uma área para o utilizador, sem interferir com os volumes de projecção. A utilização de projecção frontal seria fruto de problemas em relação à geometria da imagem, composição do mosaico e nunca seria possível usufruir da área de interacção totalmente. A secção 2 descreve globalmente a infra-estrutura, apresentando os componentes essenciais ao sistema. A Calibração dos projectores tem especial relevância e é descrita na secção 3. A secção 4 foca-se na gestão dos componentes existentes na sala e a secção 5 apresenta as soluções para a visualização de imagem única. A interacção utilizador é contemplada na secção 6 e as conclusões ao artigo são apresentadas na secção 7.

2. DESCRIÇÃO DA INFRA-ESTRUTURA

O sistema de multi-projecção é composto por três elementos principais: uma tela que oferece o suporte físico para a visualização, uma estrutura contendo a matriz de projectores e um aglomerado de máquina de forma a controlar a projecção. O sistema de projecção é complementado por um sistema de som 5.1 e microfones para enfatizar a experiência imersiva, assim como possibilitar interacções baseadas no uso da voz quer seja através do reconhecimento de comandos discretos ou do uso de linguagem natural. Para suportar o desenvolvimento de novos mecanismos de interacção, um conjunto de cinco câmaras foi instalado e futuramente esta sala inteligente possuirá uma rede de sensores de pressão colocados no chão falso. A configuração da instalação é apresentada na Figura 2.

2.1 Ecrã

O ecrã cujas características são apresentadas na Tabela 1 é uma tela de material flexível que suporta retro-

projectão oferecendo um baixo ganho de 1.0 e uma boa distribuição deste em função do ângulo de incidência do utilizador em relação a tela, de forma a minimizar as variações de intensidade. Existem várias alternativas flexíveis ou rígidas, sendo relevante as características do material, tais como, o ganho, contraste, brilho entre outras. Optámos por uma configuração flexível tendo em conta a relação qualidade/custo. Apesar da qualidade ser menor e a tela estar sujeita a oscilações, o custo é bastante menor e obtém-se os resultados requeridos.

2.2 Matriz de Projectores

Os 12 projectores estão organizados numa estrutura formando uma matriz de 3 linhas de 4 projectores. Essa matriz de projectores (Figura 3) está colocada numa estrutura de alumínio modular oferecendo flexibilidade para futuras extensões da instalação assim como a estabilidade necessária para evitar turbulências na imagem durante a projecção. Cada um dos 12 projectores é suportado por uma base de alumínio especialmente concebida para permitir o controlo dos seis graus de liberdade com a alta precisão requerida para efeitos de calibração.

Os projectores (Tabela 1) são digitais (DLP) apresentando uma resolução nativa XGA. A escolha de projectores é um elemento essencial na realização de um ecrã de grandes dimensões e é uma das decisões mais difíceis de tomar devido à grande diversidade de marcas e modelos disponível. Comparado à tecnologia LCD, a tecnologia DLP apresenta melhor contraste e a imagem é menos sensível a degradações ao longo do tempo. O projector escolhido apresenta uma boa relação custo/qualidade nomeadamente dado a sua alta luminosidade e resolução.

Tela flexível	Draper Cineplex
Dimensões	4,00 m x 2,25 m
Ganho	1.0
Projectores	HP VP 6120 (Tipo DLP)
Luminosidade	2000 Lumens
Resolução	1024 x 768
Contraste	2000 : 1
Computadores	HP wx 4100
Processador	Pentium 4 (FSB 800) 3.0 MHz
Memória	2 Gb RAM (PC 3200)
Placa Grafica	NV QuadroFX 3000,256 Mb
Servidor	HP wx 8000
Processador	Pentium 4 Xeon, 3.2 MHz
Memória	4 Gb RAM
Placa Gráfica	NV Quadro FX 3000,256 Mb

Tabela 1: Características do material



Figura 3: Estrutura de suporte dos projectores

2.3 Aglomerado de Computadores

dois sistemas: estações gráficas de memória partilhada tal como as SGI Onys2 ou um aglomerado de PC's. Optámos por um aglomerado de trezes PC's Linux; cujas características são apresentadas na Tabela 1 dado apresentar um custo mais atractivo, uma fácil manutenção e boas possibilidades de extensão para aumentar o desempenho computacional. O nosso aglomerado é formado por doze computadores dedicados para o controlo da visualização sendo cada computador ligado a um dos projectores da matriz de visualização. Estes computadores são organizados numa rede de um gigabit de forma a possibilitar uma inter-comunicação rápida e é controlada por um servidor responsável por dividir e coordenar a projecção individual. Para além dos computadores de visualização e do servidor existem igualmente dois computadores para gerir as câmaras instaladas na sala, assim como um computador para processamento de voz e som. O núcleo central das varias aplicações e demonstrações, existentes e futuras, estão instaladas no servidor que con-trola o aglomerado Linux



Figura 4: Protótipo da base para um projector

de visualização. A distribuição escolhida foi Gentoo de forma a compilar os vários módulos tendo em conta as características materiais de cada computador.

3. CALIBRAÇÃO

Um sistema de multi-projecção necessita de vários tipos de calibração e correcções de forma a produzir uma única imagem de alta resolução sem ser detectável o mosaico de imagens que a compõe. Os problemas de calibração deste tipo de projecção são descritos nesta secção tendo em conta as características da tela e dos projectores.

3.1 Calibração geométrica do mosaico

A calibração geométrica contempla o alinhamento dos projectores entre eles, que requer alta precisão; no nosso sistema um pixel corresponde a um milímetro. Para esse efeito desenvolvemos um dispositivo que controla os seis graus de liberdade do projector. O dispositivo como se pode ver na Figura 4 é constituído por dois andares. O primeiro passo é garantir que o projector está horizontal de forma a não utilizar as funcionalidades de keystone dos projectores dado que distorce a linearidade dos pixels devido a ser digital. A horizontalidade é controlada pelo andar inferior do dispositivo que permite igualmente alinhar o paralelismo do projector com a tela. Quanto ao andar superior possibilita controlar as translações do projector de forma a emparelhar os vários mosaicos entre eles. O processo de calibração geométrica é moroso sendo necessário dedicar, em média, 2 horas por cada projector usando o dispositivo que oferece um controlo das translações da ordem de 0.1 mm a uma distância de projecção de 1,72 metros. Este processo de calibração é auxiliado por uma aplicação que é capaz de projectar vários padrões de grelhas e pelo uso de um nível óptico laser que projecta um linha horizontal ou vertical de forma a facilitar o alinhamento dos mosaicos. Existem abordagens [Raskar03,Brown02] que não necessitam de calibração geométrica, mas não possibilitam usar toda a resolução dos projectores para a imagem final, sendo destinados a configurações temporárias.

3.2 Correção devido a sobreposição

Mesmo tendo em conta a alta precisão oferecida pelo dispositivo de suporte e a ferramenta usada para auxiliar a calibração, a justaposição homogénea de dois mosaicos revela-se impossível e a transição da luz nas arestas da própria imagem de cada projector não é completamente abrupta. Por esse motivo é necessário sobrepor ligeiramente os mosaicos. A sobreposição provoca luminosidades diferentes dado que na zona sobreposta podem incidir dois ou quatro projectores. A resolução desse problema não foi ainda implementada, mas existem uma solução denominada "edge-blending" que se baseia em corrigir as intensidades. Existem dois tipos de "edge blending": soluções físicas obstruindo o contorno do frustrum do projector [Li99, Hereld00, Stone01] ou soluções por software mais flexíveis e eficazes usando filtros na imagem contemplando as zonas de luminosidade [Surati99]. A segunda abordagem será seguida na nossa configuração.

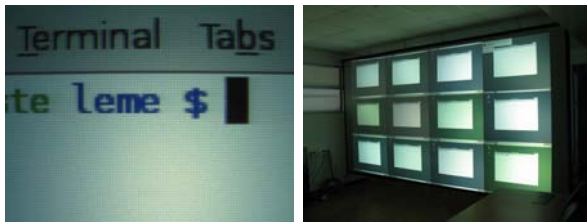


Figura 5: Problemas de brilho e de cor

3.3 Correção de brilhos e cores

Como se pode verificar na Figura 5, existem dois tipos de problemas relacionados com cor e brilho a nível de um sistema de multi-projectção: problemas intra-projector e inter-projector. Um dos problemas intra-projector é a intensidade da luz do projector na tela ser variável num mesmo mosaico. Isto deve-se ao facto do projector ser uma fonte de luz pontual levando a que a incidência sejam diferentes ao longo do plano que é a tela. [Stone01] apresenta uma solução usando uma lente de Fresnel para cada mosaico, mas esta solução revela-se demasiado cara e não possibilita corrigir as arestas dado que a sobreposição não é possível. O uso de mapas de fotometria permite apresentar uma solução software sem custos adicionais [Majunder02].

Relativo aos problemas inter-projectores como se pode ver na Figura 5, a luminosidade e cor de dois projectores do mesmo modelo nunca é exactamente igual. Por esse motivo, é necessário aplicar correcções específicas a cada projector de forma a corrigir a distribuição de cores e o brilho na tela. Existem várias alternativas usando mapas de gama [Majunder04] de forma a apresentar uma imagem homogénea que tencionamos aplicar no nosso sistema de visualização de uma forma transparente para as aplicações. Métodos exactos aplicam a regra do mais fraco mas reduzem demasiado a intensidade final do projector. Um compromisso tem de ser implementado seguindo abordagens alternativas [Majunder05].

4. GESTÃO DE INFRA-ESTRUTURA

O servidor do aglomerado tem como função principal controlar os mecanismos de interacção e visualização existentes na sala. É através deste que os computadores e respectivos projectores são comandados e se torna possível uma utilização eficaz da sala. Para atingir o controlo centralizado foi necessário desenvolver uma ferramenta para a gestão dos dispositivos existentes que contempla os computadores do aglomerado e os projectores.

No que diz respeito à gestão dos computadores, a ferramenta desenvolvida contempla o ligar/desligar de todos os computadores do aglomerado (através da tecnologia Wake On Lan), lançar o gestor de janelas de cada máquina, bem como o lançamento de demonstrações e o acesso remoto aos sistemas de ficheiros dos constituintes do aglomerado. A ferramenta oferece também uma aplicação para auxílio à calibração referida na secção anterior. Esta baseia-se no desenho de grelhas para a calibração geométrica e padrões de cor para as calibrações das características inerentes ao projector.

O facto de termos um servidor que controla um aglomerado permite-nos utilizar esses computadores como canalizadores de informação para os projectores. Assim, é possível obter o ambiente de trabalho do servidor projectado no ecrã de larga escala. Para efectuar esta distribuição foi utilizado o [DMX], Distributed Multihead X, sistema que possibilita a qualquer servidor X a distribuição do ambiente de trabalho num ecrã de larga escala. Xdmx é um servidor proxy X que providencia suporte para vários ecrãs ligados a máquinas diferentes (em que cada um corre um servidor X típico). Quando o Xinerama é usado em conjunto com o Xdmx, as várias imagens das diferentes máquinas são apresentadas ao utilizador como um ecrã único. Para interagir com o ambiente de trabalho foi utilizado a aplicação [Synergy] que facilita a partilha de rato e teclado único entre vários computadores, sem necessidade de hardware adicional. Poderíamos ter optado pela solução hardware KVM mas seria um custo extra, sem vantagens adicionais.

5. VISUALIZAÇÃO

Existem varias soluções para apresentar uma imagem única num sistema de multi-projectção. Esse tipo de aplicações é responsável por dividir a janela de visualização em várias sub-janelas que serão associadas a cada projector. Dado que o sistema é controlado por um aglomerado de computadores, o servidor central coordena a visualização e efectua a distribuição dessa pelos computadores. Toda a configuração das sub-janelas é efectuada pelo servidor. No caso de aplicações interactivas como por exemplo animações, o servidor é responsável por sincronizar as trocas de frame buffer de cada computador de forma a evitar dessincronismos entre elementos do mosaico. Nos próximos parágrafos apresentamos e discutimos vários sistemas distribuídos de visualização testados na nossa tela.

5.1 Sistema CHROMIUM

Começamos por configurar uma solução baseada no Chromium [Humphreys02]. O sistema Chromium é uma camada de abstracção que substitui o controlador OpenGL oferecendo a possibilidade de distribuição da visualização para qualquer aplicação baseada na API OpenGL sem necessitar alterações na aplicação. Através da configuração do Chromium é possível definir o mosai-



Figura 6: QuakeIII Arena no Chromium

co atribuindo a cada computador as características da sua sub-janela. Em cada computador fica a correr um daemon que espera comandos OpenGL. Durante a execução do programa, as chamadas OpenGL do servidor são redireccionadas para cada um dos nós do aglomerado. Cada computador recebendo as chamadas de visualização é assim capaz de renderizar a sua sub-janela de OpenGL de uma forma autónoma. Sendo assim, o conjunto dos doze projectores é capaz de apresentar uma única imagem. Efectuámos a experiência com varias aplicações de forma a analisar o desempenho desta solução. Por exemplo com se pode ver na Figura 6, instalámos a jogo Quake dado que é baseado na API OpenGL e obtivemos um desempenho médio de 35 imagens por segundo. Este bom desempenho só é possível devido a optimização do uso do OpenGL por parte do jogo, que tem cuidado em minimizar o número de chamadas OpenGL e simplificar as geometrias das cenas. Notámos igualmente, que é necessário um longo período de carregamento superior a cinco minutos e em fases de jogo em que exista mais geometria, o desempenho reduz drasticamente para valores da ordem dos 10-15 imagens por segundo. Dado que cada nó é autónomo, não existe sincronização das imagens provocando desfasamento entre as imagens durante a interacção. Outra desvantagem desta solução é não tirar proveito do aglomerado para maximizar o processamento, limitando o desempenho gráfico às capacidades do servidor. A Figura 7 mostra outro exemplo com o Chromium usando o visualizador de imagens aéreas de multi-resolução da NCSA[TsBlaster], onde o desempenho é igualmente afectado devido ao carregamento de texturas.

5.2 Sistemas OPENSG e OpenSceneGraph

Ao contrário do Chromium, existem outras soluções tais como o [OpenSG] e [OpenSceneGraph] que consistem em oferecer um grafo de cena distribuído. Este tipo de solução é fácil de configurar mas necessita que a aplicação esteja desenvolvida baseada no grafo de cena da solução. O princípio de funcionamento baseia-se na duplicação dos nós dos grafo de cena em cada um dos computadores do aglomerado, sendo cada computador responsável por renderizar o grafo de cena usando a definição local da sua sub-janela. O servidor central limita-se a sincronizar e transmitir as manipulações e alterações do grafo de cena, reduzindo o tráfego na sub-rede.



Figura 7: TsBlaster (NSCA) com Chromium



Figura 8: Demo usando o JINX

Esta abordagem é escalável a modelos mais complexos (ver Figura 1 em baixo) que a solução anterior e possibilita que cada mosaico aproveite ao máximo as suas características gráficas. Para além desta vantagem é possível estender essas soluções de forma a oferecer sincronização da troca de imagem. Uma das desvantagens é o controlo da aplicação continua centralizado, não possibilitando o aproveitamento de cada um dos nós para outros processamentos a não ser gráfico.

Neste momento estão a ser desenvolvidas novas demonstrações nomeadamente um visualizador de imagem assim como um visualizador genérico de cenas tridimensionais oferecendo navegação baseada no OpenSG. Vários protótipos do grupo de IMMI do INESC-ID estão a ser adaptados usando a solução OpenSG nomeadamente o sistema GIDES [Jorge03] e aplicação Blobmaker [Araujo03]. O OpenSG foi a ferramenta escolhida para o desenvolvimento de futuras aplicações interactivas no âmbito da nossa sala.

5.3 Sistemas alternativos para CAVES

Foram testadas várias alternativas nomeadamente pacotes de distribuição de visualização para sistemas imersivos CAVE e realidade virtual, tais como a ferramenta SYZYGY [Schaeffer03] e JINX [Soares04] (Figura 8). Tal como os sistemas anteriores, estas soluções oferecem mecanismos de sincronização e distribuição assim como bibliotecas gráficas de gestão de grafo de cena e objectos baseados na API OpenGL. Os paradigmas seguidos são muito semelhantes ao anterior, sendo mais interessante o OpenSG e OpenSceneGraph dado que estas ferramentas não se limitam a visualização mas oferecem outras funcionalidades para o desenvolvimento de aplicações.

6. INTERACÇÃO, APLICAÇÕES EXISTENTES

Para além do efeito de imersão atingido com o ecrã de larga escala, a sala do LEME pretende também oferecer ao utilizador um ambiente inteligente, sendo este objectivo atingido através de diversos mecanismos e modalidades. Os dispositivos tradicionais (WIMP) pressupõem a utilização individual e a posição fixa do indivíduo, para além de não oferecerem a naturalidade de interacção requerida. Deve ser possível ao utilizador andar pela sala enquanto interage com o sistema e visualiza os resultados no ecrã. Esta interacção deve-se aproximar da realizada

entre humanos e, devem, portanto, ser estudados todas as formas naturais de expressão do ser humano. Para esse efeito a sala encontra-se equipada com câmaras, colunas, microfones, aparelho de electromiografia, transmissores ultra-som e contempla a instalação de sensores de pressão, num futuro próximo. As câmaras serão utilizadas para tracking de movimentos para permitir ao utilizador a interacção através de gestos. Para detectar a posição dos utilizadores na sala, encontra-se em desenvolvimento um sistema baseado em transmissão de ultra-sons. Neste sistema, os utilizadores têm que ser equipados com um mini-transmissor que envia sincronamente sinais para um conjunto de receptores, estrategicamente colocados na sala. Através do tempo de transmissão para quatro receptores, é possível descobrir a posição de um utilizador, sendo o sistema escalável a mais utilizadores através do uso de janelas de tempo diferentes para cada um. A alternativa contemplada é através dos sensores de pressão que servem este objectivo bem como outros bastante interessantes. Os microfones adquiridos serão utilizados para interacção através da Fala. Esta comunicação é complementada por um sistema de reconhecimento de voz, bem como pelo sistema de som e aplicações TTS (Síntese de Fala). Para controlar aplicações e o próprio ambiente, contemplamos também o uso de PDAs, que funcionarão como controlo remoto. Encontra-se já aplicada na sala uma interacção através de um aparelho de electromiografia. O controlo das aplicações é efectuado através da monitorização do sinal miográfico do utilizador e associação de acções a determinadas contracções voluntárias. É possível, por exemplo, controlar o ponteiro do rato através de activações musculares.

Para resolver o problema da integração de aplicações heterogéneas (ex: LINUX vs WINDOWS) foi utilizado um sistema denominado OSGA (Open Software Group Architecture) que se baseia no XmlBlaster e comunica através de mensagens. Esta será a solução adoptada para a integração das diferentes aplicações de interacção a desenvolver na sala.

