Sistemas de Informação N.º 17 / 83-90

Collaborative Multimodal Authoring of Virtual Worlds

Vítor Sá, Adérito F. Marcos •
Filipe Marreiros, Vítor Sá ••
Adérito F. Marcos •••

Resumo

O objectivo deste trabalho é a criação de mundos virtuais por equipas de trabalho com recurso a técnicas avançadas de interacção, tais como comandos de fala e gestos, e a dispositivos móveis. As características físicas dos ambientes de realidade virtual conduzem-nos a diferentes tipos de interfaces de utilizador. Contudo, encontramos certa utilidade no clássico paradigma WIMP, que é utilizado recorrendo a um PDA, para complementar as possibilidades de interacção. Em termos de descrição da aplicação utilizamos a linguagem de modelação para realidade virtual – VRML. Um dos principais objectivos consiste na separação/combinação de arquivos VRML para permitir o trabalho individual a cada um dos elementos da equipa, e posteriormente a combinação de resultados para produzir o mundo virtual global.

Abstract

This work aims the creation of virtual worlds by working teams with resource to advanced interaction techniques, such as speech commands and gestures, as well as the resource to portable devices. The physical characteristics of virtual reality environments lead us to a different kind of user interfaces. However, we found some usefulness on the classic WIMP paradigm, which we also apply by using a PDA, in order to complement the interaction possibilities. In terms of application description we use the Virtual Reality Modeling Language. One of the main goals is the separation/combination of VRML files in order to allow the work to be cared out individually by the members of the team, and later combine the results to produce the global virtual world.

1. Introduction

The present article describes an experimental work in the area of the creation of virtual worlds by working teams with resource to advanced interaction techniques, such as speech commands and gestures, as well as the resource to portable devices. The use of these last ones becomes indispensable due to the physical characteristics of the environment, large projection screens or workbenches, in which it's also useful to have access to WIMP interfaces in order to complement the user interaction possibilities.

The prototype under development is very flexible mainly because: users can use several modalities, in whatever combination they desire (multi-modality); users can change modalities, even during a single interaction - the more convenient for a specific situation (flexi-modality); users may be using a workstation, standing up at

• University of Minho - Campus de Azurém - P-4800-058 Guimarães - vitor.sa@dsi.uminho.pt
•• Computer Graphics Center - Fraunhoferstr. 5 - D-64283 Darmstadt - fmarreiros@zgdv.de
••• Computer Graphics Center - R. Teixeira de Pascoais, 596 - P-4800-073 Guimarães - aderito.marcos@ccg.pt

large displays or in a mobile situation, and can still continue they work somehow (multi-machine).

In order to exemplify possible scenarios of application, imagine the user creates the virtual world by using speech and direct manipulation in a 3D environment:

- There are things which are difficult to achieve with speech commands, e.g. precise measurements;
- We would like to add the vocabulary grammar with new concepts, e.g. if we are building the geometry of a chair and at the end we want to refer to it as "chair";
- We would like to set the parameters of a color that is not on the grammar, or we would like to refine some other world properties.

This kind of things can be done with our solution in the following ways:

- Separately, in a workstation without the need of a VR system apparatus;
- Collaboratively, by a group of persons that will perform their individual tasks to accomplish a common goal – the creation of the global virtual world;
- Complementally, by using a handheld device without leaving the workbench place, and seeing automatically the consequent results.

To achieve all these goals, several technologies were used and we start by presenting them on the next section. On section 3, we present the application functionality and the encountered solution. We finish with some conclusions and possible future improvements.

2. Involved technologies

2.1. Virtual reality

The virtual reality (VR) environment we used for our experiments is based on a workbench with a display surface of about 1.36 m * 1.2 m, which means a volume of interaction of about 3 m * 3 m * 1.5 m above and in front of the table (width * depth * height). This means that the user can interact with the virtual world within this volume, where his positions and movements have to be precise and efficiently tracked. In order to "run" the virtual table we used the "Avalon" system [Behr 2003], developed at the Computer Graphics Center in Darmstadt.

