UNIVERSIDADE ABERTA
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

FACIAL ANALYSIS WITH DEPTH MAPS AND DEEP LEARNING

Paulo Miguel Franco Correia de Brito

Doutoramento em Ciência e Tecnologia Web

(doutoramento em associação)

Tese orientada pela Professora Doutora Elizabeth Simão Carvalho

2018

DECLARAÇÃO

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE, APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade Aberta, ____ / ____ / _____

Assinatura: _____

Acknowledgments

Many people contributed to this work. I would like to use these lines to acknowledge every person and institution that contributed to its elaboration.

First of all, I would like to thank my supervisor, Professor Elizabeth Simão Carvalho, from Department of Science and Technologies of Universidade Aberta, for giving me the opportunity to do this project and for their guidance and encouragement.

Finally, I would like to thank my family and friends for their support especially to my wife, for all her encouragement and comprehension.

Paulo Brito

Abstract

Collecting and analyzing in real time multimodal sensor data of a human face is an important problem in computer vision, with applications in medical and monitoring analysis, entertainment, and security. However, due to the exigent nature of the problem, there is a lack of affordable and easy to use systems, with real time annotations capability, 3d analysis, replay capability and with a frame speed capable of detecting facial patterns in working behavior environments. In the context of an ongoing effort to develop tools to support the monitoring and evaluation of human affective state in working environments, this research will investigate the applicability of a facial analysis approach to map and evaluate human facial patterns. Our objective consists in investigating a set of systems and techniques that make it possible to answer the question regarding how to use multimodal sensor data to obtain a classification system in order to identify facial patterns. With that in mind, it will be developed tools to implement a real-time system in a way that it will be able to recognize facial patterns from 3d data. The challenge is to interpret this multi-modal sensor data to classify it with deep learning algorithms and fulfill the follow requirements: annotations capability, 3d analysis and replay capability. In addition, the system will be able to enhance continuously the output result of the system with a training process in order to improve and evaluate different patterns of the human face. FACE ANALYSYS is a tool developed in the context of this doctoral thesis, in order to research the relations of various sensor data with human facial affective state. This work is useful to develop an appropriate visualization system for better insight of a large amount of behavioral data.

Key-words: pattern recognition, facial analysis, deep learning algorithms

Resumo

A recolha e análise sequencial de dados multimodais do rosto humano é um problema importante em visão por computador, com aplicações variadas na análise e monitorização médica, entretenimento e segurança. No entanto, devido à natureza do problema, há uma falta de sistemas acessíveis e fáceis de usar, em tempo real, com capacidade de anotações, análise 3d, capacidade de reanalisar e com uma velocidade capaz de detetar padrões faciais em ambientes de trabalho. No âmbito de um esforço contínuo, para desenvolver ferramentas de apoio à monitorização e avaliação de emoções/sinais em ambiente de trabalho, será realizada uma investigação relativa à aplicabilidade de uma abordagem de análise facial para mapear e avaliar os padrões faciais humanos. O objetivo consiste em investigar um conjunto de sistemas e técnicas que possibilitem responder à questão de como usar dados de sensores multimodais para obter um sistema de classificação para identificar padrões faciais. Com isso em mente, foi planeado desenvolver ferramentas para implementar um sistema em tempo real de forma a reconhecer padrões faciais. O desafio é interpretar esses dados de sensores multimodais para classificá-los com algoritmos de aprendizagem profunda e cumprir os seguintes requisitos: capacidade de anotações, análise 3d e capacidade de reanalisar. Além disso, o sistema tem que ser capaze de melhorar continuamente o resultado do modelo de classificação para melhorar e avaliar diferentes padrões do rosto humano. A FACE ANALYSYS, uma ferramenta desenvolvida no contexto desta tese de doutoramento, será complementada por várias aplicações para investigar as relações de vários dados de sensores com estados emocionais/sinais. Este trabalho é útil para desenvolver um sistema de análise adequado para a perceção de grandes quantidades de dados comportamentais.