7. CONCLUSÕES

O lançamento de um novo laboratório contendo um sistema de multi-projecção apresenta inúmeros desafios devido à complexidade e novidade dos vários elementos que o compõem. Neste artigo apresentámos os primeiros passos na construção do nosso laboratório, descrevendo os principais problemas relacionados com um sistema de projecção de larga escala. Abordámos o tema da calibração de projectores e do controlo de um sistema de visualização por um aglomerado de computadores. Apresentámos igualmente a instalação de ferramentas de visualização distribuída, assim como paradigmas de interacção para este tipo de dispositivo, passo essencial para o desenvolvimento de futuras aplicações.

AGRADECIMENTOS

Os autores agradecem à Hewlett-Packard e ao Engº Carlos Janicas o apoio dado na realização da LEMEWall. Este trabalho foi apoiado pela Comissão Europeia através do projecto IMPROVE (FP6-IST-2003-004785).

REFERÊNCIAS

- [Araujo03] De Araujo B. and Jorge J., BlobMaker: Free-form modelling with variational implicit surfaces, Proceedings of 12º EPCG, Porto, Portugal, Oct 2003
- [Brown02] M. S. Brown, et al "A practical and flexible tiled display system", IEEE Pacific Graphics, 2002
- [DMX] <http://dmx.sourceforge.net/>
- [Hereld00] M. Hereld, et al., Introduction to Building Projection-based Tiled Display Systems, IEEE CGA, pages 22-28, Vol. 20, no. 4, Spring/Summer 2000
- [Humphreys02] G. Humphreys et al.: Chromium: a stream-processing framework for interactive rendering on clusters. Proc. of ACM SIGGRAPH 2002.
- [IGD A4] HeyeWall A4 IGD, Germany, <http://www.igd.fhg.de/igd-a4/projects37.html.en>
- [Jorge03] J. Jorge, N. Silva, T. Cardoso, Gides++, Proceedings of 12º EPCG, Porto, Portugal, Oct 2003
- [Li99] K. Li and Y. Chen, Optical Blending for Multi-Projector Display Wall System, Proc. 12th Lasers and Electro-Optics Society 1999 IEEE Press, N.J., Nov. 1999.
- [Li00] K. Li et al., Building and Using A Scalable Display Wall System, IEEE Computer Graphics and Applications, vol. 20, no.4, pp. 29-37, July/August 2000
- [Majumder02] A. Majumder and R. Stevens. LAM: Luminance Attenuation Map for Photometric Uniformity Across Projection Based Displays. ACM VRST 2002
- [Majumder04] A. Majumder and R. Stevens. Color Non-Uniformity in Projection Based Displays: Analysis and Solutions. IEEE TVCG, Vol. 10, No. 2, 2004
- [Majumder05] A. Majumder and R. Stevens. Perceptual Photometric Seamlessness in Tiled Projection-Based Displays. ACM Transactions on Graphics, 24(1), Jan. 2005
- [OpenScenegraph] <http://www.openscenegraph.org>
- [OpenSG] <http://www.opensg.org/>
- [PennState] LionEyes Display Wall, PennState Univ., USA, <http://viz.aset.psu.edu/ga5in/DisplayWall.htm>
- [Raskar03] R. Raskar et al., iLamps: Geometrically aware and SelfConfiguring Projectors, ACM SIGGRAPH 2003.
- [Soares04] L. Soares, M. Zuffo: JINX: an X3D browser for VR immersive simulation based on clusters of commodity computers. Web3D 2004
- [Schaeffer03] B. Schaeffer, C. Goudeseune, Syzygy: Native PC Cluster VR, Proceedings of the IEEE VR 2003, March 2003. <http://www.isl.uiuc.edu/syzygy.htm>
- [Surati99] R. Surati, Scalable Self-Calibrating Display Technology for Seamless Large-Scale Displays. Ph.D. thesis, Massachusetts Institute of Technology, 1999
- [Synergy] <http://synergy2.sourceforge.net/index.html>
- [TsBlaster03] <http://brighton.ncsa.uiuc.edu/~prajlich/>

Visual Syntax Analysis for Calligraphic Interfaces

Joaquim A. Jorge, Manuel J. Fonseca

INESC-ID/IST/UTL
R. Alves Redol, 9, 1000-029 Lisboa
jaj@inesc-id.pt, mjf@inesc-id.pt

Filipe M Garcia Pereira

IST/UTL
Lisboa
fmgp@mega.ist.utl.pt

Abstract

This paper presents an approach to analyse sketches of user interfaces by means of Visual Languages. To this end we use a shape recognizer, CALI [Fonseca00], and build on the ideas presented in JavaSketchIt [Caetano02] where the parsing is done by recognizing pairs of sketched symbols using a restricted visual syntax. The present paper presents our ongoing work on Visual Language Analysis to expand the possibilities afforded by JavaSketchIt in a more flexible manner. Our approach provides more flexibility with the ability to describe an increasing number of visual symbols, and making use of more complex productions which wouldn't be so simple to handle with previous pattern recognition techniques employed in JavaSketchIt, such as recursive productions and ambiguous arrangements involving more than two symbols.

Keywords

Calligraphic Interfaces, Visual Parser, Visual Syntax Parsing, Visual Grammar.

1. INTRODUCTION

A desire common to developers, programmers and especially interface designers, is to pick up those hand drawn sketches and drafts and magically create the interface they've been so carefully designing. However, the majority of commercially available tools do not support this. To overcome these problems we are working on calligraphic interfaces. These focus on sketch and gesture recognition and thus allow users to sketch the outline of user interfaces on a tablet PC, instead of entering a sequence of commands and menu selections.

In this paper we present an approach based on parsing sketches using visual grammars. Our technique draws on earlier research work, which it attempts to generalize. After a brief overview of related research we describe our approach from a few selected examples.

2. RELATED WORK

Until recently, there have been different approaches to specify visual syntax and parse visual sentences into machine symbols. One approach is DENIM [Lin01] (an evolution from SILK [Landay95]) to simulate interfaces from sketches in a storyboard environment. Designers are able to sketch low-fidelity prototypes and dynamically simulate their behaviour. However, there are no beautifications or elaborate translation steps in the system.

JavaSketchIt [Caetano02] provides a framework to bridge the gap between conceptual mock-ups and functional prototypes. Shapes are recognized using a lexicon provided by CALI [Fonseca00]. User interface widgets

can be recognized by combining primitive shapes using spatial relations and geometric attributes. These constitute a visual language that cannot be expanded since it is embedded in the application code.

SketchiXML [Coyette04] is another system based on JavaSketchIt and aimed at user interface prototyping. SketchiXML uses a multi-agent platform to perform the different tasks associated with parsing. Individual tasks are performed by distinct agents. One agent recognizes input scribbles, while others support removing redundancy and ambiguity. It is also possible to include other agents to perform extra tasks such as beautifying and aligning recognized widgets, as well as suggesting improvements to layout and design.

3. OUR APPROACH

To extend JavaSketchIt, we aim at developing a method for visual syntax analysis which should be both simple and general enough to allow users to define their own visual languages. Such visual languages are to be specified by means of a grammar using a set basic symbols and composition rules, using a textual language such as XML which should be easy to modify. While JavaSketchIt was a first attempt at this, the limitations inherent to the programming language used (Scheme) which resulted in syntax rules becoming embedded in the program code and thus difficult to modify and extend. Moreover, *widgets* (production rules) in JavaSketchIt could only rewrite at most groups of two symbols. Our aim is to provide a mechanism both strong and general enough to allow users to define productions with any

number of symbols. Also it should be possible to include recursive productions by repeating on the right-hand-side of a production the same symbol specified in the left hand-side.

To specify widgets using sketches we need to recognize sets of strokes as geometrical entities and then to associate these entities using a visual grammar. The terminal alphabet allowed for the possible visual languages correspond to the scribbles recognized by CALI as shown in Figure 1.

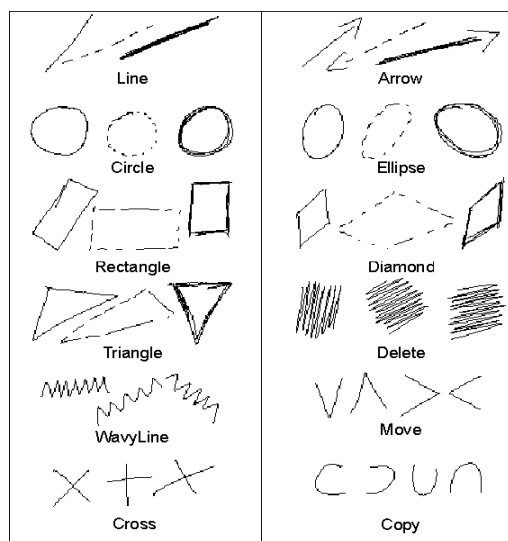


Figure 1: Scribbles recognized by CALI[Fonseca00]

Using this vocabulary, we define a set of rules based on spatial relations applied to constituent symbols. Remember from Visual Languages that it is spatial and topological relations that specify constraints rather than sequence [Jorge94]. Spatial constraints are defined either as unary or binary. Unary rules are fuzzy predicates that assert conditions on individual items such as `isHorizontal(Symbol)`. Binary rules are used to denote spatial relationships involving two visual elements. One such example is the binary fuzzy spatial relation predicate `isInside(Symbol1, Symbol2)` which allows the parser to check whether `Symbol1` is Inside `Symbol2`. Fuzzy predicates [Jorge94] differ from regular Boolean predicates in that they return a *degree of likelihood* between 0 and 1, rather than a crisp value of either true or false depending on whether the arguments satisfy (or not) the predicate.

By working out a simple example, these ideas can be better understood. Let's begin by providing a sample grammar definition for a simple widget (`TextField`), which is depicted by an horizontal line inside a rectangle. We use XML to specify grammars, as this provides a very flexible and easy to understand textual notation.

```
<widget type="TextField">
  <symbol type="Line" id="1"/>
  <symbol type="Rectangle" id="2"/>
```

```
<rule name="isHorizontal" arg1="1"/>
<rule name="isInside" arg1="1" arg2="2"/>
</widget>
```

In the example above, **widget** specifies a production in our grammar. Visual tokens correspond to `symbol` markers. Each **symbol** recognized by CALI will have a **type** assigned to it. We use positional identifiers to associate each symbol with arguments to constraints or **rules**. In principle any number of symbols or rules in a production.

Figure 2 shows the architecture of our system. The input is provided by strokes entered with a tablet/stylus combination. We use CALI to recognize sets of temporally contiguous strokes (scribbles) which are classified as one of the shapes shown in Figure 1.

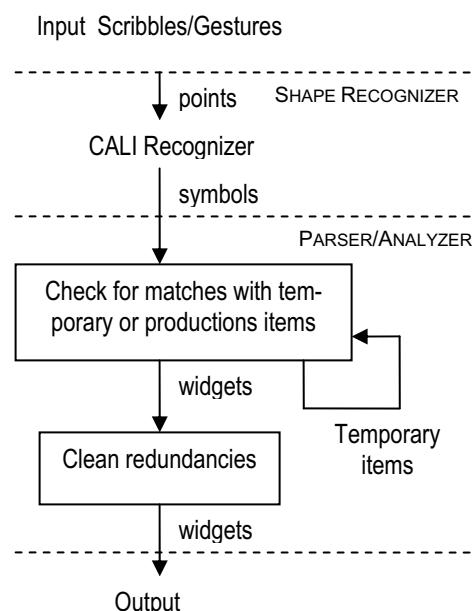


Figure 2: System architecture

Recognized scribbles are sent to a bottom-up parser. Since the order of constituents in a production is irrelevant, we need to search for items to associate using spatial relations and to use a set of heuristics to handle ambiguous productions. Our parser works by constructing partial items. These result from associating productions in the specified grammar with tokens captured by the recognizer. Incomplete *parse items* are stored in a Temporary Widgets List (TWL). When a new symbol is recognized by CALI, it gets checked against all elements in the TWL, generating new items as partial matches occur. New tokens are also checked against candidate productions in the grammar and may yield complete or incomplete items. A complete item is an instance of a production in which all symbol slots are filled. In the current system, complete items correspond to widgets and get stored in the results list. In contrast, incomplete items still require one or more symbols. A final set of heuristics cleans up both the temporary (TWL) and results list

eliminating redundant items. The pseudo code below illustrates the way our parsing approach works:

```

Parse(scribble)
  newSym ← Recognize(scribble)
  for each item in TWL
    if item.match(newSym) then
      nItem ← item.reduce(newSym)
      if nItem.isComplete() then
        results.add(nItem)
      else
        TWL.add(nItem)
  for each prod in Grammar
    if prod.match(newSym, item) then
      nItem ← prod.reduce(newSym)
      if nItem.isComplete() then
        results.add(nItem)
      else
        TWL.add(nItem)
CleanRedundantWidgets()

```

These ideas can be better understood by looking at a simple example. Consider the production `TextField` previously described. Figure 3 shows a possible sketch of a `TextField`.

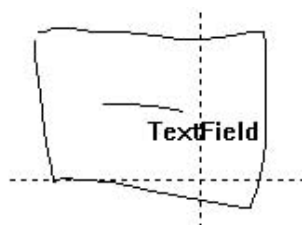


Figure 3: Sketch of a TextField widget (dashed lines are provided for context)

There are two different ways to draw this widget: either we can draw first the `Line` and then the `Rectangle` or the opposite sequence. For simplicity, let us begin with the `Rectangle` as show in Figure 4. The visual token recognized by CALI gets matched against all items in TWL. There is one incomplete `TextField` widget with a missing `Line` slot.

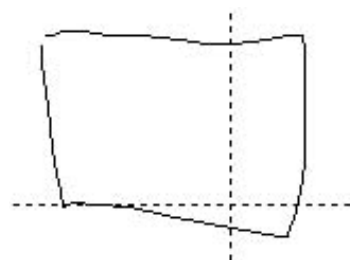


Figure 4: A Rectangle scribble

After the scribble is recognized by CALI as a `Rectangle`, we look in TWL for matching items. However the TWL is initially empty. We then check the Grammar for new entries. We find a matching `TextField` widget and generate the corresponding temporary item. Since the item is not complete (we need an horizontal line) we add it to TWL.

Finally, we draw a `Line` inside the `Rectangle` as shown in Figure 5. After it gets recognized by CALI we check the resulting shape against all items in the TWL. It matches the `TextField` item described above.

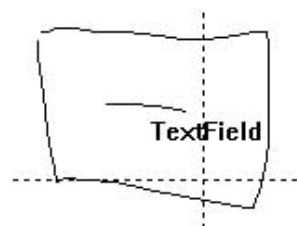


Figure 5: Sequence of two strokes

Since the temporary item thus found verifies both rules as checked by `item.match()`, i.e. `Line` is horizontal and is inside the `Rectangle` symbol in `item`, we reduce a new item. This item is complete. Thus, it gets added to `results` list, meaning that we have recognized a `TextField` widget. `CleanRedundantWidgets()` gets called afterwards to check whether there are temporary elements in TWL that are no longer needed. It finds the incomplete `TextField` item and checks this against the contents of `results`. Since the incomplete item matches a completed parse item in `results`, it gets deleted from TWL.

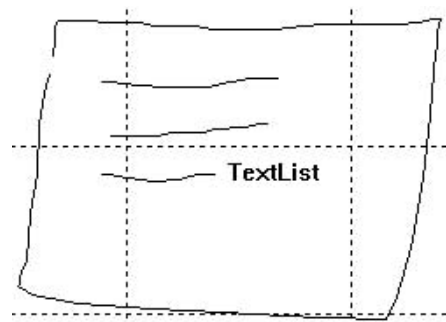


Figure 6: The TextList widget

Let's examine a more complex example, using a longer production rule involving also lines and rectangles. The visual representation for the widget `TextList` is shown in Figure 6. The grammar fragment below shows two productions one describing a simple `Text` widget (an horizontal line) and the other a `TextList` widget (three horizontal lines inside a rectangle):

```
<widget type="Text">
  <symbol type="Line" id="1"/>
  <rule name="isHorizontal" arg1="1"/>
</widget>
<widget type="TextList">
  <symbol type="Line" id="1"/>
  <symbol type="Line" id="2"/>
  <symbol type="Line" id="3"/>
  <symbol type="Rectangle" id="4"/>
  <rule name="isHorizontal" arg1="1"/>
  <rule name="isHorizontal" arg1="2"/>
  <rule name="isHorizontal" arg1="3"/>
  <rule name="isInside" arg1="1" arg2="4"/>
  <rule name="isInside" arg1="2" arg2="4"/>
  <rule name="isInside" arg1="3" arg2="4"/>
</widget>
```

We describe how the parser handles the input leading to the desired widget. Focusing on one of the possible permutations, we draw the following elements for this example in the sequence `Line(1)`, `Line(2)`, `Line(3)` and `Rectangle` as scribbles recognized by CALI.

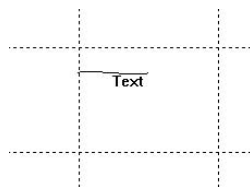


Figure 7: A `Text` widget parsed from the input of the horizontal "Line"

Figures 7 to 10 illustrate this sequence. Let us start by looking at `Line(1)`. At this step the parser checks to see if `Line(1)` is a valid entry for any of the temporary generated symbols and then the production rules in the grammar. Given that it matches all the input constraints for a `Text` widget, it is recognized as such, even though it may originate more temporary items to be recognized as more input is provided.

At this stage the predicate that has been asserted so far is `isHorizontal(Line(1))` which causes the recognition of the target widget. With this entry, there isn't any visible progress: the temporary data grow, as well as the previous temporary elements gain more and more information leading them closer to form a complete widget. Figure 8, shows the

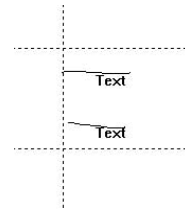


Figure 8: Two `Text` widgets parsed from input of both horizontal `Line` symbols

Figure 8 shows the state after two lines are drawn. Again what is seen and is possible to evaluate given the last input is still `isHorizontal(Line(1))` on the target widget, even though now we gathered two symbols already.

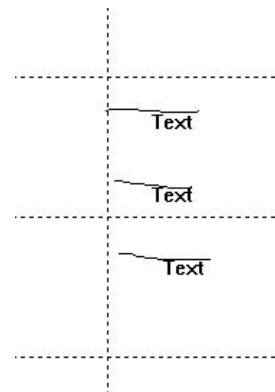


Figure 9: Three `Text` widgets parsed from input of all horizontal `Line` symbols

With this entry, again, there isn't really any visible progress, though the temporary data grow as temporary objects gather enough information to allow the next entry to generate the desired widget:

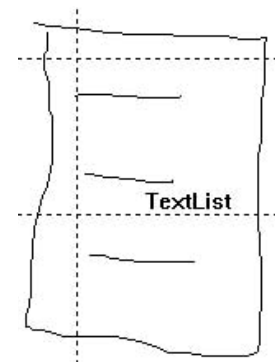


Figure 10: Three `Text` widgets parsed from input of all horizontal `Line` symbols and a rectangle

Figure 10 shows the complete input. The result is achieved when the `Rectangle` is provided to the temporary object list for evaluation. The parser checks that this is a valid entry and most important, it validates the missing rules that have been on hold which are `isInside(Line, Rectangle)` for each `Line` previously

provided. We now have a complete item and can assert that a `TextList` was recognized.