Fig. 1 – Workbench environment

"Avalon" uses the Virtual Reality Modeling Language [VRML] with some extensions as scene description language. The use of VRML has several advantages: the interface is well defined by a non-proprietary, platform

and company independent standard (ISO/IEC 14772), the application developer can use a wide range of VRML modeling tools and, very important in our case, it is possible to display the same VRML file on a 2D browser on a desktop PC using 2D input devices, as well as on a VR system with 3D displays using 6D input devices. For this purpose, "Avalon" extends the concepts of the VRML "Viewpoint" and "TouchSensor" nodes: the first one still sets the global viewing direction into the scene, but the view frustum is modified according to the current head position and orientation of the user; the second node reacts on the collision of a 3D cursor with objects in the 3D scene [Sá 2002].

The user controls the 3D cursor with his hand that gets tracked by a tracking system. By this way it is possible to interact with the virtual scene in a very simple and natural way by pointing at objects.

2.2. Gesture recognition

Regarding the interaction mechanisms, for the direct manipulation (gesture modality) of the virtual world we are using the tracking system "EOS", also developed at the Computer Graphics Center [Schwald 2002]. The system uses a stereoscopic approach allowing natural interaction (with 6DOF) within the virtual world via a pointing gesture. This is combined with a speech recognition component to enhance the independent uni-modal inputs by an integrated multi-modal approach.

We are using video-based tracking with infrared beacons and retro reflective markers, which allows good real-time results even without special light conditions. By this way, the VR system keeps track not only of the user's head position, to render the images in the correct perspective, but also of the user's interaction device, the way the user has to perform direct manipulations.

2.3. Mobile device

Our mobile device consisted on an iPAQ Pocket PC with wireless network access. We made our experiments using Virtual Private Network (VPN) technology, inside the *firewall of our organization*. The mobile unit worked as a VPN client connected to a VPN server; the gateway to the other computers behind it on the subnet.

In terms of application development, we adopted the Java technology, taking advantage of its portability, network support and multithreading. We have programmed in conformance with Java 2 Micro Edition (J2ME), and our iPAQ was equipped with Jeode Java Virtual Machine [Jeode].

The J2ME defines two major categories of components: configurations and profiles [J2ME]. The components we needed were the Connected Device Configuration (the one more suited for high-end PDAs), the Foundation Profile and the Personal Profile. The first component is a vertical set of APIs that provides the base functionality, such as memory footprint and network connectivity. The rest constitute a horizontal set of high-level APIs providing access to the device capabilities ranging from I/O to Graphical User Interface.

Fig. 2 – Graphical interface of the mobile unit

2.4. Speech recognition

We have used the *Java Speech API* (JSAPI) that defines a software interface to state-of-the-art speech technology. The two core speech technologies (recognition and synthesis) are supported by JSAPI. There are several engines on the market with very reasonable accuracy for both technologies, e.g. IBM Via Voice and Microsoft Speech, which are speaker-independent and fully-continuous.

The goal of speech recognition is to transform the user voice (an audio data stream) into a text string. This text string respects specific grammar rules and includes only syntactically correct words. The closer we get to fully unrestricted natural language, the more difficulties we encounter. The use of an artificial language of special commands fulfills our user requirements. Using the Java Speech Grammar Format (JSGF), we built a command-and-control recognizer covering several types of interaction: objects generation, attributes changing, movements and inspections.

2.5. Multimodal interaction

In terms of integration we follow the classical three-level approach, with its lexical, syntactic and semantic layers; a rather straightforward adoption of the LANGUAGE model described by Foley and Van Dam [Foley 1982]. In our context, the lexical layer corresponds to the binding of hardware primitives to software events, in which temporal issues are of main importance; the syntactic layer is where the sequencing of events is performed, that is the combination of data to obtain a complete command, and; the semantic layer is related to the functional combination of commands in order to generate new, more complex ones. So, in a multimodal point of view, each individual modality can be in a stage considered semantic by itself, but without having any meaning in the overall

context – this means a correct multi-modal syntax, without any meaning or semantic.