Palavras-chave: reconhecimento de padrões, análise facial, algoritmos de aprendizagem profunda

Contents

1 Introduction 1

1.1. Motivation 2

1.2. Goals, challenges and main contributions 3

1.3. Research Methodology 6

1.4. Structure of the document 9

2 Literature Review 10

2.1. Human Computer Interface 10

2.2. Facial Tracking 13

2.3. 3D geometric information in real time 15

2.4. Facial expressions detection 19

2.5. Affective Computing 26

2.6. Conclusions 29

3 Experimental Results 30

3.1. Facial analysis exploratory research 30

3.2. Proposed method 35

3.3. Data Analysis and tests 44

3.3.1 First phase 44

3.3.2 Findings 51

3.3.3 Second phase 53

3.4. Conclusions 58

4 Main results and future work 63

5 Bibliography 66

6 Appendix 72

6.1. Appendix 1 73

6.2. Appendix 2 75

6.3. Appendix 3 78

6.4. Appendix 4 80

6.5. Appendix 5 82

List of Figures

Figure 1 – HCI is a multidisciplinary scientific field [16] 12

Figure 2 – Kinect fusion algorithm 18

Figure 3 – Universal Facial expressions 24

Figure 4 – Paul Ekman training application for micro expressions 25

Figure 5 – Detect engagement through computer vision techniques 27

Figure 6 – Application of Affective Computing in e-learning 28

Figure 7 – Prototype system interface 31

Figure 8 – System capturing process in Kinect studio v2.0 32

Figure 9 – Prototype 3D point visualization and depth map with 3D eyebrows and nose points mapped into 33

Figure 10 – Front view of 3d cloud point from sensor 33

Figure 11 – Side view of 3d cloud point from sensor 34

Figure 12 – Flowchart of our proposed method (see for more detail Appendix 5) 35

Figure 13 – Setup scenario with the Kinect sensor 36

Figure 14 – Kinect v2 sensor 37

Figure 15 – Facial patterns that users had to mimic (see for more detail Appendix 1 and 2) 37

Figure 16 – CNTK architecture 39

Figure 17 – LSTM model from CNTK used to train data [68] 40

Figure 18 – 3 screenshots of the application running with different configurations and evaluating sequences 42

Figure 19 – Line charts showing different numbers of sequences being evaluated..... 43

Figure 20 – Sum of the 3D vector direction angles per all the 15 frames. 47

Figure 21 – Sparkline charts of the sum of the Euclidean distance per all the 15 frames. 47

Figure 22 – The Euclidean distance and the 3D vector direction per frame and per subject in captured sequences 1, 2 and 3 48

Figure 23 – Line charts per subject concatenating all four patterns 50

Figure 24 – Range of ages and genders in dataset 54

Figure 25 – Control points: 7 –left eyebrow + nose; 21 – eye brows + nose + mouth; 8 - mouth 56

Figure 26 – Research cycle 64

List of Tables

Table 1 - Goals, challenges, and main contributions 4

Table 2 - Experimental validation models 8

Table 3 – Sequence with 15 frames - training data for CNTK engine LSTM network 41

Table 4 - Comparison of results between subjects (charts in figures 13 and 14) 58

Table 5 - Comparison between facial patterns - average gap values (Euclidean distance chart in figure 15) 59

Table 6 – Average comparison between facial patterns – Euclidean distance in a frame with 7 points: right eyebrow and nose 60

Table 7 – Average comparison between facial patterns – Euclidean distance in a frame with 8 points: mouth 61

Table 8 – Average ccomparison between facial patterns – Euclidean distance in a frame with 7 points: left eyebrow plus nose 62

List of Equations

Equation 1 - Euclidean distance (see for more detail Appendix 3) 45

Equation 2 - direction of the 3D movement of each control point (see for more detail Appendix 4)..... 46

1 Introduction

Facial expressions are currently used for inferring emotions, but they also can be used to indicate mental states. Some authors [1] looked at facial expressions while students worked with an online tutoring system and identified that frustration was associated with activity in the inner and outer brow raiser and dimple; confusion was associated with brow lowered, lip tightened and lip corner puller. Moreover, preliminary results by [2] suggested that high and low stressing situations could be discriminated based on the facial activity in the mouth and eyebrow regions. In addition, stress can be inferred using multimodal sensor data as stated in a survey [3].

Vast amounts of data from different sensors can be used to try to deduce human facial affective states. Interpreting this data in a meaningful way, is challenging.