4. HEURISTICS USED

Given the parsing techniques described above, the remaining issues are redundancy and ambiguity.

The redundancy is most notably seen when generating temporary objects from grammar productions due to possible permutations for more complex widgets. An example occurs in the `TextList` where we have three lines which obey to the same exact rules and in the same position when parameters for binary rules. Associating items in pairs would lead to many redundant temporaries. The resulting idea was to identify such occurrences from the start and when checking for valid entries, identify only one of these pairs as relevant and not consider others.

Ambiguity arises when recognizing and defining which widgets are complete. This causes immediately the appearance of multiple widgets with a different number of pieces gathered among the input symbols. From the example in Figure 10, we could have identified three occurrences of `Text` and at least one occurrence of `TextList`. We solve this by picking the widget with the largest count of input pieces and keeping the others as alternative results.

5. FUTURE WORK

There are some obvious limitations to the current status of the project. One example is container widgets. For these the recognition process might cause collisions with the widgets inside them and generate undesired results, such as an exponential growth in the number of temporary items. This will be dealt with in the near future.

Another limitation comes from the apparent simplicity of the input language/grammar file, which is only a first stage, given that it still wouldn't allow defining the widget of Figure 10 in the way shown below:

```
<widget type="Text">
  <symbol type="Line" id="1"/>
  <rule name="isHorizontal" arg1="1"/>
</widget>
<widget type="TextList">
  <widget type="Text" id="1"/>
  <widget type="Text" id="2"/>
  <widget type="Text" id="3"/>
  <symbol type="Rectangle" id="4"/>
  <rule name="isInside" arg1="1" arg2="4"/>
  <rule name="isInside" arg1="2" arg2="4"/>
  <rule name="isInside" arg1="3" arg2="4"/>
</widget>
```

The main differences between this and the previous examples lie in that the latter production references *non-terminal* items (widgets) instead of symbols. Indeed currently we can only parse a conceptual two level hierarchy of syntax rules.

As a final remark, this approach still lacks the mechanisms to suggest when to align groups of widgets, or to allow the users to create their own symbols or even learn design patterns from the users. These problems require a top-down approach while what we have implemented so far is a bottom-up technique. We plan to address these issues in the near future.

6. REFERENCES

- [Caetano02] Caetano, A., Goulart, N., Fonseca, M. and Jorge, J.: JavaSketchIt: Issues in Sketching the Look of User Interfaces. In Proceedings of the 2002 AAAI Spring Symposium - Sketch Understanding, pages 9-14, Palo Alto, USA, 2002.
http://immi.inesc.pt/publication.php?publication_id=40
- [Coyette04] Adrien Coyette, Stéphane Faulkner, Manuel Kolp, Quentin Limbourg, Jean Vanderdonckt, "SketchiXML: Towards a Multi-Agent Design Tool for Sketching User Interfaces Based on UsiXML". In: Proc. of 3rd Int. Workshop on Task Models and Diagrams for user interface design [TAMODIA'2004](#) (Prague, November 15-16, 2004), Ph. Palanque, P. Slavik, M. Winckler (eds.), ACM Press, New York, 2004, pp. 75-82.
<http://www.isys.ucl.ac.be/bchi/publications/2004/Coyette-TAMODIA2004.pdf>
- [Fonseca00] M.Fonseca and J.Jorge. CALI: A Software Library for Calligraphic Interfaces. INESC-ID, available at <http://immi.inesc-id.pt/projects/cali/>, 2000.
<http://immi.inesc.pt/projects/cali/papers/9epcg2k.pdf>
- [Jorge94] Joaquim A Jorge, Parsing Adjacency Grammars for Calligraphic Interfaces, Phd Thesis, Rensselaer Polytechnic Institute, Troy, NY, 1994.
- [Lin01] James Lin, Mark W. Newman, Jason I. Hong, James A. Landay, "DENIM: An Informal Tool for Early Stage Web Site Design." Video poster in Extended Abstracts of Human Factors in Computing Systems: CHI 2001, Seattle, WA, March 31-April 5, 2001, pp. 205-206.
<<http://dub.washington.edu/projects/denim/pubs/denim-chi-2001-video.pdf>>
- [Landay95] James A. Landay and Brad A. Myers, "Interactive Sketching for the Early Stages of User Interface Design." In Proceedings of Human Factors in Computing Systems: CHI 95, Denver, CO, May 1995, pp. 43-50.
<<http://www.cs.berkeley.edu/~landay/research/publications/storyboard-tr/storyboard.html>>

Recuperação de Desenhos de Moldes Combinando Imagens e Esboços

Alfredo Ferreira Manuel J. Fonseca Joaquim A. Jorge

Departamento de Engenharia Informática

INESC-ID/IST/Universidade Técnica de Lisboa

R. Alves Redol, 9, 1000-029 Lisboa, Portugal

alfredo.ferreira@inesc-id.pt, mjf@inesc-id.pt, jorgej@acm.org

Resumo

Durante a concepção de um molde, os desenhadores consultam regularmente projectos existentes de modo a reaproveitar soluções anteriormente desenvolvidas. Desta forma conseguem reduzir o tempo de desenvolvimento de um molde. No entanto, as soluções existentes para encontrar o que desejam em desenhos anteriores são lentas e complexas.

Com o intuito de melhorar o processo de pesquisa de desenhos, desenvolvemos uma abordagem capaz de efectuar a indexação automática de desenhos de moldes e que permite a sua recuperação através esboços, imagens ou uma combinação de ambos. Neste artigo apresentamos essa abordagem, bem como os respectivos protótipos e a avaliação experimental a que estes foram sujeitos. Testes com utilizadores mostraram que o conceito apresentado agradou aos desenhadores e que os nossos algoritmos são simultaneamente eficientes e precisos.

Palavras-Chave

Interfaces Caligráficas, Recuperação Baseada em Conteúdos

1 Introdução

No decorrer das duas últimas décadas generalizou-se o uso de ferramentas de desenho assistido por computador (CAD) na indústria de moldes. Estas ferramentas facilitam a criação e edição de desenhos, dando origem a um grande número de desenhos de moldes em formato electrónico. Apesar de a reutilização de desenhos existentes otimizar o processo de desenvolvimento de moldes e ser uma técnica largamente utilizada, os métodos actualmente disponíveis para encontrar um desenho de uma peça ou molde são usualmente lentos e problemáticos, obrigando o utilizador a procurar em árvores de directorias ou a navegar em complexos menus de bibliotecas de componentes.

A maioria das soluções para este problema baseiam-se em bases de dados textuais [1, 3], as quais utilizam palavras chave e informação adicional, tal como a referência do projecto ou peça, datas de criação e alteração, nomes dos desenhadores ou descrições textuais do desenho. No entanto, estas soluções não são satisfatórias pois obrigam os utilizadores a conhecer em detalhe a meta-informação utilizada para caracterizar os desenhos. Mais, nestas abordagens a introdução desta informação nas bases de dados é usualmente realizada manualmente pelos desenhadores.

Neste artigo apresentamos uma abordagem que permite a classificação automática de desenhos técnicos de moldes e

sua recuperação baseada exclusivamente no seu conteúdo visual. Utilizando informação geométrica dos elementos dos desenhos e suas relações espaciais desenvolvemos um sistema que permite tirar partido da memória visual dos desenhadores e através de uma interface caligráfica para especificar as interrogações exploramos a sua capacidade para esboçar as peças ou moldes a encontrar. Adicionalmente, os utilizadores podem ainda utilizar desenhos impressos para especificar as interrogações, bastando para o efeito proceder à sua digitalização com um comum scanner de mesa, conseguindo inclusive esboçar sobre as versões digitalizadas dos desenhos, alterando-as.

2 Arquitectura

A arquitectura para a classificação automática e recuperação de desenhos técnicos de moldes está dividida em três partes: (1) classificação, (2) interrogação e (3) indexação e pesquisa. O módulo de classificação analisa os desenhos existentes em formato electrónico numa base de dados, produzindo descritores numéricos que descrevem as suas características geométricas e topológicas. O módulo de interrogação utiliza desenhos digitalizados, esboços ou uma combinação de ambos para produzir descritores semelhantes aos criados na classificação que são usados para pesquisar nas bases de dados. Por fim, o módulo de indexação e pesquisa é responsável por armazenar os descritores gerados durante

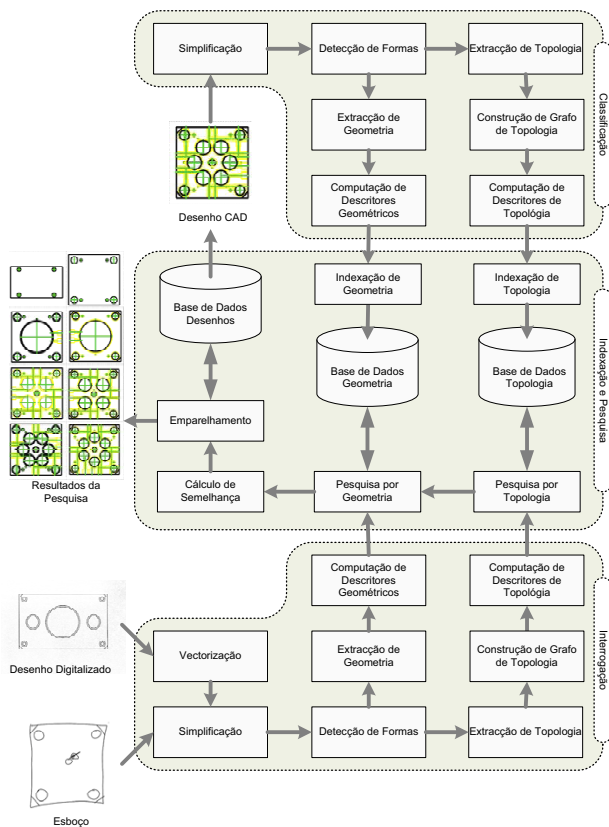


Figura 1. Arquitectura do sistema.

a classificação numa estrutura multidimensional e por fornecer mecanismos de pesquisa, assim como garantir o correcto mapeamento dos desenhos existentes com os respectivos descritores.

O primeiro passo na classificação é a simplificação do desenho. Este passo tem como principal objectivo identificar e remover características irrelevantes. De seguida são detectadas as formas que compõem o desenho simplificado. Para tal é utilizado o algoritmo de detecção de polígonos apresentado em [5], que aplica um método de identificação e remoção de intersecções em conjuntos de linhas para construir um grafo [2]. Sobre este é aplicado o algoritmo proposto por Horton [11] para determinar a base mínima de ciclos (MCB) de um grafo. A partir dessa MCB construímos o conjunto dos polígonos existentes no desenho. Recorrendo então à biblioteca CALI [10] para extrair a geometria dos polígonos resultantes e calculamos um conjunto de descritores geométricos que representam o desenho. Simultaneamente, é extraída informação topológica dos polígonos a partir das relações espaciais e constrói-se um grafo de topologia, a partir do qual são calculados descritores topológicos, conforme proposto em [7].

Para armazenar os descritores de geometria e topologia utilizamos uma estrutura de indexação multi-dimensional, a NB-Tree [9]. Sendo uma estrutura simples, a NB-Tree permite a indexação eficiente para pontos de dimensão elevada e variável, recorrendo à redução dimensional. Para o efeito esta mapeia os pontos multi-dimensionais numa linha uni-dimensional através do cálculo da sua norma

Euclideana. A NB-Tree suporta interrogações pontuais, por intervalos ou vizinhança, através da utilização do algoritmo *k-nearest neighbor* [12] (KNN) ¹.

Comparar uma interrogação com todos os desenhos da base de dados tem custos computacionais proibitivos, especialmente se considerarmos que estamos a trabalhar num ambiente real onde existe um elevado número de desenhos armazenados. Assim sendo, para tornar mais eficiente a pesquisa, dividimos o nosso processo em três passos distintos (Figura 2). O primeiro passo pesquisa por desenhos topologicamente semelhantes, funcionando como um primeiro filtro para evitar comparações geométricas desnecessárias. De seguida executamos uma pesquisa por geometria, comparando apenas os desenhos resultantes da pesquisa por topologia. Finalmente, aplicamos um algoritmo que calcula o grau de semelhança entre a interrogação e os desenhos recuperados da base de dados.

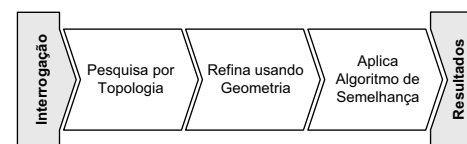


Figura 2. Diagrama de blocos da pesquisa.

O módulo de interrogação é bastante semelhante ao de classificação, tendo a mais um componente de vectorização que transforma em informação vectorial os desenhos digitalizados a usar como base para a pesquisa. Os restantes componentes de simplificação, detecção de formas, extração de características e computação de descritores são exactamente os mesmos em ambos os módulos. A diferença reside no facto que enquanto num caso são analisados desenhos em formato electrónico, noutro são rascunhos criados pelos utilizadores ou versões vectorizadas de desenhos digitalizados.

Quando se pretende utilizar um desenho digitalizado para especificar uma interrogação este deverá ser inicialmente convertido em vectores de forma a permitir a sua edição e análise tal como se de um rascunho se tratasse. Uma descrição detalhada do processo de vectorização de desenhos técnicos digitalizados pode ser encontrado em [6].

3 Protótipos

A nossa abordagem deu origem a dois protótipos distintos. Para proceder à classificação automática de desenhos desenvolvemos o *dbBuilder*, enquanto que para realizar interrogações à base de dados desenvolvemos o *SIBR*².

O protótipo *dbBuilder* (Figura 3) alivia os utilizadores da pesada tarefa de classificar manualmente os desenhos. Para tal, este protótipo implementa algoritmos de classificação e indexação automática. Como cada desenho pode levar alguns segundos ou muitos minutos a classificar, o *dbBuilder* permite que este processo seja executado em "batch", clas-

¹Dado um ponto P a pesquisar num espaço multi-dimensional, o algoritmo KNN devolve os K resultados mais próximos de acordo com uma métrica pré-estabelecida.

²*SIBR* é um acrónimo para *Sketch and Image Based Retrieval*.

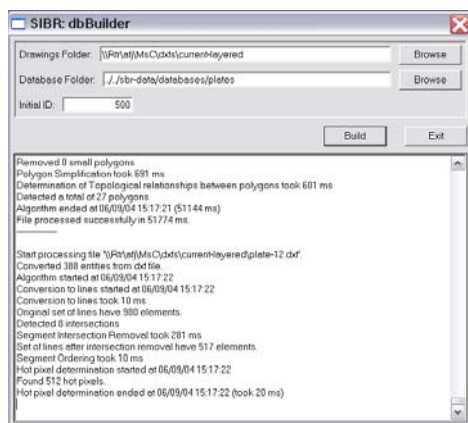


Figura 3. Ecrã do dbBuilder durante o processamento de desenhos.

sificando e indexando sequencialmente grupos de desenhos. Assim sendo, o utilizador apenas tem de indicar ao *dbBuilder* qual a localização dos desenhos a incluir na base de dados para que este os processe automaticamente, sem ser necessária qualquer intervenção por parte do utilizador.

O protótipo *SIBR* permite aos desenhadores recuperar os desenhos existentes na base de dados, previamente classificados pelo *dbBuilder*, recorrendo a esboços, imagens digitalizadas ou a uma combinação de ambos. Para tal, este protótipo possui uma interface caligráfica e incorpora o módulo de interrogação e parte do módulo de indexação e pesquisa. A interface caligráfica foi dividida em três áreas distintas, sendo esta distribuição produto de avaliações preliminares desta envolvendo desenhadores. A dominar espacialmente a interface temos uma grande área de esboço, onde são especificadas as interrogações. À direita desta existe uma área de resultados onde são mostrados os desenhos similares à última interrogação realizada, podendo exibir cinco resultados em simultâneo. Por fim, uma área de botões permite ao utilizador controlar a aplicação.

Utilizando a área de esboço, o desenhador pode esboçar uma representação bi-dimensional aproximada do molde da peça a procurar e efectuar uma interrogação na base de dados com esse esboço. Como resultado dessa interrogação o sistema apresenta um conjunto de desenhos similares ao procurado na respectiva área de resultados. Estes são ordenados de acordo com o seu grau de semelhança com a interrogação, sendo apresentado no topo o mais semelhante.

Na prática, este cenário é aplicável quando os desenhadores apenas têm uma memória visual da peça a procurar ou eventualmente têm à sua frente a peça propriamente dita. No entanto, se possuírem um desenho impresso em papel e necessitarem de encontrar a versão electrónica do mesmo podem digitalizar esse desenho usando um simples scanner e utilizar a imagem resultante para especificar uma interrogação. Para tal, basta um clique no botão de aquisição de imagem para que o sistema proceda à vectorização dessa imagem, sendo o resultado mostrado na área de esboço, onde poderá inclusivamente ser edi-

tado como se de um normal esboço se tratasse. Desta forma poderá apagar elementos usando apenas um gesto pré-definido ou seleccionar apenas parte do esboço a ter em conta na interrogação. A Figura 4 ilustra um caso em que um desenho em papel foi digitalizado e transformado num esboço, sobre o qual foram acrescentados alguns elementos e onde apenas parte deste irá ser utilizado para interrogar a base de dados.

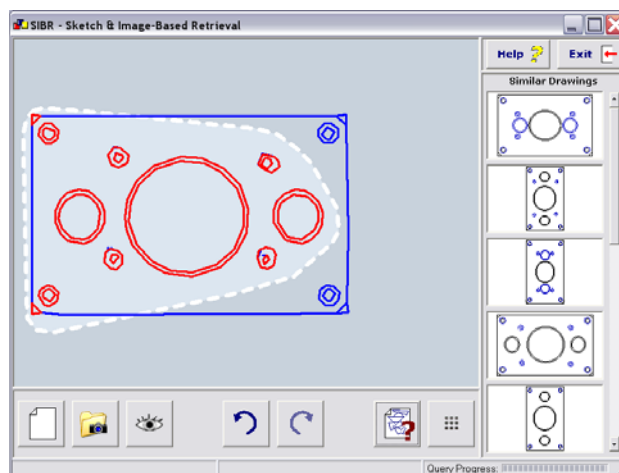


Figura 4. Ecrã da interface caligráfica após uma interrogação.

Estes protótipos foram utilizados para procedermos a uma avaliação experimental da nossa abordagem, a qual descrevemos sumariamente de seguida, mas que se encontra relatada em detalhe na tese de mestrado de Alfredo Ferreira [4].