In terms of integration of the different modalities we are using a semantic fusion approach, considering the modalities in a so-called late fusion state. This is appropriate when the modes differ substantially in the time scale characteristics of their features [Wu 1999]. The functional combination of commands in order to generate new, more complex ones, rather trivial at the moment, is being performed by methods such as state machines or parsers (note in our context that state machines are parsers for regular grammars).

3. Application functionality

The application goal is the creation of a virtual world by a group of persons, using different devices dependent on availability or specific needs. The result is textually described in VRML, which means the need of some separation/combination mechanism of this kind of files.

The interaction possibilities are different depending on the system we are working on, since we have commutations between 3D and 2D environments. In the desktop system the user looks at a 2D projection of the 3D world that is independent of his actual head position and orientation. This limits the possibilities of interaction with the 3D world. For example, the VRML "TouchSensor" node detects the virtual object the user is pointing at by shooting a ray into the scene, while in 3D this is done by collision, as we mentioned previously.

Opposed to the kind of work that is done at the workstation, the mobile device is used to send information to the VR system, with immediate reaction. It is not possible to do much more with this type of device, but the possibility of having it on the palm of the hand is very useful to complement authoring tasks.

3.1. Authoring

From a implementation point of view, the range of operations that is possible to perform are the accomplishment of a set of VRML nodes, with a root element, to a specific node already existing in the current world, the counterpart remove operation, adding and removing routes, sending time stamps (for the time-dependent nodes) and sending/receiving events.

The dialog manager, rather trivial, is based on the parse returned by the speech recognizer. We encode meta-level information about the utterances, using the tag facility of the Java Speech Grammar Format. This information together with the gesturing selections is then used to determine the action that is requested by the user and what objects will be affected.

3.2. Collaboration

In terms of the multi-machine characteristic, the work can be carried out in several devices, thanks to the distributed file system used - Samba Unix. But one of our goals is to allow the combination and separation of the work. Has referred earlier we pretend that a virtual world may be constructed by several users. One immediate question arises: how do we separate the space? One possibility is to attribute portions of the space to the several users. Other possibility would be to assign objects to the users, i.e. each user has its own set of

objects. Many questions would arise using any of the approaches taken. To give an example lets consider that a user creates a light source; this light may affect other users, and problems occur when trying to separate and combine the created virtual world. In our prototype we considered 8 users, all with the same amount of space, like is presented in the figure:

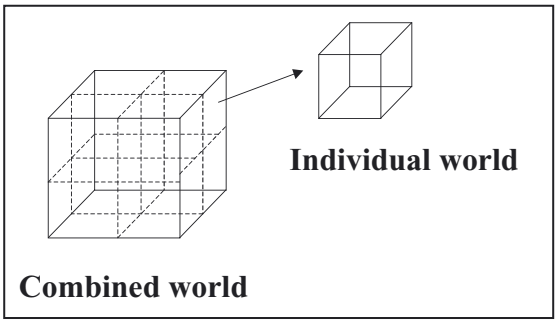

Fig. 3 – Spatial world subdivision

We still didn't consider the cases were objects share several individual worlds. Has we referred earlier, further work has to be carried our.

Concerning the combination and separation of VRML files, by parsing them we can find out where an object is located in the space, and this way we can assign the objects to the users. So we can create a VRML file for each user. All the users have to specify the region of space that he/she owns, and the input and output files. So to make a combination and a separation we have the following commands with the format specified:

- combine <nsr> <fiw> <fcw> <ipcw> for combination;
- separate <nsr> <fiw> <fcw> <ipiw> for separation,

where:

- nsr is the identifier of the selected region, in our case we can select 1 of 8 possible regions;
- fiw is the name, including the path, of the file containing the individual world;
- fcw is the name, including the path, of the file containing the combined world;
- ipiw is the IP address of one of the machines that we want the individual world to be saved to;
- ipcw is the IP address of one of the machines that contains the combined world. If the file doesn't exist a new file (world) will be created.