Also it can be seen in [4, 5, 6] several techniques that can produce 3d facial data from a range of sensors allowing the 3d reconstruction of faces in real time. This data can be used later on to evaluate patterns in a facial dataset.

In this research, we apply several computer science techniques, such as visual analytics, deep learning and facial tracking, to a facial dataset in order to interpret a multimodal sensor data. We alternate between data analysis, to find out structures in the data, and visualizations, to gain insights. This exploratory analysis was aimed at finding relations between facial patterns and affective states and behaviors that can be measured with sensors.

1.1. **Motivation**

The human being indicates most of the time what is hidden through his eyes and facial expression [7]. Tools that do give support in terms of monitoring and evaluating these subtle changes in human's face expressions, will also introduce a higher degree of accuracy and refinement in the overall diagnosis procedure of the human affective state.

Through a randomized controlled human trial of a multi-modal monitoring task based intervention, we will try to obtain evidence, for best practice, in the use of multi-modal data sensor in the evaluation of human affective state. Thus, the intervention will be with a computer and a task based application. The task will be a set of emotional stimulus activities. The evidence will be obtained with a computer system solution that will be developed and implemented in the scope of the research component of this doctoral program.

The solution will encompass the following characteristic: a Monitoring System that will be able to capture facial information. This system will allow the visualization and detection of facial patterns. It will provide a visualization tool for the human changes in affective state, which is further associated with motivation and learning level[8]. Ultimately, the fusion of the monitoring data will deliver important qualitative and quantitative information regarding the human affective state in a specific environment place.

The architecture of the system must be capable, from a multimodal data stream, to do, in real time, annotations, 3d analysis, detect facial patterns in a working environment and be able to have replay capability.

1.2. Goals, challenges and main contributions

An important indicator of emotions is the expression of the human face. Affective computing is the study and development of systems and devices that can recognize, interpret, process, and simulate human emotions.

The goal of this research is to **investigate a set of systems and techniques that make it possible to answer the question regarding how to use multimodal sensor data to obtain a classification system in order to identify facial patterns**. The underlying assumption is that the knowledge of such systems may give a substantial base of work to develop a system that will be able to infer affective states related to working environment tasks.

We assume that it **is possible to classify a fusion of 2D and 3D data using deep learning algorithms and techniques**, allowing the combination of it to detect facial patterns.

The prototype system to be developed must be capable, from a multimodal data stream to do in real time: annotations, 3d analysis, detect facial patterns in working behavior environments and have replay capability. The software program should interpret the emotional state of humans and adapt its behavior to them, giving an appropriate response for those emotions.

The aim of this research is to investigate **a set of systems and techniques that make it possible to answer the question regarding the 3d and 2d data fusion in order to obtain detection of facial patterns that can be classified**.

The main challenging tasks here are:

1) How to classify a fusion of 3d and 2d data with deep learning?

2) How to combine 3d and 2d data to detect facial patterns?

Contributions: 1) Research into the relations of multimodal sensor data with affective states; 2) Focus on differences between users and how to cope with them in data processing and visualization; 3) Improve visualization techniques regarding large amount of behavioral data recorded with sensors

Problem: Vast amounts of data from different sensors can be used to try to infer human facial affective states, however, interpreting this data in a meaningful way is still very challenging.

Thesis: It is possible to classify a fusion of 3d and 2d data with deep learning and combine it to detect facial patterns.

Goal: Investigate a set of systems and techniques that make it possible to answer the question regarding how to use multimodal sensor data to obtain a classification system in order to identify facial patterns.

Goals	Tasks	Results and contributions
Review the state-of-the-art of deep learning algorithms and facial classification in real time	Bibliography review	Conceptual framework of concepts associated with facial classification and deep learning: - Recurrent Neural Networks - Facial patterns to evaluate - CNTK framework -Techniques and representations of data/classification
Review the state-of-the-art of Human Computer Interaction, Face Tracking and Affective Computing	Bibliography review	Conceptual framework of concepts associated with HCI, Face tracking and Affective Computing: - face tracking systems (Multimodal sensors - Kinect) - 3d and 2d facial data - Algorithms and techniques regarding HCI, Affective Computing and facial tracking
Design and concept of the application “Facial analysis”	Implementation of a prototype of the application “Facial analysis”	Architecture model of the application
Developing an application based on the design and concept		Application and dataset for validation and test
Application validation		

Table 1 - Goals, challenges, and main contributions

The first objective is to frame and clarify the concepts associated with deep learning algorithms and facial classification in real time used, which serve as a basis for suggesting a possible solution for the algorithm of the application proposed in this thesis. In addition to the various definitions presented, the facial patterns and data models will be analyzed, as well techniques used in deep learning like recurrent neural networks.