4 Avaliação Experimental

Os principais objectivos da avaliação experimental consistiram na obtenção de *feedback* por parte de utilizadores chave, no caso presente desenhadores da indústria dos moldes, sobre as funcionalidades disponibilizadas pelo nosso sistema, assim como da sua interface caligráfica. Paralelamente, testámos também a qualidade dos resultados devolvidos pelos nossos algoritmos.

4.1 Descrição da Experiência

Esta experiência foi realizada no CENTIMFE³ e envolveu cinco desenhadores desta instituição e um desenhador da Aníbal Abrantes⁴. Todos eles possuem elevada experiência na utilização de ferramentas CAD e três já haviam experimentado anteriormente um sistema de recuperação de desenhos baseado em esboços, pois participaram em testes

³CENTIMFE é um centro tecnológico da indústria de moldes, que se posiciona como uma organização de interface que, através do suporte técnico, pelo desenvolvimento tecnológico e pela promoção de uma sistemática interacção entre empresas e a comunidade científica e tecnológica, pretende representar um importante valor acrescentado para a indústria.

⁴Aníbal Abrantes é uma empresa pioneira na indústria Portuguesa de moldes. Fundada em 1946, é actualmente conhecida como a "Universidade dos Moldes", tendo já produzido cerca de 8000 moldes para mais de cinquenta países.

preliminares efectuados durante o desenvolvimento dos protótipos aqui apresentados. Aos utilizadores inexperientes foi dada uma breve formação sobre como especificar as interrogações. Adicionalmente, efectuámos uma breve demonstração do sistema e das suas funcionalidades a todos os utilizadores, de modo a que estes conseguissem tirar pleno partido do protótipo.

Para minimizar a interferência destas experiências como dia-a-dia dos desenhadores, a avaliação foi organizada em sessões individuais. Por sua vez, cada sessão de teste foi dividida em quatro partes distintas. Na primeira parte, explicámos a experiência e apresentámos o protótipo aos utilizadores. De seguida, estes realizaram quatro interrogações simples para se familiarizarem com o sistema. Após isso, os utilizadores pesquisaram na base de dados vários desenhos técnicos de moldes, executando um conjunto de tarefas que lhes apresentámos. Por fim, responderam a um questionário, no qual tentámos identificar o seu perfil, as suas opiniões sobre o protótipo e a sua avaliação da interface. Tentámos ainda obter, de forma informal, as suas opiniões e eventuais sugestões.

Durante a fase de treino e durante a execução das tarefas os utilizadores foram encorajados a fazer comentários ("pensar alto") e a colocar questões aos observadores que os acompanhavam. Esta forma de interacção mostrou-se bastante proveitosa pois permitiu-nos recolher informação útil, não limitada a questões preparadas previamente.

Para executar as tarefas pedidas, era apresentado ao utilizador um desenho impresso, sendo-lhes pedido que esboçassem uma interrogação, a submetessem à base de dados e analisassem os resultados devolvidos pelo sistema. Caso o desenho procurado não aparecesse no conjunto de doze resultados apresentados pelo protótipo, o utilizador poderia alterar o esboço acrescentando-lhe novas forma, eliminando as existentes ou recomeçar de novo. No decorrer da experiência foram apresentados sequencialmente aos utilizadores os cinco desenhos mostrados na Figura 5, sendo os passos anteriormente referidos repetidos para cada um deles. Estes desenhos constavam na base de dados usada nesta experiência e que continha um total de 98 desenhos, sendo vinte de moldes reais e os restantes de peças ou outros desenhos mais simples.

Conforme lhes havia sido sugerido, os utilizadores esboçaram as interrogações desenhando as características que consideravam mais relevantes no desenho a pesquisar. No entanto quando estes tentavam representar um grande número de características mas não o conseguiam fazer com a precisão necessária, os resultados nem sempre eram os

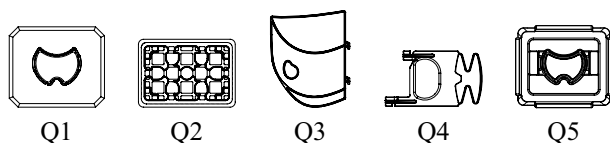


Figura 5. Desenhos técnicos apresentados como interrogações.

desejados. Os melhores resultados eram obtidos quando os utilizadores se centravam nas características principais, não se preocupando em demasia com detalhes. No entanto, mesmo estes esboços deveriam ser desenhados com alguma precisão.

Assim sendo, para desenhar uma interrogação que produzisse bons resultados era importante que o utilizador fosse capaz de identificar os elementos relevantes do desenho e de os esboçar com alguma precisão. Como durante experiências realizadas anteriormente [8] já havíamos concluído que desenhadores distintos identificavam os mesmos elementos como relevantes, os nossos algoritmos estão preparados para corresponder a essas escolhas.

Para avaliar os resultados destes testes monitorizámos um conjunto de parâmetros na fase de execução de tarefas. Um deles foi a posição em que o desenho procurado era apresentado na lista de resultados de uma interrogação bem sucedida. Outro, foi o número de interrogações necessárias até o utilizador encontrar o desenho pretendido. Medimos ainda outros parâmetros tal como a complexidade dos esboços, contabilizando o número de traços desenhados, ou a quantidade de redesenhos de raiz que os utilizadores sentiram necessidade de fazer. Como demos aos utilizadores liberdade para fazer sugestões e colocar questões durante o teste não seria útil medir o tempo gasto a definir as interrogações.

4.2 Análise dos Resultados

Ao analisar os resultados obtidos vimos que em mais de cinquenta por cento dos casos os desenhos pretendidos eram encontrados com apenas uma interrogação (Figura 6). Mais, praticamente todas as pesquisas bem sucedidas foram conseguidas com menos de três tentativas. Podemos assim afirmar que utilizando as interrogações e a base de dados descritas neste artigo, o nosso sistema foi capaz de identificar o desenho desejado em 90% dos casos, o que satisfaz plenamente os utilizadores, conforme as suas respostas aos questionários confirmam.

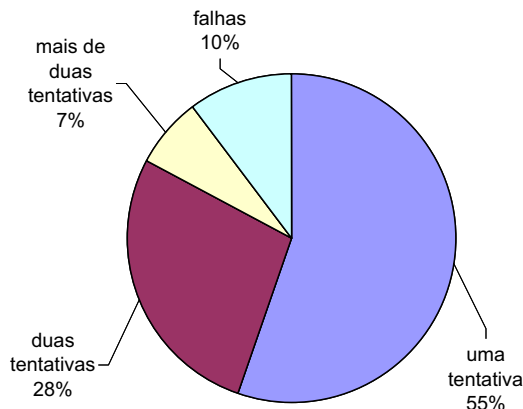


Figura 6. Tentativas antes de se encontrar o resultado correcto.

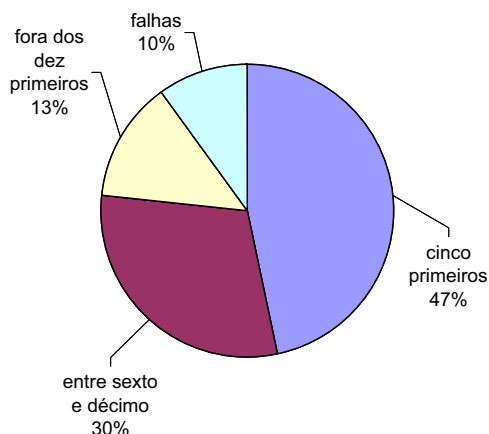


Figura 7. Distribuição dos resultados das interrogações.

Além disso, em cerca de 47% de todas as interrogações o desenho procurado encontrava-se nos cinco primeiros, conforme se pode ver na Figura 8. Estar nos cinco primeiros implica, no protótipo utilizado, que seja imediatamente visível, sem que o utilizador tenha de rolar a lista de resultados. Caso não consideremos os casos em que o sistema não devolve o desenho desejado na lista de resultados⁵ (falha), vemos que mais de metade das interrogações bem sucedidas mostram o desenho procurado entre os cinco primeiros.

Conforme referido anteriormente três utilizadores tiveram no momento da avaliação o seu primeiro contacto com um sistema de recuperação baseado em esboços, enquanto os restantes já haviam utilizado versões preliminares do protótipo. Aachamos então pertinente classificar os utilizadores em duas categorias distintas de acordo com a sua experiência. Tendo em conta esta classificação, analisámos os resultados obtidos pelas interrogações efectuadas e concluímos que os utilizadores experientes conseguem, em média, melhores resultados que os utilizadores noviços, conforme se pode verificar na Figura 7. No entanto, esta diferença dilui-se à medida que os utilizadores noviços vão ganhando experiência no sistema.

No cômputo geral, os resultados, comentários e sugestões que obtivemos na avaliação dos nossos protótipos mostraram que, apesar de algumas queixas no que refere a detalhes da interacção, os utilizadores mostraram-se bastante satisfeitos com a interface. Além disso os nossos algoritmos produziram resposta com boa precisão num tempo aceitável ao procurar por desenhos complexos na base de dados.

5 Conclusões e Trabalho Futuro

Neste artigo apresentámos uma abordagem para indexar e recuperar desenhos técnicos de moldes utilizando esboços, desenhos digitalizados ou uma combinação de ambos.

⁵Para estes testes o sistema foi preparado para devolver os doze desenhos mais próximos da interrogação sendo estes mostrados na lista de resultados ao utilizador.

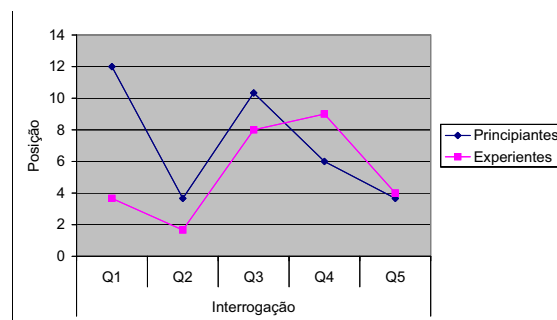


Figura 8. Posição do resultado esperado agrupado por tipo de utilizador.

Acreditamos que uma ferramenta desenvolvida sobre a arquitectura apresentada poderá reduzir o tempo de desenvolvimento de novos moldes, pois permite a re-utilização de soluções anteriores sem que o desenhador tenha de perder tempo a navegar em longas árvores de directorias ou complexos menus.

Para demonstrarmos a validade da nossa abordagem desenvolvemos dois protótipos funcionais. Enquanto um classifica de forma automática os desenhos existentes, o outro fornece ao utilizador uma interface caligráfica para especificar as interrogações. Estes protótipos foram submetidos a uma avaliação experimental na qual envolvemos desenhadores da indústria de moldes. A descrição destes testes e respectivos resultados são a principal contribuição do presente artigo face a anteriores publicações.

A partir da análise dos resultados dessa avaliação concluímos que este novo paradigma para recuperação de desenhos de moldes agradou aos utilizadores, os quais demonstraram bastante interesse em possuir este tipo de ferramenta no seu ambiente de trabalho. Mais, os algoritmos implementados provaram ser eficientes e precisos o suficiente para satisfazer as necessidades dos desenhadores. Infelizmente, devido a questões relacionadas com propriedade intelectual e industrial, não nos foi possível construir uma base de dados realmente grande, com largas centenas de desenhos de moldes. Actualmente a nossa base de dados contém cerca de uma centena de desenhos e apenas vinte deles são de moldes reais. Além disso e apesar do precioso contributo da equipa do CENTIMFE, a disponibilidade de desenhadores para participarem na avaliação do protótipo ficou aquém das nossas expectativas iniciais pois apenas conseguimos envolver na nossa experiência meia dúzia de utilizadores.

Apesar de tudo, sugestões e comentários feitos pelos utilizadores em conjunto com observações que fizemos durante a avaliação do protótipo vão permitir que sejam efectuados alguns melhoramentos nos algoritmos de classificação e pesquisa, bem como na interface propriamente dita. Assim, num futuro próximo esperamos testar uma evolução deste protótipo com uma base dados real, envolvendo dezenas de desenhadores, de modo a podermos afirmar com maior confiança que o sistema proposto tem um bom desempenho e precisão em ambiente de produção.

6 Agradecimentos

Gostaríamos de agradecer aos técnicos do CENTIMFE pelo valioso contributo que deram a este projecto. Este trabalho foi patrocinado em parte pela Comissão Europeia, projecto integrado do sexto programa quadro *Eurotooling 21*, IP 505901-5.

Referências

- [1] D. V. Bakergem. Image collections in the design studio. In *The Electronic Design Studio: Architectural Knowledge and Media in the Computer Age*, pages 261–272. MIT Press, 1990.
- [2] Ulrike Bartuschka, Kurt Mehlhorn, and Stefan Naher. A robust and efficient implementation of a sweep line algorithm for the straight line segment intersection. In *Proceedings of Workshop on Algorithm Engineering*, pages 124–135, Venice, Italy, September 1997.
- [3] M. Clayton and H. Wiesenhal. Enhancing the sketchbook. In *Proc. of the Association for Computer Aided Design in Architecture (ACADIA'91)*, Los Angeles, CA, 1991.
- [4] Alfredo Ferreira. Sketch and image based retrieval of technical drawings. Master's thesis, Instituto Superior Técnico / Universidade Técnica de Lisboa, 06 2005.
- [5] Alfredo Ferreira, Manuel J. Fonseca, and Joaquim A. Jorge. Polygon detection from a set of lines. In *Proceedings of 12º Encontro Português de Computação Gráfica (12th EPCG)*, pages 159–162, Porto, Portugal, October 2003.
- [6] Alfredo Ferreira, Manuel J. Fonseca, Joaquim A. Jorge, and Mário Ramalho. Mixing images and sketches for retrieving vector drawings. In *Proceedings of the 7th Eurographics Workshop on Multimedia*, Nanjing, China, October 2004.
- [7] Manuel J. Fonseca, Bruno Barroso, Pedro Ribeiro, and Joaquim A. Jorge. Retrieving vector graphics using sketches. In Andreas Butz, Antonio Krüger, and Patrick Olivier, editors, *Proceedings of the 4th International Symposium on Smart Graphics (SG'04)*, volume 3031 of *Lecture Notes in Computer Science*, pages 66–76. Springer-Verlag, May 2004.
- [8] Manuel J. Fonseca and Alfredo Ferreira. Technical drawing retrieval by sketches usability tests at centimfe. Technical Report SS-WP6-D13, INESC-ID, Lisbon, July 2005.
- [9] Manuel J. Fonseca and Joaquim A. Jorge. Indexing high-dimensional data for content-based retrieval in large databases. In *Proceedings of the 8th International Conference on Database Systems for Advanced Applications (DASFAA'03)*, pages 267–274, Kyoto, Japan, March 2003. IEEE Computer Society Press.
- [10] Manuel J. Fonseca, César Pimentel, and Joaquim A. Jorge. Cali: An online scribble recognizer for calligraphic interfaces. In *Proceedings of the 2002 AAAI Spring Symposium - Sketch Understanding*, pages 51–58, Palo Alto, USA, March 2002.
- [11] J.D.Horton. A polynomial-time algorithm to find the shortest cycle basis of a graph. *SIAM Journal on Computing*, 16(2):358–366, April 1987.
- [12] Nick Roussopoulos, Stephen Kelley, and Frédéric Vincent. Nearest neighbor queries. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 71–79. ACM Press, 1995.

Simplificação de Plantas de Arquitectura para Classificação Automática

António Simão Ricardo Ramião
Alfredo Ferreira Manuel J. Fonseca Joaquim A. Jorge
Departamento de Engenharia Informática
INESC-ID/IST/UTL

R. Alves Redol, 9, 1000-029 Lisboa, Portugal

crimsonzinho@sapo.pt, rmmr@netcabo.pt

alfredo.ferreira@inesc-id.pt, mjf@inesc-id.pt, jorgej@acm.org

Resumo

Durante a fase de criação de um novo projecto de arquitectura, normalmente os arquitectos e desenhadores recorrem a trabalhos antigos para procurar soluções já atingidas no passado. No entanto a procura de plantas antigas obriga a uma pesquisa demorada, na medida em que não existem mecanismos de pesquisa baseados no conteúdo. Os que existem obrigam os utilizadores a inserir um conjunto de informação adicional que posteriormente servirá para recuperar as plantas. A solução por nós proposta vai no sentido de extrair de forma automática informação sobre o conteúdo das plantas, para permitir a sua posterior recuperação usando apenas esboços ou plantas fornecidas pelo utilizador. Para que esta abordagem produza bons resultados, é necessário simplificar as plantas de modo a eliminar informação redundante. Assim, reduzimos a quantidade de informação a catalogar, guardar e pesquisar. Neste artigo, descrevemos o processo desenvolvido para simplificar plantas de arquitectura para que estas possam ser submetidas a um processo de catalogação automático. Apresentamos ainda o modo como este mecanismo de simplificação se integra numa plataforma mais genérica de recuperação de informação multimédia.

Palavras-Chave

Simplificação de Plantas, Desenhos Vectoriais, Classificação Automática, Recuperação usando Esboços

1. Introdução

Estudos [11, 5] referem que a utilização de bibliotecas de casos antigos é importante para ajudar o arquitecto a identificar características a incluir ou problemas a evitar em novos desenhos. Por outro lado, em algumas firmas de arquitectura, constitui prática comum a cópia de desenhos entre colegas de equipa, para posterior desenvolvimento [4]. Em conversas informais com alguns arquitectos constatamos que durante a fase de criação de novos trabalhos, estes recorrem com alguma frequência a desenhos antigos para retirarem ideias ou soluções de problemas específicos já resolvidos.

Tradicionalmente o processo de localização destes trabalhos é moroso, complexo e pouco fiável, exigindo uma inspecção visual exaustiva. Infelizmente a popularização dos sistemas de CAD, ao facilitar a criação e edição de novos desenhos, agudiza este problema na medida em que o número de trabalhos produzidos aumenta, sem que existam mecanismos de busca apropriados a esta tarefa. Mesmo os arquitectos que armazenam os seus trabalhos e desenhos em formato electrónico, usando um sistema de CAD, continuam a fazer a pesquisa usando a versão em papel, e só

depois de localizarem o desenho pretendido e identificarem o ficheiro correspondente, utilizam a versão electrónica deste. Infelizmente, a única referência que os arquitectos têm à sua disposição para lhes dar alguma informação sobre o conteúdo de um determinado desenho é o nome do ficheiro.

Algumas das soluções propostas [1, 3] para este problema utilizam bases de dados textuais para organizar o conteúdo dos desenhos. Normalmente os desenhos são classificados usando palavras chave e outra informação como nome do arquitecto, estilo, data e uma descrição textual. Porém, as soluções baseadas em interrogações textuais não resolvem o problema de forma satisfatória, pois obrigam os arquitectos a conhecerem em detalhe a meta-informação usada para caracterizar os desenhos e necessitam de intervenção humana para gerar a classificação destes.