Has can be noticed we are using communications that will allow our data to be saved in a desired machine. This is done using a TCP/IP socket connection. To allow the desired actions to be performed, we have running a TPC/IP server on the several machines that take part of the process of creating the world. These servers are waiting for messages from the clients. These messages are sent when the referred commands are used. The clients (commands) use their input parameters to produce the desired message to be sent.

This easy exchange of information is a relevant benefit in terms of collaboration. The users have an easy way of combining there individual work. Furthermore, if the combined world is changed the users can also get that information, provided by the person(s) that are managing the combined virtual world.

4. Conclusion and future work

This experimental work is still in its early stage. As referred earlier one of the problems is the world combination and separation. Regarding this issue we pretend to experiment other techniques besides the one presented here.

We also plan to expand the type of possible devices, namely for mobility and new interaction possibilities. In order to take the maximum advantage of the mobile devices, we need to have an easy exchange of information between mobile and static devices. This way, if the user pretends to continue the work in a mobile device he just has to fetch the file from the static one, if this is the case. For the file exchange we are using the TCP/IP sockets connection. We are also considering approaches similar to the Coda File System [Coda] for persistent caching of files, which means that files recently used exist in a local drive.

About the new interaction possibilities, we pretend to explore them in order to facilitate the creation of the virtual worlds. Regarding the gesture modality, there is work being done to integrate natural gesturing [Sá 2001], instead of having some special device as is the case of the EOS system.

5. Acknowledgements

This work is partially supported by "Fundação para a Ciência e a Tecnologia", through a scholarship in the context of the Information Society Operational Program (reference PRAXIS XXI/BD/20095/99).

We would like to thank the Computer Graphics Center (ZGDV) in Darmstadt, namely the Visual Computing Department, for the facilities provided in order to test the work presented here.

References

Behr, J., Dähne, P.: "AVALON: Ein komponentenorientiertes Rahmensystem für dynamische Mixed-Reality Anwendungen", In: Thema Forschung, 1, pp. 66-73, Germany, 2003.

Coda, Online reference http://www.coda.cs.cmu.edu/ .

Foley, J. et al.: Fundamentals of interactive computer graphics. Addison-Wesley, Reading, MA, 1982.

Jeode, Online reference http://www.esmertec.com/products/jeode_dell_runtime.shtm

J2ME, Online reference http://wireless.java.sun.com/configurations/articles/cdc/ .

Schwald, B., Malerczyk, C.: "Controlling Virtual Worlds Using Interaction Spheres", In: Vidal, Creto Augusto (Ed.) u.a.; Brazilian Computer Society (SBC) u.a.: Proceedings of the 5th Symposium on Virtual Reality, pp. 3-14, Fortaleza, CE, Brazil, 2002.

Sá, V., Malerczyk, C., Schnaider, M., "Vision-Based Interaction within a Multimodal Framework", Proceedings of the 10th Conference of the Eurographics Portuguese Chapter, Lisbon, October 2001. Http://virtual.inesc.pt/virtual/ 10epcg/actas/pdfs/sa.pdf

Sá, V.; Dähne, P.: "Accessing Financial Data through Virtual Reality", In: Figueiredo, Antonio Dias de (Ed.) u.a.: Proceedings of 3ª Conferência da Associação Portuguesa de Sistemas de Informação, Coimbra, 2002.

VRML Specification on-line, http://www.vrml.org/.

Wu L., Oviatt S., Cohen P.: "Multimodal Integration – A Statistical View", IEEE Transactions on Multimedia, 1(4):334-341, 1999.