Mark Gross e Ellen Do exploraram a utilização de diagramas para indexar bases de dados de desenhos de arquitectura [12, 13]. A cada desenho era associado um diagrama, desenhado manualmente, que depois era comparado com outro diagrama, desenhado pelo utilizador durante a fase de pesquisa. Embora este sistema seja funcional para con-

juntos pequenos de imagens, torna-se impraticável para colecções maiores, uma vez que não suporta a classificação automática destas.

Em oposição à organização textual e à classificação manual, utilizamos uma classificação visual baseada em topologia e geometria, que consideramos mais adequada a este tipo de problemas, na medida em que utiliza a memória visual evidenciada pelos arquitectos e explora a sua capacidade para fazer rascunhos, como mecanismo de interrogação. No entanto, a utilização de soluções deste tipo pressupõe a existência de mecanismos capazes de extrair de forma automática a informação que descreve o conteúdo dos desenhos. Neste artigo apresentamos um conjunto de técnicas para simplificar plantas de arquitectura, em formato electrónico (DXF), de modo a facilitar a extracção, descrição e comparação de plantas de arquitectura com interrogações sob a forma de esboços. A nossa abordagem simplifica o desenho de modo a extrair apenas a topologia e a geometria das divisões que constituem uma planta. Toda a informação restante (acessórios, portas, janelas, móveis, etc.) são retirados de modo a reduzir a complexidade do desenho. No resto do artigo, descrevemos os vários passos de simplificação de plantas de arquitectura, mostramos como este conjunto de técnicas se integram numa plataforma de recuperação de informação multimédia e finalmente apresentamos conclusões e trabalho futuro.

2. Simplificação

A simplificação de uma planta de arquitectura começa pela identificação da camada onde se encontram as paredes. Deste modo, grande parte da informação irrelevante para a descrição do conteúdo é descartada. De seguida, reduzimos a representação das paredes a apenas uma linha. Devido a esta redução, algumas paredes ficam desalinhadas, o que obriga a mais um passo de alinhamento. Finalmente, procuramos fechar todas as paredes, de modo a definir as divisões. A Figura 1 apresenta os vários passos da simplificação, que descreveremos com mais detalhe nas subsecções seguintes.

2.1. Pré-Requisitos das Plantas

Para que a simplificação ocorra sem problemas temos que garantir algumas condições iniciais na planta que é fornecida. O ficheiro DXF que contém a planta deve estar dividido em várias camadas, de modo a agrupar em cada uma delas um tipo específico de elemento. No nosso caso, a situação mais desejada, é o isolamento das paredes de tudo o resto. Caso esta última condição não se verifique, e existam outros elementos na camada das paredes, então tudo o que estiver na camada das paredes tem que ter uma cor diferente. Finalmente, os algoritmos desenvolvidos assu-

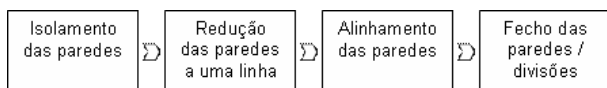


Figura 1. Passos da simplificação.

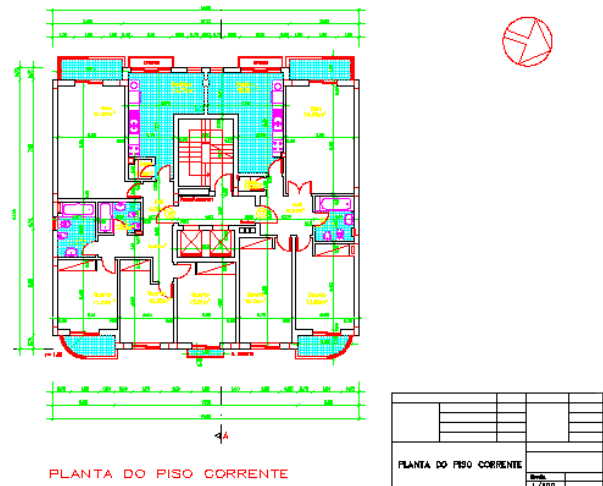


Figura 2. Planta inicial com todas as camadas.

mem que todas as paredes são representadas com duas linhas paralelas, como é comum em desenhos técnicos. A maior parte destes pré-requisitos seriam desnecessários, se os arquitectos ou desenhadores cumprissem as normas internacionais existentes.

2.2. Isolamento das Paredes

O primeiro passo na simplificação de uma planta de arquitectura é identificar a camada onde se encontram as paredes. Começamos por comparar os nomes de cada uma das camadas do ficheiro com o que está definido no standard ISO DIS 13567 [2]. Caso o ficheiro respeite esta norma, então a detecção da camada de paredes é directa. Por outro lado, como algumas empresas definem a sua própria norma para a atribuição de nomes às camadas, é possível especificar qual o nome da camada que contém as paredes.

Caso não seja possível identificar a camada que contém as paredes apenas pelo seu nome, então aplicamos os nossos algoritmos. Primeiro, retirámos as linhas mais pequenas, eliminando informação irrelevante, como por exemplo os limites das cotas e zonas sombreadas. Por fim, contamos o número de linhas em cada camada, escolhendo a que tem mais linhas como sendo a das paredes.

Uma vez identificada a camada que contém as paredes vamos verificar se todos os elementos são realmente paredes. Para isso, construímos um histograma de cor, de modo a identificar a cor predominante. Todos os elementos que não tenham a cor predominante são descartados. Após este passo as paredes estão completamente isolados do resto da planta. A Figura 3 apresenta o resultado da aplicação destes algoritmos sobre a planta da Figura 2.

2.3. Redução das Paredes a uma Linha

Tipicamente, as paredes das plantas de arquitectura são desenhadas com duas linhas de modo a indicar a sua espessura. No entanto, como essa informação não é importante para a detecção, análise e descrição da topologia das divisões, vamos procurar eliminá-la. O processo utilizado,

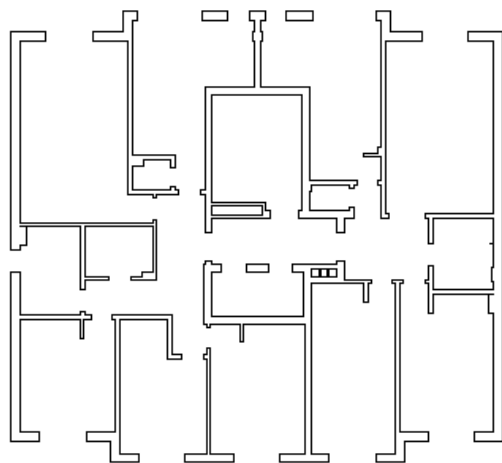


Figura 3. Planta apenas com as paredes.

começa com uma linha e procura linhas paralelas com alguma zona em comum e a uma distância relativamente próxima (espessura máxima de uma parede). Quando se detectam duas linhas nestas condições, é criada uma nova linha de comprimento igual à parte comum a ambas e situada no meio delas. Este passo é necessário pois existem linhas que podem ter várias linhas paralelas de dimensão menor. Se a linha que está a ser processada for menor que a encontrada, então o algoritmo passa à linha seguinte. Caso contrário, marcamos a linha encontrada como removida, o que implica que já não será tratada, e continuamos a iterar com a linha inicial.

No final deste passo temos uma lista de pontos que representa as extremidades das novas linhas e que será utilizada nos passos seguintes. A Figura 4 mostra o resultado da aplicação deste algoritmo.

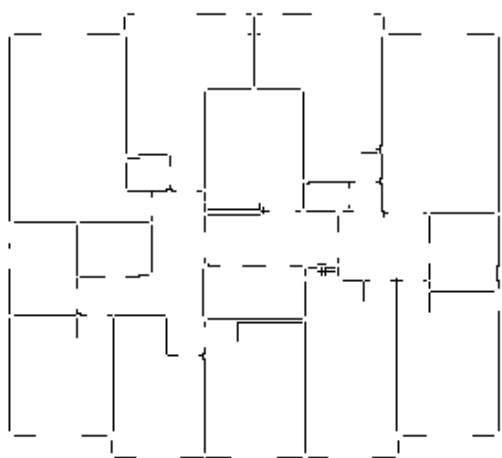


Figura 4. Paredes representadas apenas com uma linha.

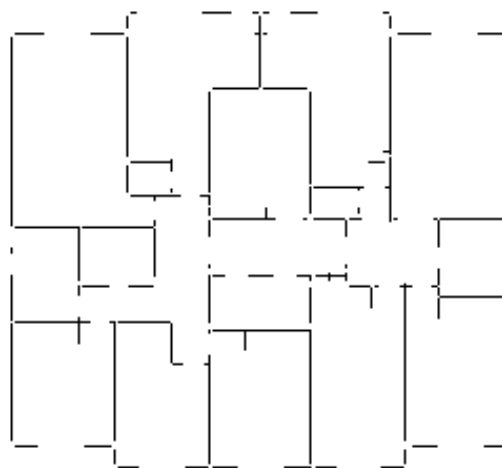


Figura 5. Paredes depois de alinhadas.

2.4. Alinhamento das Paredes

Como as paredes de uma planta não são todas da mesma espessura e também devido à própria morfologia de uma planta, as novas linhas poderão ter alinhamentos diferentes, como se pode ver na Figura 4. De modo a facilitar a tarefa de ligar as paredes entre si, aplicamos mais um passo que permite colocar as paredes no mesmo alinhamento. Para cada linha verificamos se existe uma linha paralela com um desfasamento inferior ao valor máximo da espessura de uma parede e que se situe num raio de proximidade de metade desse valor. Deste modo apenas fazemos alinhamentos locais.

Quando existem sobreposições de linhas, (que nós assumimos que são apenas do tipo em que uma linha está contida na outra), eliminamos a que ficaria contida. Na Figura 5 mostramos a planta com as linhas alinhadas verticalmente e horizontalmente.

2.5. Fecho das Paredes

O último passo da simplificação está dividido em duas etapas. Na primeira, procuramos ligar paredes próximas que no processo de passagem a linhas simples ficaram separadas devido às diferenças de tamanhos das linhas que as formavam. Começamos por procurar linhas que tenham um extremo não ligado. Para cada um destes pontos procuramos qual o ponto que se encontra mais perto dentro de um raio pré-definido (é necessário ter em atenção que esse ponto mais próximo não poderá ser a outra extremidade da linha). Encontrado o ponto mais próximo criamos uma nova linha entre os dois, ligando assim duas paredes. Ambos os pontos são marcados como não livres e passamos à próxima iteração.

A segunda etapa serve para ligar apenas linhas que sejam paralelas, se encontrem no mesmo alinhamento e cujos extremos mais próximos não distem de um valor pré-definido (tipicamente a largura de uma parede). Para cada linha que tenha pelo menos um dos extremos marcados como livre,

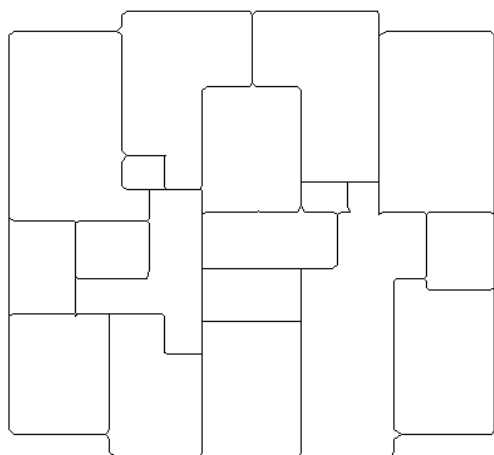


Figura 6. Planta simplificada depois de fechar as linhas.

percorremos a lista de linhas à procura de outra nas mesmas condições e cuja distância à primeira seja maior que o raio definido para as ligações curtas (para não ligarmos novamente as linhas próximas) e ligamos as duas. É ainda necessário que a nova linha não se sobreponha a outra já existente. Desta forma efectuamos ligações mais longas correspondentes aos locais onde existiam portas ou janelas.

Finalmente, aplicamos mais um passo para eliminar todas as linhas que apresentem extremos soltos. O resultado final é uma planta onde as paredes são representadas usando apenas uma linha e onde todas as divisões estão fechadas (ver Figura 6).

3. Integração com Plataforma de Recuperação

Todo este processo de simplificação de plantas de arquitectura é realizado para reduzir a quantidade de informação a processar e armazenar durante a análise de conteúdo. A partir das plantas, extraímos informação sobre topologia (arranjo espacial das divisões) e geometria (proporção das divisões).

A topologia das divisões é guardada sob a forma de um grafo de topologia, onde cada nó representa uma divisão e cada ligação horizontal significa adjacência. A Figura 7 apresenta o grafo de topologia para parte da planta da Figura 6. A partir destes grafos de topologia aplica-se uma técnica de descrição por níveis, que permite descrever a mesma planta usando diferentes níveis de detalhe, e também descrever partes da planta. Deste modo o utilizador durante a pesquisa pode usar interrogações simples para procurar plantas complexas. Uma descrição mais detalhada deste tipo de grafos de topologia assim como da descrição por níveis pode ser encontrada em [9, 6]. Os grafos de topologia são posteriormente convertidos em descritores vectoriais através do cálculo dos valores próprios das matrizes de adjacência dos grafos. Estes descritores são depois inseridos numa estrutura de indexação multidimensional

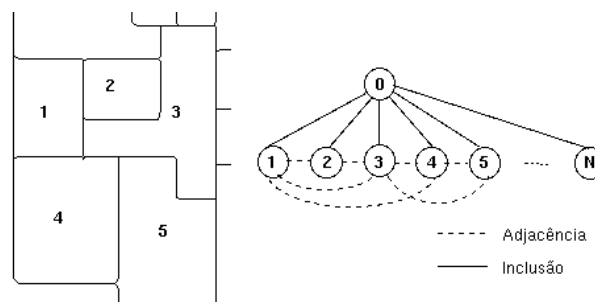


Figura 7. Grafo de topologia (parcial).

dimensional que permite uma pesquisa eficiente dos vizinhos mais próximos. Assim, a comparação entre grafos é transformada na comparação de vectores multidimensionais.

Para extrair a geometria das divisões da planta, usamos uma biblioteca de reconhecimento de formas geométricas, CALI [8, 10, 7], que produz um conjunto de descritores geométricos que representam a geometria do desenho. Tal como no caso da topologia, também a semelhança entre geometrias é convertida na comparação de vectores numéricos.

A plataforma de recuperação que irá ser usada para completar o processo de recuperação de plantas de arquitectura usando esboços, foi desenvolvida por Fonseca [6] e permite a pesquisa de desenhos vectoriais em conjuntos com uma elevada quantidade de elementos. O processo de pesquisa oferecido pela plataforma faz uma primeira filtragem, comparando a topologia da interrogação com a topologia dos desenhos catalogados. Os desenhos que passarem a filtragem por topologia são depois comparados usando a geometria, e no final ficamos com um conjunto de plantas similares à interrogação submetida. De salientar que o utilizador não necessita de criar uma interrogação de complexidade semelhante às plantas existentes na base de dados, pois o mecanismo de comparação da plataforma consegue comparar interrogações com partes de plantas.

4. Conclusões

Neste artigo descrevemos um conjunto de algoritmos simples para simplificar plantas de arquitectura. O principal objectivo desta simplificação é possibilitar a extracção automática de informação visual sobre o conteúdo dos desenhos, para posterior catalogação e pesquisa. Apresentamos ainda o enquadramento destas técnicas de simplificação numa plataforma mais genérica de recuperação de informação multimédia, através da conversão da organização espacial das divisões da planta num grafo de topologia.

Embora os algoritmos tenham sido testados com várias plantas, durante a fase de desenvolvimento, ainda não temos resultados experimentais para apresentar, pois este trabalho ainda está numa fase inicial do seu desenvolvimento. Planeamos num futuro próximo, realizar testes experimentais para: medir tempos de simplificação, e verificar se são aceitáveis; analisar os resultados simplificados, para ver

se são perceptíveis; e verificar se os utilizadores realizam esboços que se assemelham às plantas simplificadas.

Agradecimentos

Este trabalho foi financiado em parte pela Fundação para Ciência e Tecnologia, Bolsa POSI/EIA/59938/2004.

Referências

- [1] D. V. Bakergem. Image Collections in The Design Studio. In *The Electronic Design Studio: Architectural Knowledge and Media in the Computer Age*, pages 261–272. MIT Press, 1990.
- [2] B.-C. Björk, K. Löwnertz, and A. Kiviniemi. ISO DIS 13567 - The Proposed International Standard for Structuring Layers in Computer Aided Building Design. *Electronic Journal of Information technology in Construction*, 2:32–56, 1997.
- [3] M. Clayton and H. Wiesenthal. Enhancing the Sketchbook. In *Proceedings of the Association for Computer Aided Design in Architecture (ACADIA'91)*, pages 113–125, Los Angeles, CA, 1991.
- [4] Ellen Y. Do. What's in a Diagram that a Computer Should Understand? In M. Tan and R. Teh, editors, *Proceedings of The Sixth International Conference on Computer Aided Architectural Design Futures (CAADF'95)*, pages 103–114, National University of Singapore, 1995. The Global Design Studio.
- [5] Ellen Y. Do. *The right tool at the right time*. PhD Thesis, Georgia Institute of Technology, September 1998.
- [6] Manuel J. Fonseca. *Sketch-Based Retrieval in Large Sets of Drawings*. PhD thesis, IST/UTL, Lisboa, Portugal, July 2004.
- [7] Manuel J. Fonseca and Joaquim A. Jorge. CALI : A Software Library for Calligraphic Interfaces. INESC-ID, available at <http://immi.inesc-id.pt/cali/>, 2000.
- [8] Manuel J. Fonseca and Joaquim A. Jorge. Experimental Evaluation of an on-line Scribble Recognizer. *Pattern Recognition Letters*, 22(12):1311–1319, 2001.
- [9] Manuel J. Fonseca, Alfredo Ferreira Jr., and Joaquim A. Jorge. Content-Based Retrieval of Technical Drawings. *International Journal of Computer Applications in Technology (IJCAT)*, 23(2/3/4):86–100, 2005.
- [10] Manuel J. Fonseca, César Pimentel, and Joaquim A. Jorge. Cali: An online scribble recognizer for calligraphic interfaces. In *Proceedings of the 2002 AAAI Spring Symposium - Sketch Understanding*, pages 51–58, Palo Alto, USA, March 2002.
- [11] M. Gross, C. Zimring, and E. Do. Using Diagrams to Access a Case Library of Design. In *Artificial Intelligence in Design (AID'94)*, pages 129–144, Netherlands, 1994. Kluwer Academic Publishers.
- [12] Mark D. Gross. Indexing Visual Databases of Designs with Diagrams. In A. Koutamanis, H. Timmermans, and I. Vermeulen, editors, *Visual Databases in Architecture*, pages 1–14, Avebury: Aldershot, UK, 1995.
- [13] Mark D. Gross and Ellen Yui-Luen Do. Diagram Query & Image Retrieval in Design. In *Proceedings of the 2nd IEEE International Conference on Image Processing*, volume 1 of 308–311, Washinton D.C., 1995. IEEE Computer Society Press.

POSTERS

Publicação Digital em Computação Gráfica: Uma proposta de redesenho da Revista Virtual

Nuno Gonçalves, Rodrigo Machado, Frederico C. Figueiredo, Joaquim A. Jorge
Instituto Superior Técnico / Departamento de Engenharia Informática / INESC-ID
R. Alves Redol, 9, Sala 523, 1000-029 Lisboa, Portugal

nfrg@mega.ist.utl.pt, rdm@mega.ist.utl.pt, fepf@immi.inesc-id.pt, jorgej@acm.org

Resumo

A revista *VIRtual* é uma publicação electrónica profissional abrangendo as áreas de Visualização, Interacção e Reconhecimento de Padrões. A sua edição tem servido a comunidade nacional constituindo o principal repositório de actividade científica de Computação Gráfica e Multimédia em Portugal. A revista tem actualmente uma página de Internet com alguns problemas de usabilidade, como por exemplo, a inexistência de um motor de pesquisa para que os utilizadores possam pesquisar por publicações, e apresenta um workflow pouco eficiente, dados os requisitos do processo editorial.

O objectivo do presente consiste em introduzir melhorias no workflow e permitir aos utilizadores submeter, avaliar, publicar, e pesquisar documentos com maior simplicidade. Este artigo descreve a proposta de trabalho e revê as etapas iniciais de redesenho centrado no utilizador, bem como da nova infra-estrutura de suporte. O trabalho centra-se em três vertentes: workflow, gestão de conteúdos e desenho centrado no utilizador. Do levantamento de requisitos discutimos as principais opções de projecto e apresentamos o protótipo actual.

Palavras-Chave

Recensão por Pares, Computação Gráfica, Gestão de Conteúdos, Workflow, Interfaces Pessoa-Máquina

1 Introdução

O suporte digital é, actualmente, o meio mais utilizado na disseminação e arquivo de documentos. A *VIRtual* é um jornal electrónico de publicação profissional nas áreas de Visualização, Interacção e Reconhecimento de Padrões. Todos os conteúdos disponíveis encontram-se armazenados, em formato digital. Estes são acessíveis, gratuitamente, através de um conjunto de páginas Internet estáticas, no endereço <http://virtual.inesc-id.pt>. Desde 1998, a revista tem servido a comunidade Portuguesa com um arquivo digital de publicações científicas, nas mais variadas áreas da Computação Gráfica. Serve também de repositório para as actas de várias conferências de prestígio, nacionais e internacionais, tais como: EPCG (Encontro Português de Computação Gráfica), DVS-IS (Workshop Internacional em Desenho, Especificação e Verificação de Sistemas Interactivos), entre outras.

Actualmente, as páginas Internet e o sistema editorial da revista apresentam vários problemas de usabilidade que dificultam e tornam ineficientes os processos de publicação e acesso aos conteúdos disponíveis, no repositório digital. Os mecanismos de recensão e de publicação não se encontram optimizados nem automatizados, sendo o processo elaborado de uma forma bastante tradicional, onde os recursos humanos e de tempo são muito exigentes. Na submissão de documentos, a *VIRtual* sugere o HTML como

formato favorito de publicação. Tal requer, por parte dos autores, um conhecimento prévio da linguagem, limitando assim a disponibilidade ao nível da contribuição uma vez que existem alternativas, como o PDF e XML, mais flexíveis e adequadas à publicação digital. A publicação de actas de conferências necessita de, por parte do administrador (i.e. o responsável pela manutenção), um investimento de tempo e trabalho na criação das mais variadas páginas de suporte, bem como na indexação e arquivo de todas as publicações. A inexistência de mecanismos eficientes de indexação e pesquisa de documentos dificultam a acessibilidade e visibilidade dos documentos disponíveis na revista.

A nossa proposta pretende: (1) introduzir melhorias significativas no workflow editorial, com a utilização de um Gestor de Conteúdos; (2) optimizar a interacção entre os utilizadores e a revista, com o desenho de interfaces centradas no utilizador; e (3) permitir a pesquisa de documentos e outra meta-informação. Para o efeito, pretendemos criar um conjunto de ferramentas, com bons níveis de usabilidade, tendo em vista melhorar a gestão do local e introduzir um processo editorial optimizado e mais eficiente para tarefas chave, tais como: submissão e avaliação de documentos; contacto com os autores; publicação de artigos; indexação e pesquisa de conteúdos; entre outras. Desta forma aspiramos trazer maior visibilidade à *VIR-*

tual e descer os custos (de tempo e recursos humanos) na submissão, gestão, manutenção, publicação e pesquisa de documentos científicos.

A secção seguinte descreve os principais requisitos para uma infra-estrutura e um mecanismo electrónico, optimizado, de publicação digital. Em seguida é apresentada, de forma sucinta, o estudo e análise realizado com vários Gestores de Conteúdos e de Conferências. Seguem-se as propostas de reformulação da interface e do *work ow* e a apresentação do protótipo inicial. Finalizamos com as conclusões e direcções para trabalho futuro.

2 Requisitos

Com vista a definir os requisitos do novo sistema de publicação digital, para a revista VIRTual, começámos por fazer um estudo do processo de recessão (*work ow*) e da interface actual, detalhados nas Secções 4 e 5. De seguida, elaborámos uma análise qualitativa de vários Gestores de Conteúdos e de Sistemas de Gestão de Conferências, descrito na Secção 3. Com a elaboração de um inquérito de opinião, foi possível validar e acrescentar novos requisitos à nossa proposta [3]. O inquérito consistiu num questionário on-line que possuía perguntas específicas com o objectivo de obter informação sobre 4 grupos de utilizadores: visitantes de revistas, autores de artigos, revisores e administradores/editores. A amostragem dos inquiridos incluiu 50 pessoas de ambos os sexos e com idades situadas entre os 18 e os 45 anos. Com base nos resultados obtidos foi definido o perfil do utilizador:

Conhecimento e Experiência:

- Habilitações Literárias: Licenciatura
- Experiência Computador e Internet: Alta
- Experiência com as Tarefas: Pouca
- Experiência com o Sistema: Nenhuma

Características das Tarefa:

- Frequência de Uso: Pouca
- Treino Primário: Nenhum
- Importância e Complexidade das Tarefas: Variável

As respostas aos questionários permitiu validar os requisitos levantados, dos quais destacamos:

- (Re)Submissão de docs em formato ZIP e PDF;
- O editor publica/rejeita documentos submetidos;
- O editor pode comunicar e interagir com os autores;
- Listar os últimos documentos publicados;
- Indexação e pesquisa de documentos/conteúdos;
- Possibilidade de inscrever numa *newsletter* para estar a par das últimas novidades, eventos e publicações;

- *Login* baseado no papel do utilizador;
- Gestão de privilégios dos utilizadores;
- Atribuir editores a diferentes secções do local.

3 Trabalho Relacionado

Actualmente, muitos consideram crítico o uso de sistemas para a gestão de conteúdos (CMS, *Content Management Systems*) no sucesso de todos os locais Internet, nomeadamente organizações e instituições públicas e privadas [5]. Aqueles que são responsáveis por manter actualizado um local Internet estão cientes das dificuldades associadas. Dependendo dos objectivos do local, é necessário dominar um conjunto diverso de tecnologias para criar, editar e gerir conteúdo on-line, mas também conhecer técnicas, ferramentas e ter presente um conhecimento vasto em Engenharia e Informática.

Um CMS pode ser definido de duas formas. Uma definição mais generalista seria "a separação entre a gestão do conteúdo e o desenho/apresentação"[1]. Em alternativa, "Um sistema de base-de-dados que armazena, separadamente, os elementos referentes a conteúdos e ao desenho/estrutura do local Internet. Um CMS permite aos utilizadores editar, adicionar e apagar conteúdo no mesmo local, sem afectar o seu desenho"[6].

Realizámos um estudo sobre o estado-da-arte em CMSs [3] para perceber quais, dos actualmente disponíveis, suportariam e resolveriam os problemas apresentados nas secções anteriores. Todos os CMSs analisados correm em ambiente UNIX (Linux) e são de domínio público (*open source*). No total, testamos 6 CMS's, tendo o estudo incidido maioritariamente na facilidade de uso e nas funcionalidades que permitiam utilizar. A Figura 1 apresenta um sumário de 3 dos CMS's testados que mostraram ser menos complexos, limitados e mais fáceis de (re)utilizar.

O Mambo [4] foi o que mais se destacou. A facilidade de uso, o suporte à maioria dos requisitos definidos, a linguagem de desenvolvimento e a grande comunidade de colaboradores e participantes na iniciativa, foram os aspectos mais positivos da análise. O Plone foi o que apresentou uma interface mais difícil de utilizar. A linguagem e o suporte Web pouco reconhecidos limitam o uso e reaproveitamento do CMS. No entanto o Plone apresentou um conjunto interessante de funcionalidades que permitiriam criar um mecanismo de publicação digital com alguns dos requisitos levantados. O WebGui é baseado numa linguagem já relativamente antiga, para a criação de serviços Internet, e apresenta uma interface relativamente fácil de usar. Em comparação com o Mambo, tem a vantagem de poder recorrer ao conceito de *drag-n-drop* durante a realização das tarefas, algo que só por si só não justifica a escolha em detrimento da primeira alternativa.

Em conclusão, o CMS escolhido para o desenvolvimento da infra-estrutura de suporte à Revista VIRTual foi o Mambo. Destacam-se as seguintes funcionalidades: registo de utilizadores; a submissão de documentos;

	Mambo	Plone	WebGui
Linguagem	PHP	Python	Pearl 5.6
Base de dados	MySQL	Zope	MySQL
Servidor	Apache	Zope	Apache
Suporte Web	PHP 4.1.2	Zope	Mod_Perl
Motor de pesquisa	Sim	Sim	Sim
Gestor de ficheiros	Plug-in	Sim	Sim
Facilidade de uso	Boa	Má	Razoável
Drag-N-Drop	Não	Não	Sim

Figura 1. Comparação entre CMSs

a aprovação de documentos a publicar; o uso de *templates* para a definição da Interface; e a possibilidade de gerir e organizar automaticamente os documentos publicados on-line, sem a intervenção do editor / administrador.

Com o objectivo de melhorar o *work ow* da VIRTUAL realizámos um estudo sobre Gestores de Conferências [3]. Verificou-se que a maioria permite: a submissão, por autores, do trabalho em diversos formatos (PDF, DOC, PS, HTML, etc.); a atribuição, por um editor-chefe, a um conjunto de avaliadores competentes no tópico em que se incide a submissão; a avaliação do documento; e a aprovação/rejeição da submissão com base nas avaliações produzidas. Todas estas etapas são realizadas on-line. O CyberChair [7] foi um dos vários incluídos no estudo.

A usabilidade tem um papel muito importante no desempenho do utilizador, em interfaces pessoa-máquina. Por definição, usabilidade é o grau de facilidade que um utilizador tem em realizar uma tarefa específica, num determinado sistema/contexto. Para desenvolver sistemas usáveis, úteis e de fácil aprendizagem, é preciso avaliar todas as etapas do desenvolvimento. Várias técnicas e estratégias podem ser aplicadas, dependendo do contexto e do objectivo [2]. Os primeiros testes realizados, num protótipo inicial, estão descritos na Secção 5.

4 Modificações ao Workflow

O *work ow* actual da VIRTUAL torna o processo de submissão, revisão e publicação de artigos científicos, complexo e demorado. O administrador recebe, na maior parte das vezes, os artigos individuais e as várias publicações (de uma conferência, por exemplo) em formato digital. Independentemente do tipo de submissão (individual ou um conjunto de artigos), o administrador tem que: (1) analisar o conteúdo dos ficheiros; (2) contactar com os autores, via correio electrónico, para quaisquer alterações que sejam necessárias efectuar; (3) alterar, em caso de necessidade, o nome dos ficheiros; (4) modificar a submissão por forma a respeitar os requisitos de *layout*; e (5) no caso de submissão de actas de conferências, tem que criar uma secção especial para alojar os vários artigos e criar as habituais subsecções (lista de autores, lista de artigos, programa, etc.). O *work ow* actual está ilustrado na Figura 2.

A nossa proposta consiste num mecanismo que permitirá

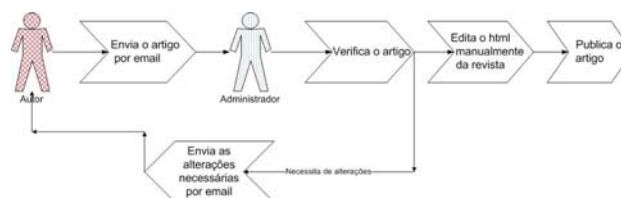


Figura 2. Workflow Actual

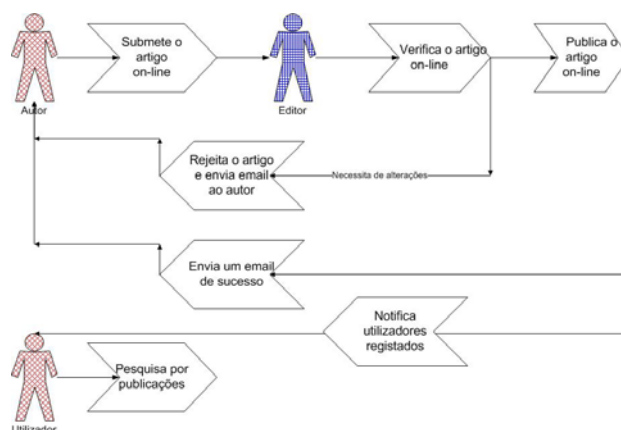


Figura 3. Workflow Proposto

optimizar todo o processo. Para tal vamos recorrer a um CMS (Mambo) que servirá de infra-estrutura de suporte ao mecanismo de publicação e interacção com a revista e os seus conteúdos. Após os estudos realizados foi simples perceber qual a melhor forma de optimizar e automatizar *work ow*. Na nossa proposta, vamos estruturar e automatizar os processos de submissão, avaliação, publicação e pesquisa de documentos, para ultrapassar os problemas identificados na Secção 1. A Figura 3 descreve o novo *work ow*, suportado e automatizado pelo uso, modificação e extensão do CMS escolhido.

5 Proposta de Redesenho e Resultados Iniciais

A interface actual da revista VIRTUAL, Figura 4, é composta por uma página inicial muito extensa (com cerca de 5 ecrãs de *scroll* vertical, numa resolução de 1024x768). Não existe um menu de navegação. Existe uma zona na página inicial com uma estrutura mais ou menos formal, onde se encontram elos para secções da página que explicam os objectivos da revista, as instruções para submeter um documento, a política editorial, etc. O facto de não existir um motor de pesquisa prejudica significativamente a usabilidade e torna complicada, se não impossível, a tarefa de encontrar documentos disponíveis na revista. Não existe qualquer interface de administração uma vez que todo o processo é feito manualmente, através da edição e criação de páginas HTML. Para optimizar todo o processo é necessário criar uma estrutura bem definida para a organização dos conteúdos.

Para desenhar e testar uma nova interface, foram criados esboços das várias páginas da revista (protótipo de baixa fidelidade, Figura 5) e foram realizados testes de usabilidade com 10 utilizadores. Para os testes foram definidas 4 tare-



Figura 4. Interface Actual

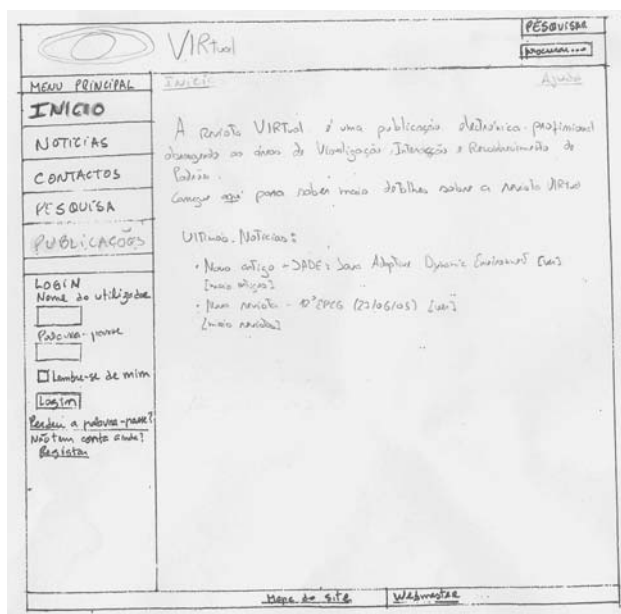


Figura 5. Protótipo de Baixa Fidelidade

fas típicas, do uso do sistema, com o objectivo de medir o nº de erros, o tempo necessário para completar, e o número de "cliques" necessários. 4 dos 10 utilizadores do protótipo são experimentados em avaliação de heurísticas o que nos permite obter outro tipo de resultados. Neste caso, houve uma interacção com os avaliadores dando mais ênfase à sua opinião sobre a interface e aos erros encontrados na mesma. No final foram colocadas, aos utilizadores, algumas questões específicas sobre a interface, de forma a responder a dúvidas que não são imediatamente observáveis, durante a realização dos testes.

Resultados iniciais demonstram a necessidade de uma organização dos conteúdos, bem como uma apresentação sobre a forma de *menus* e *sub-menus*, com vista a facilitar a navegação, por parte do utilizador. É imperativo que uma pesquisa (simples e avançada) esteja sempre presente e visível em todas as páginas/secções do local. A submissão de documentos, por autores, deve permitir a introdução de toda a meta-informação necessária, bem como a escolha da edição/conferência a que a submissão se refere. Verificou-se também a necessidade de agrupar opções de um mesmo contexto, em zonas destacadas e de visíveis.

6 Conclusões e Trabalho Futuro

A publicação digital é, actualmente, o suporte utilizado pela maioria das revistas electrónicas, nas mais diversas áreas de investigação científica. Processos e mecanismos de recensão eficientes, flexíveis e otimizados são essenciais para a manutenção e visibilidade destes repositórios. A VIRTUAL é uma revista e um arquivo de publicação electrónica, na área da Computação Gráfica, de grande importância para a comunidade nacional. Analisando o mecanismo e interface actual da revista foi possível perceber os vários condicionantes no processo de submissão, revisão, publicação e pesquisa de artigos científicos. Processos que ainda hoje são realizados de uma forma tradicional/manual e com grande sobrecarga para o responsável pela sua manutenção.

A introdução, uso e modificação de um gestor de conteúdos permitirá simplificar, automatizar e otimizar todo o processo. Com um desenvolvimento iterativo da interface, centrada no utilizador, poderemos obter melhores níveis de usabilidade e aumentar a facilidade e eficiência na realização das tarefas. Uma organização cuidada, e bem definida, dos conteúdos, e a inclusão de um mecanismo de pesquisa, aumenta a acessibilidade e visibilidade dos recursos. As modificações propostas visam também reduzir o tempo e os recursos humanos necessários em todo o processo de manutenção da VIRTUAL. É importante referir que este trabalho ainda se encontra em desenvolvimento, como pode ser constatado nos resultados apresentados.

Futuramente, propomo-nos a continuar o desenvolvimento e análise do protótipo, efectuando testes mais alargados, de forma a adequarmos a interface às necessidades dos vários tipos de utilizadores envolvidos no processo. O objectivo será perceber se a optimização e automatização do processo de submissão, auxiliada com o uso de um gestor de conteúdos, poderá trazer maior visibilidade e diminuir os custos de manutenção da Revista VIRTUAL.

Referências

- [1] ATNF. Cms evaluation. *Australia Telescope National Facility, Relatório Técnico*, 2002.
- [2] A. Dix et al. Human-computer interaction. *Prentice Hall*, 1998.
- [3] N. Gonçalves et al. Gestor de conteúdos para publicações digitais (relatório intercalar). *IST/DEI, Relatório Técnico*, Novembro 2004.
- [4] Mambo. Available at <http://www.mamboserver.com/>, acedido Novembro 2004.
- [5] J. Robertson. So, what is a content management system? *KM Column*, Junho 2003.
- [6] M. Sipokazi. Content management system: Users guide. Technical report, Rhodes University, 2003.
- [7] R. van de Stadt. Cyberchair: A web-based groupware application to facilitate the paper reviewing process. Technical report, 2001.

SeaView: a Seafloor and Habitat Visualizer

Daniel Almeida

Samuel Nunes

Hans du Buf

Vision Laboratory, University of Algarve, Faro, Portugal

{dalmeida, srnunes, dubuf}@ualg.pt

url: w3.ualg.pt/~dalmeida/pub/EPCG2005-SeaView-color.pdf

Abstract

In this paper we present a benthic habitat and seafloor visualizer called SeaView that is currently being developed for the EU-funded project EXOCET/D. This is an open-source and multi-platform application that uses OpenGL[®] and the GTK+2 toolkit for visualization and user interface. The goal of this application is to help biologists in marine research, by integrating seafloor and benthic information by means of a fast interactive visualization. The application will be prepared to handle huge data volumes for large-scale surveys.

Keywords

Scientific visualization, digital terrain maps, multimodel data.

1. INTRODUCTION

Interactive visualization of a Digital Terrain Map (DTM) of the seafloor, acquired by any kind of mapping technology [Kenny00], is very important to biologists and other scientists working in marine research. Such applications offer the possibility to develop a good "feeling" of the seafloor's structure. More information, like fauna and flora distributions, can be displayed to enhance the overall impression of the 3D shape. Optical images, if available, must also be correctly superimposed on the map.

Of great concern is fast interactivity when dealing with huge data volumes [Loke03]. Without a proper approach, the frame rate is drastically affected. We achieve a substantial improvement of rendering performance, even with huge data volumes, through (a) a multi-resolution representation of the original data, made possible by mesh-simplification algorithms [Lam01], [Bremer04], [Cignoni97]; and (b) view frustum culling, effectively achieved using a space-partitioning method [Fu04], [Abrash96], [Lindstrom02].

The EU-funded project EXOCET/D [Sarradin04] aims at developing, implementing and testing specific instruments for exploring, describing, quantifying and monitoring biodiversity in deep-sea fragmented habitats, as well as identifying links between community structure and environmental dynamics. One goal of this project is to develop a DTM of the seafloor in the deep ocean, using sonar.

Our objective is to create an intuitive and flexible 3D visualization tool for the EXOCET/D project as well as similar studies. This software aims at helping biologists - possibly at different laboratories and using computers with different operating systems - in marine research.

Because it is important to develop an open-source and multi-platform application, OpenGL and the GTK+2 toolkit are used for 3D visualization and graphic user interface. This combination enables us to create one application that can be compiled on multiple platforms, e.g. Linux, MacOS X and Windows XP. The viewer - which is based on state-of-the-art computer graphics - has simple and fast interactivity and will offer a very appealing way to investigate acquired seafloor data.

2. APPLICATION DESCRIPTION

Two different modes of visualization are planned: (a) a scene centered global (3D map) view of the DTM, and (b) a viewer centered simulated camera view. In global view mode, navigation consists of (1) scrolling through the 6 possible directions and (2) rotating around the center point. Zooming in and out is also possible, enabling detailed inspection of selected areas. While in global view mode, it is possible to define multiple regions to be observed. These will be properly represented on the 3D map and by small preview windows. Each region's position and angle can be changed intuitively using the mouse and configuration gadgets.

The simulated camera mode allows a more realistic experience, giving scientists the opportunity to explore the DTM from a first-person perspective. This translates into a simple and fast visualization by "flying" over the terrain, with perspective projection, as if the observer were a camera installed on an ROV with real-time control over the vehicle. Added effects, like fog, texturing, animated drifting particles and the representation of a model of the vehicle as it manoeuvres over the terrain (augmented reality), will contribute to enhance the overall experience. In the future we aim at

achieving photorealistic representations on high-end computers, while preserving smooth animations on older systems.

In both visualization modes, available seafloor classifications (flora/fauna) obtained with sonar on the terrain can be represented by superimposing color-coded, and/or pattern-coded information. It will also be possible to examine local information in detail, e.g. by selecting the desired location/region with the mouse.

Optical images of the seafloor, if available, must also be correctly mapped on the 3D representation. If necessary, overlapping images must be tiled and, if space- and timestamps are not accurate enough, optical details must be correlated with the DTM in order to correct positions [Ruiz04]. Additionally, the optional representation of altitude using false-color coding and/or topographic contour lines will ease the overall perception of the DTM and the study of correlations between seafloor classification and altitude. Furthermore, terrain coordinates can be represented by a grid with latitude and longitude intervals, displayed above the terrain at constant altitude and/or projected on the terrain.

As in any professional software, a wide range of file formats must be supported to allow the use of data from different sources. These will include known file formats, for either triangle meshes or point clouds.

The application is very flexible, intuitive and functional. Two key shortcuts may be assigned to the same action, giving input flexibility to users with different preferences. Simple functions are always a mouse click or a key shortcut away. In all circumstances we give special attention to fast interactivity, i.e. with no waiting times beyond one or two seconds.

3. SPEED CONSIDERATIONS

We achieve simple and fast interactive visualization, even with huge data volumes, by using (a) a multi-resolution representation and (b) view frustum culling.

A multi-resolution representation allows simplified models of the terrain to be displayed in situations where surface details are not apparent to the user [Cignoni], [Bremer04], e.g. when the viewpoint is too far from the surface to present such details. This optimization implies the integration of mesh simplification algorithms [Lam01], [Bremer04], and effectively solves part of the problem of vast terrain surface visualization.

View frustum culling is another optimization method that enables a substantial improvement of our rendering performance, by suppressing mesh triangles that fall outside the view volume. This can be effectively achieved using a space-partitioning method [Fu04], [Abrash96], [Lindstrom02], and making use of its hierarchical data structure. This way frame rates are almost unaffected by the volume of data. Space partitioning is also useful to speed up calculations (e.g. of normal vectors).

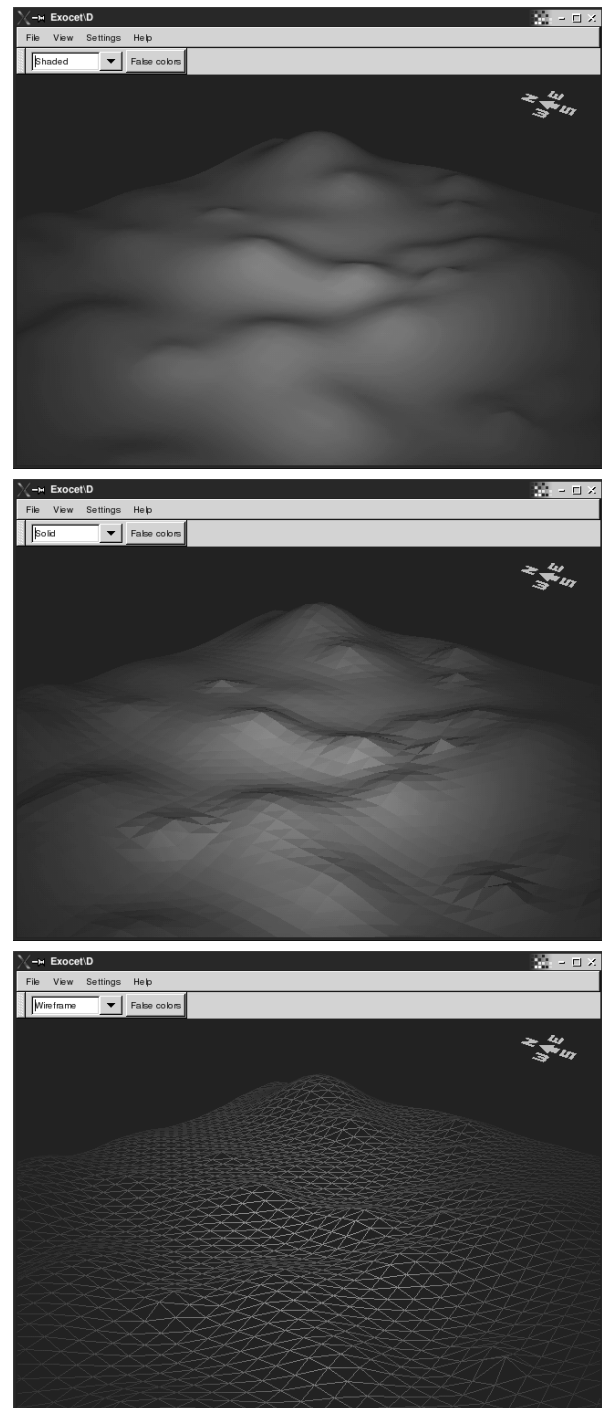


Figure 1: Screenshots of the main display window showing a seabed surface rendered with (top) Gouraud shading, (middle) flat shading and (bottom) wireframe.

The use of display lists and vertex arrays can also improve rendering performance [Woo93], [Shreiner01], [Weiler02], by preventing the same calculations from being repeated. To avoid long delays whenever a new DTM is loaded, the display lists for each partition are only created when first needed, and are then cached until (a) the DTM is closed, (b) the application terminates, or (c) the system is running low on memory resources. In the latter case, the least probably needed display lists

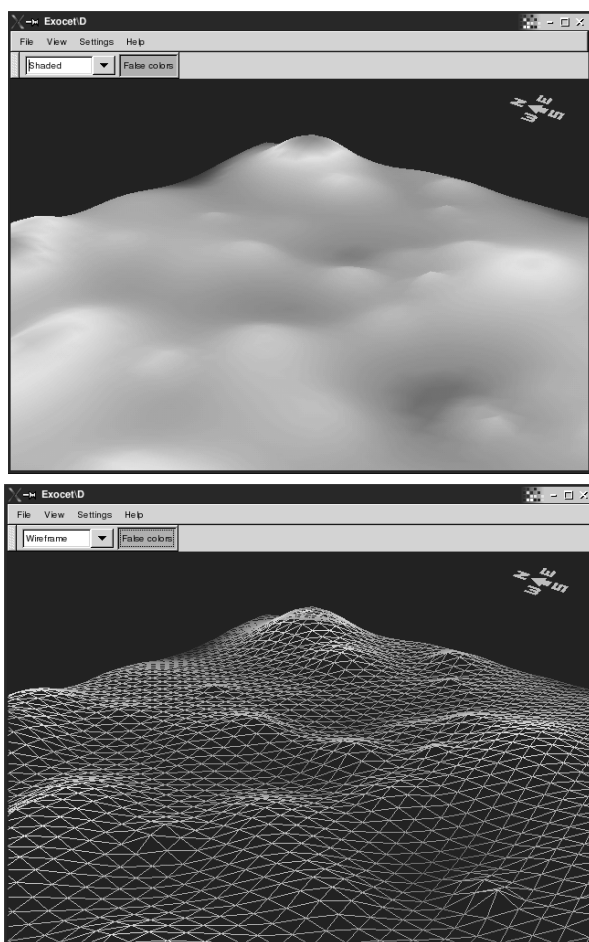


Figure 2: Representation of altitude using false-color coding with (top) solid triangles and (bottom) a wireframe representation.

(according to their locations relative to the navigation angle and position) are freed first.

4. CONCLUSIONS AND FUTURE WORK

The application is at an early development stage. However, a considerable number of features are already implemented. Display and navigation are fully operational in simulated camera mode, as well as multiple cameras and light sources. It is possible to visualize a DTM with solid triangles or just the wireframe and rendering is done using flat or Gouraud shading (Fig. 1), with optional representation of altitude by false-color coding (Fig. 2). Note that the color version of this paper is available at the webpage of the authors. Normal vectors are also automatically calculated when not present in the input DTM file.

Several configuration windows are available for the implemented features. It is possible to define several light sources, up to the maximum of eight lights supported by the OpenGL standard (Fig. 3, top). These can be fixed to the camera or to the scene, with the position, direction, color and ambient values definable by the user (Fig. 3, bottom). Likewise, a large number of cameras is possible, and the position, direction and type of each one is fully configurable at any time (Fig. 4). Possible camera types

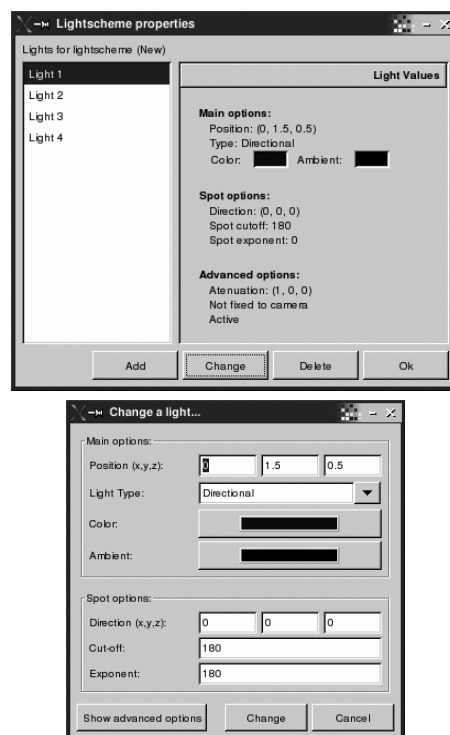


Figure 3: Configuration windows for (top) light schemes and (bottom) light properties.

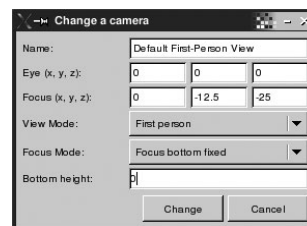


Figure 4: Configuration window for camera properties.

are (a) with fixed focus, (b) focused on a fixed depth or (c) focused on an auto-detected depth (detected bottom). Configuration windows for false-color maps (Fig. 5 top) and key shortcuts (Fig. 5 bottom) are also available.

The filetype currently supported is the “Comma Separated Value” (CSV) type, similar to the one used by Matlab[®]. This can be used to import matrices and/or point cloud vertices. The latter require the implementation of a triangulation algorithm [Bern92], e.g. Delaunay triangulation. Considering the nature of the data - a single surface where vertices with different height values do not overlap - a simpler 2D triangulation method can be used. More filetype importers and exporters will be implemented in the future. The presentation of important information like seafloor classifications, terrain coordinates, and optical images, is under development.

Multi-resolution visualization and space partitioning are also important features being implemented. These will enable much larger DTMs to be viewed, and help distributing and largely reduce processing times. Alternatively, a view-dependent level-of-detail

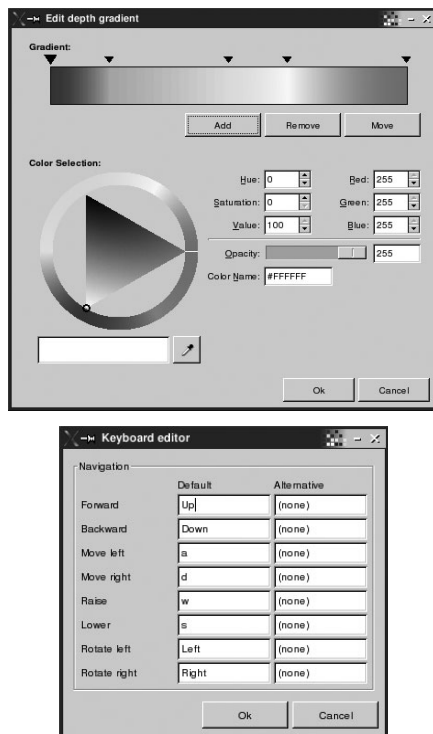


Figure 5: Configuration windows for (top) defining false-color maps and (bottom) keyboard shortcuts.

algorithm, complete with geomorphing [Lindstrom02], [Azuma03], can be implemented to, for each frame, construct an adaptive mesh from scratch. Although more complex, these systems enable a finer tessellation of regions close to the camera, while coarsely representing those far away.

For demonstration purposes it will be a good idea to implement a special navigation function that shows the DTM following the path of the vessel - even with augmented reality, showing the vessel itself in perspective - with optional recording into a video file. Alternatively, a path defined by the user can be used and/or a set of focus-of-attention spots can be defined, e.g. to show particular areas of interest from better angles.

Future features may also include the possibility to modify or recreate the DTM from the original sonar data, e.g. by correcting landmark positions. Further improvements on application intuitivity and flexibility will be achieved, by giving special attention to feedback from biologists.

5. ACKNOWLEDGMENTS

This research is funded by the EU project EXOCET/D, contract FP6-GOCE-CT-2003-505342 with CINTAL (Centro de Investigação Tecnológica do Algarve).

6. REFERENCES

- [Yao83] F. F. Yao, A 3-space partition and its applications, Proc. 15th Annual ACM Symposium on Theory of Computing (1983), pp 258-263.
- [Bern92] M. Bern, D. Eppstein, Mesh generation and optimal triangulation. In: Computing in Euclidean

Geometry (Ding-Zhu Du and Frank Hwang, editors), Lecture Notes Series on Computing, vol. 1, (1992) pp 23-90. World Scientific, Singapore.

- [Woo93] M. Woo, J. Neider, T. Davis, OpenGL Programming Guide. The Official Guide to Learning OpenGL, Release 1 (1993), Addison-Wesley.
- [Abrash96] M. Abrash, Inside Quake: visible-surface determination, Dr. Dobbs's Sourcebook (1996), pp 41-45.
- [Cignoni97] P. Cignoni, C. Montani, E. Puppo, R. Scopigno, Multiresolution representation and visualization of volume data, IEEE Tr. Visual. and Computer Graphics, Vol. 3 (4) (1997), pp 352-369.
- [Kenny00] A. J. Kenny, et al. An overview of seabed mapping technologies in the context of marine habitat classification, ICES, (September 2000).
- [Lam01] R. Lam, R. Loke, H. du Buf, Smoothing and reduction of triangle meshes, Proc. 10th Port. Comp. Graphics Meet. (2001), Lisbon, pp 97-105.
- [Shreiner01] D. Shreiner, Performance OpenGL: platform independent techniques or a bunch of good habits that will help the performance of any OpenGL program, SIGGRAPH 2001, Course #3 (July 2001).
- [Lindstrom02] P. Lindstrom and V. Pascucci, Terrain simplification simplified: a general framework for view-dependant out-of-core visualization, IEEE Tr. Vis. Comp. Graphics, Vol. 8 (3) (2002), pp 239-254.
- [Weiler02] M. Weiler, M. Kraus, T. Ert, Hardware-based view-independent cell projection, Proc. IEEE Vol. Vis. and Graphics Symp. 2002, pp. 13-22.
- [Azuma03] D. I. Azuma, et al., View-dependent refinement of multiresolution meshes with subdivision connectivity, Proc. Afrigraph (2003), pp 69-78.
- [Loke03] Loke, R.E. and du Buf, J.M.H.. Interactively visualizing 18-connected object boundaries in huge data volumes, 11th Int. Conf. Discrete Geometry for Computer Imagery, Naples. Springer LNCS Vol. 2886 (2003), pp 544-553.
- [Bremer04] P. Bremer, H. Edelsbrunner, B. Hamann, V. Pascucci, A topological hierarchy for functions on triangulated surfaces. IEEE Tr. on Visualization and Computer Graphics, Vol. 10 (4) (2004), pp 385-396.
- [Fu04] C. Fu, et al., Binary-space-partitioned images for resolving image-based visibility, IEEE Tr. on Visual. Computer Graphics, Vol. 10 (1) (2004), pp 352-369.
- [Ruiz04] I. T. Ruiz, et al., Concurrent mapping and localization using sidescan sonar. IEEE J. of Oceanic Engineering, Vol. 29 (2) (2004), pp. 442-456.
- [Sarradin04] P. Sarradin, J. Sarradin. EXtreme ecosystem studies in the deep OCEan: Technological Develop., 2004. <<http://www.ifremer.fr/exocetd>>

Simulação Visual de Trajectórias Aéreas

Cesário Nogueira Raúl Costa
Univ. de Trás-os-Montes e Alto Douro
5000, Vila Real
{engcesario,raulscosta}@gmail.com

Carlos Amaro
PT Inovação
3810-106 Aveiro
camaro@ptinovacao.pt

António Coelho;

Maximino Bessa¹

Manuel Cabral

António Valente

Raul Morais

Salviano Soares

CETAV, Universidade de Trás-os-Montes e Alto Douro
5000 Vila Real

{acoelho, maxbessa, mcabral, avalente, rmorais, salblues}@utad.pt

Resumo

A realidade virtual é uma área da computação gráfica que engloba a síntese de ambientes imersivos com o intuito de proporcionar a sensação de estar noutro lugar.

Neste projecto foi criado um sistema de realidade virtual, projectado para trabalhar com informação referenciada por coordenadas espaciais ou geográficas, aliado a uma modelação tridimensional do terreno para recriar trajectórias aéreas. Neste artigo apresentamos a arquitectura e capacidades desenvolvidas, bem como potenciais aplicações.

Palavras-chave

GPS, VRML, SIG, Trajectórias Aéreas.

INTRODUÇÃO

As primeiras tentativas do homem voar com asa de estrutura rígida e movidas pelo impulso de um homem, ocorreram no fim do século XIX pelo engenheiro alemão Otto Lilienthal.

Hoje em dia, desportos como o voo livre, balonismo e aeromodelismo, figura 1, são cada vez mais procurados e têm em Portugal um mercado em expansão.



Figura 1 - Desportos aéreos.

Também o turismo aéreo aproveita os céus e as belas paisagens proporcionadas pela vista aérea, figura 2.

A análise destas rotas, bem como das trajectórias descritas por aves migratórias, tem despertado o interesse científico e motivado diversas investigações [Biro02, Gould80, Igor00].

Neste artigo é apresentada uma aplicação que permite simular, num ambiente virtual, trajectórias aéreas obtidas a partir de um módulo de localização *Global Positioning System* (GPS) [Nogueira05].



Figura 2 - Turismo aéreo.

Esta aplicação foi desenvolvida como uma aplicação gráfica baseada em *Virtual Reality Modelling Language* (VRML) e permite fazer a simulação de trajectórias de voos, proporcionando a sua visualização sobre um modelo tridimensional do terreno.

Este modelo contém informação sobre a cartografia do terreno, rede viária, recursos hidrográficos e edificado, possibilitando ainda a fácil introdução de outros elementos caracterizadores do terreno. É possível adicionalmente e por síntese de voz, obter informações diversas sobre os locais de maior importância nas proximidades [Soares03]. Isto acontece dinamicamente, à medida que as coordenadas lidas se aproximam dos locais referenciados. O utilizador tem ainda a possibilidade de fazer uma visualização de todo o terreno modulado e consultar por sua iniciativa todas as informações disponíveis.

AQUISIÇÃO DE COORDENADAS

Desenvolveu-se um módulo de geo-referenciação com elevada precisão, e uma frequência de amostragem adaptável às características do voo. Este sistema prima pelo elevado nível de integração, conseguindo dimensões e peso reduzidos, (30x50 mm e 20 gr) o que lhe confere grande adaptabilidade para diversas aplicações, figura 3.



Figura 3 – Módulo de localização GPS desenvolvido.

Este módulo monitoriza o voo, sendo guardadas as coordenadas do trajecto (latitude, longitude e altitude) e outras informações acerca do sinal GPS, para análise posterior. No final do voo, o equipamento é retirado do meio aéreo e a informação é transferida para o computador.

Os valores obtidos com este dispositivo são idênticos aos de qualquer sistema GPS comercial que utilize o protocolo de comunicação *National Marine Electronics Association* (NMEA). Para comprovar a sua eficácia foram efectuadas diversas experiências, a nível do solo, que demonstraram um erro inferior a 3m, figura 4, mesmo em fracas condições de recepção do sinal [Nogueira05].



Figura 4 – Mapa com trajectória recolhida em viagem experimental de ~40Km.

MODELAÇÃO DO TERRENO

O primeiro passo na implementação do Modelo Virtual de Voo foi a construção da malha tridimensional com a informação altimétrica. Para tal, recorreu-se a um conjunto de informações geográficas, disponibilizadas em formato *Shape File* no sítio *Web* do Instituto do Ambiente [InstAmb]. Estes ficheiros possuem em formato vectorial elementos sobre toponímia, distritos, altimetria, rios, entre outros.

O mapa a três dimensões (3D) modelado em VRML, foi construído com base no recurso *ElevationGrid* [lighthouse3d] que permite dar relevo a uma matriz de cotas, correspondentes aos nós da matriz criada (no presente caso a matriz tem uma resolução de 120x120m). Sobre a malha 3D é sobreposta uma textura, cujas cores são automaticamente atribuídas de acordo com a altitude nesse ponto.

Para obter a matriz necessária ao VRML, os ficheiros vectoriais foram tratados por duas aplicações de Sistema de Informação Geográfica (SIG): ArcView GIS 3.2 e Idrisi Kilimanjaro [esri, idrisi].

O primeiro passo consistiu em recortar nos mapas vectoriais uma área a analisar, pois o tratamento desta informação exige elevado processamento. Na aplicação apresentada, a área tratada foi um sector no norte de Portugal, englobando as cidades de Vila Real a Porto (figura 5). Seguidamente criou-se uma *Triangulated Irregular Network* (TIN) com base na informação das cotas. Esta TIN foi convertida para uma *Grid* e seleccionou-se a precisão desejada (neste caso 120*120m). Fez-se o cálculo dos declives e exposições do terreno. Atribuíram-se cores à imagem, de acordo com a altitude, numa gama de 15 cores. Nessa fase exportou-se a imagem *raster* e a matriz de cotas para um ficheiro do tipo .txt de forma a poderem ser modeladas em VRML.

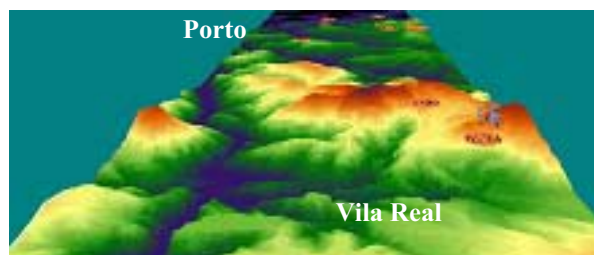


Figura 5 - Malha 3D vista de Este para Oeste.

MODO DE FUNCIONAMENTO

O menu inicial da aplicação, figura 6, permite ao utilizador seleccionar uma de 3 opções: mudar a imagem de textura da malha de elevação do terreno, ver as informações das principais cidades que se encontram catalogadas e iniciar a simulação do voo.

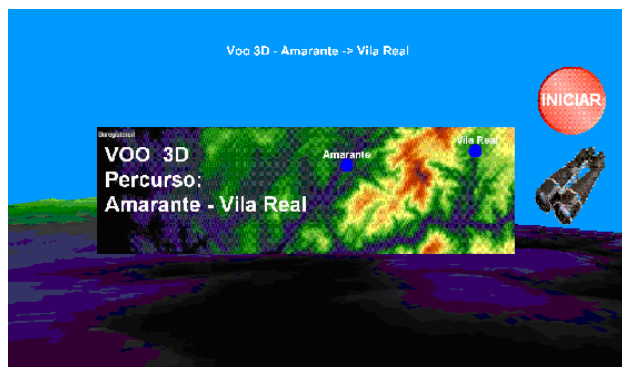


Figura 6 - Menu principal.

Ao seleccionar o ícone “Alterar Mapa”, figura 7, a textura da malha é alterada entre cinco texturas disponibilizadas. Cada uma delas pretende adicionar ao projecto um determinado tipo de informação, são elas:

- Rota.png
- Rios.png
- Autoroute.png
- Concelhos.png
- Cotas.png



Figura 7 - Alterar mapa.

O *MovieTexture* visualizado no centro do ecrã, permite ao utilizador conhecer o terreno, tal como é apresentado na *label* “Visita ao País” que surge ao colocar o cursor sobre a imagem. Ao escolher esta opção são disponibilizados cinco *ViewPoints* que agrupam conjuntos de marcadores dos locais mais importantes. Ao colocar o cursor sobre cada um desses marcadores são visualizados: o nome desse local, o distrito a que pertence e uma fotografia. Seleccionando o marcador, é reproduzido um som com o nome do mesmo.

Em cada *ViewPoint* encontram-se duas setas que funcionam como *links*. A de cor azul permuta para o *ViewPoint* seguinte de visita ao país, a de cor verde redirecciona o utilizador para o menu principal.

O ícone abaixo ilustrado, figura 8, inicia a recriação da trajectória. Esta trajectória é determinada por um conjunto de coordenadas lidas no GPS que são convertidas em distância relativamente a um ponto na imagem, origem do sistema de coordenadas em VRML. A aplicação recebe as coordenadas importadas do GPS, no sistema NMEA [Garmin, gpsglobal] e cria um ficheiro .txt com as coordenadas de saída adaptadas ao sistema de coordenadas do VRML.



Figura 8 – Iniciar voo.

Durante a viagem, o utilizador é “transportado” por um *ViewPoint* associado ao meio aéreo, colocado num ponto superior, segundo um ângulo de 45°. Desta forma tem-se um visionamento da orografia do terreno óptimo, figura 9. Por outro lado, esta perspectiva não é a ideal para se ter conhecimento do local exacto de passagem da trajectória devido ao ângulo de visão e por se encontrar muito afastada das coordenadas propriamente ditas.

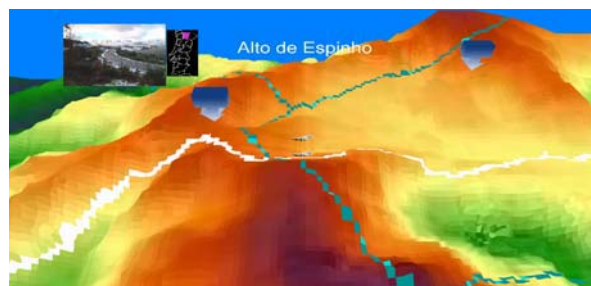


Figura 9 - Vista sobre a Serra do Marão.

Foi por isso disponibilizado um outro *ViewPoint* localizado exactamente por cima do meio aéreo a uma altitude inferior. Nesta perspectiva perde-se um pouco a noção de relevo do terreno, mas consegue-se visualmente perceber a localização em cada instante com maior precisão, figura 10. Para alternar entre estes dois *ViewPoints*, basta, durante a viagem, seleccionar o meio aéreo.



Figura 10 - Vista sobre o meio aéreo.

A cada marcador de local encontra-se associado um *ProximitySensor*. Este permite uma visualização da informação disponibilizada, à medida que o meio aéreo se vai aproximando dos vários locais.

Para conferir um maior realismo, o modelo virtual foi completado em pontos estratégicos com alguns sons representativos da realidade sonora desses locais tais como, o ruído do vento em regiões montanhosas ou da água em movimento simulando o efeito dos rios.

Concluída a viagem, o meio aéreo chega ao destino e o utilizador tem ao seu dispor mais um *link* que lhe permite navegar novamente para o menu inicial.

Para uma melhor percepção das trajectórias descritas é disponibilizada, em alguns dos mapas, a projecção sobre o solo da rota aérea descrita.

CONCLUSÃO

A componente geoespacial imbuída neste projecto pode ser extremamente vantajosa para a análise de trajectórias aéreas. A apresentação visual e auditiva das mesmas aumenta o realismo da aplicação, despertando o interesse sobre os locais mais relevantes.

Tendo em conta o desempenho do hardware, a possibilidade de, por um lado, adicionar marcadores associando-lhes imagens, sons, legendas ou mesmo vídeos, e por outro, aplicar texturas adequadas constituem uma mais valia desta aplicação. O visionamento prévio de trajectórias aéreas ajusta-se à simulação de rotas turísticas e de desportos aéreos. Pode permitir ainda o estudo de rotas de aves e associar padrões de voo a características como a orografia, a temperatura, a humidade, entre outros, pela sobreposição de diferentes mapas.

BIBLIOGRAFIA

[Biro02] Dora Biro, Tim Guilford, Giacomo Dell’Omo and Hans-Peter Lipp, “How the viewing of familiar landscapes prior to release allows pigeons to home faster: evidence from GPS tracking”, Department of Zoology, University of Oxford, Division of Neuroanatomy and behaviour, Institute of Anatomy, University of Zurich, 2002.

[Gould80] Gould, J.L., The case for magnetic sensitivity in birds and bees (such as it is). American Scientist 68, 256-267, 1980.

[Igor00] Igor Steiner, Clemens Bürgi, Stefan Werffeli, Giacomo Dell’ Omo, Paolo Valenti, Gerhard Troster, David P. Wolfer, Hans-Peter Lipp, “A GPS logger and software for analysis of homing in pigeons and small mammals”, Institute of anatomy and Center for Neuroscience, University of Zürich, Institute for Electronics, Swiss Federal Institute of Technology, ublox AG, 2000.

[Nogueira05] Nogueira, Cesário; Costa, Raúl; Valente, António; Soares, Salviano; Morais, Raul; Coelho, António; Cabral, Manuel, “Dispositivo de Baixo Peso para Estimação de Trajectórias Aéreas de Aves”, Universidade de Trás-os-Montes e Alto Douro, 2005. (Aceite para publicação na CAIP’2005 <http://home.utad.pt/~caip2005/>.)

[Soares03] Salviano Soares, Carlos Seródio, António Coelho, Pedro Mestre, Pedro Pinto; “Audio Interface for Real Time Navigation”, ConfTele 4th Conference on Telecommunication, Aveiro, Junho de 2003.

[idrisi] <http://www.clarklabs.org>

[Garmin] <http://www.Garmin.com/>

[esri] <http://www.esri.com/>

[gpsglobal] <http://www.gpsglobal.com.br/>

[InstAmb] <http://www.iambiente.pt/>

[lighthouse3d] <http://www.lighthouse3d.com/vrml/>

Índice de Autores

Almeida, Daniel	219	Lourenço, Daniel	167
Amaro, Carlos	223	Machado, Rodrigo	215
Ângelo, Pedro	151	Madeira, Joaquim	1, 95, 111, 145
Araújo, Bruno	191	Magalhães, Hélder	29
Barrenho, Filipe	133	Magalhães, Luís	119
Bernardes, Paulo	95	Marcos, Adérito	41
Bessa, Maximino	85, 223	Marcos, João	155
Bidarra, Rafael	167	Marreiros, Filipe	41
Birra, Fernando	19	Martins, Paulo	145
Buf, Hans du	179, 219	Martins, Tiago	133
Cabral, Manuel	223	Mealha, Óscar	185
Carmo, M. Beatriz	79	Morais, Raul	223
Cláudio, Ana Paula	79	Moreira, Paulo	145
Coelho, António	85, 223	Nogueira, Cesário	223
Coelho, Helder	79	Nunes, José	185
Correia, Nuno	133	Nunes, Samuel	219
Costa, Raul	223	Oliveira, Bruno	151
Couto, Pedro	119	Oliveira, Eunice	9
Cruz, José Bulas	85	Oliveira, Pedro	167
Cruz, Mário	145	Pereira, Filipe	197
Cunha, João D.	79	Pereira, João	57
Cunha, Manuel Bernardo	145	Pereira, José	57
Dias, Filipe	161	Pinto, Pedro Melo	119
Dias, Miguel	49, 155	Pinto-Albuquerque, Maria	79
Dias, Paulo	145	Pocinho, António	49
Ferreira, Alfredo	203, 209	Prata, Ana	129
Ferreira, Carlos	111	Ramião, Ricardo	209
Ferreira, Fernando	85	Rodrigues, João	179
Figueiredo, Frederico	215	Rodrigues, Nuno	105
Fonseca, Manuel J.	197, 203, 209	Romão, Teresa	133
Gabriel, Ronaldo	119	Santos, Beatriz Sousa	1, 111, 145, 185
Gomes, Mário Rui	129	Santos, Manuel Próspero	19, 105
Gomes, Rui	145	Silva, Fernando	139
Gonçalves, Alexandrino	139	Silva, Frutuoso	173
Gonçalves, Daniel	69	Silva, Samuel	1, 111
Gonçalves, Gil	105	Silvestre, Miguel	79
Gonçalves, Nuno	215	Simão, António	209
Guerreiro, Tiago	191	Soares, Salviano	223
Jorge, J.	69, 161, 191, 197, 203, 209, 215	Sousa, António Augusto	85
Lobão, Susana	129	Teixeira, José Carlos	9
Lopes, Adriano	105	Valente, António	223
Lopes, António	155	Zamfir, Florin	185

13º Encontro Português de Computação Gráfica
Vila Real, 12-14 Outubro 2005
Actas do Encontro

Adriano Lopes
(editor)

Publisher
Grupo Português de Computação Gráfica
(Eurographics Portuguese Chapter)
Edifício INESC Porto
Campus da FEUP
Rua Dr. Roberto Frias, 378
4200-465 Porto, Portugal

ISBN 972-98464-6-4

Impressão Outubro 2005