Finally, this objective is also associated with the revision of the state of the art in terms of facial classification techniques.

The second objective is to review the literature associated with the HCI (Human Computer Interface), Affective Computing and Face tracking knowledge areas, analyzing the current state in terms of applications, projects, tools and systems. This objective is also associated, in particular and as far as possible, with the identification of the current state of the art of multimodal sensors and the relation between 3d and 2d data (human face).

The third objective is to develop the “Facial Analysis” application model and its respective architecture. Its design is based on the use of a 3d facial model that allows visualizing and analyzing the human face and associated distinct features data. At the visual level, the application should, as far as possible, use simple visual techniques to illustrate facial features and their classification.

The fourth objective is to implement a prototype for the application presented in this thesis. For its implementation, Visual Studio 2015 is used and Kinect SDK v2.0, both open-source libraries from Microsoft, and C# from Microsoft as the implementation language. In addition, Helix toolkit open source library is used for graphic visualization, Microsoft Cognitive Toolkit (CNTK) for deep learning algorithms and Kinect Studio v2.0 for record and inject to the application “Facial analysis”.

The fifth objective is to validate the application (through the prototype), verifying the performance of the algorithm in the visualization and analysis of the human face, and especially in the identification of facial expressions and in the analysis of facial patterns. Validation will be carried out with a survey/case study dataset of users data captured doing some tasks related with facial patterns.

1.3. **Research Methodology**

Most research and most research writing [9] involve the use of both forms of research and both forms of research sources. Generally speaking there are two major types of research models or research paradigms, and one that is a mix of both [10].

The Quantitative model, also known as traditional, positivist, experimental or empiricist. On the other hand, we have the Qualitative method, that is constructivist, naturalistic, interpretive, post-positivist or postmodern. And finally, the Mixed approach, that combines the collection and analysis of quantitative and qualitative data.

In this domain of computer vision we have two important types of research: the Action research (Qualitative model) and the Experimental research (Quantitative model).

The experimental research [11] is based on the control and isolation of every relevant condition which determines the events investigated, so as to observe the effects when the conditions are manipulated. The design of the experiments will be based on pre-experimental and quasi-experimental. The first makes unreliable assumptions despite the lack of control over variables. The second will assume that not all conditions can be controlled but the shortcomings are identified.

Basically experimental research is an attempt by the researcher to maintain control over all factors that may affect the result of an experiment. In doing this, the researcher attempts to determine or predict what may occur. The goal of experimental research is to establish cause-and-effect relationships between variables. We hypothesize that the independent variable caused the changes in the dependent variable. However, these changes or effects may have been caused by many other factors or alternative hypotheses. The purpose, therefore, of experimental designs is to eliminate alternative hypotheses. If we can successfully eliminate all alternative hypotheses, we can argue, by a process of elimination that the independent variable is the cause.

The action research is similar to experimental research although carried out in the real world rather than in the context of a closed experimental system – it involves small scale

interventions in the functioning of the real world and a close examination of the effects of such intervention. It is essentially an “on the spot” procedure designed to deal with a specific problem evident in a particular situation where no attempt is made to separate a particular feature of the problem from its context in order to study it in isolation. Also constant monitoring and evaluation are carried out and conclusions from the findings are applied immediately and further monitored.

Action research is suitable in ‘real’ concrete situations. It is useful where change and understanding is essential in a situation in which it is usually too difficult to control variables because the situation is concrete, complex and on-going. Action provides change and research provides understanding.

The methodology of action research is a cyclic form of self-reflective inquiry. It is used in social situations by the participants, to improve their own practice and the understanding of their practice and the situation. It can be described as people reflecting upon and improving their practice by tightly interlinking their reflection and action.

Some authors [12][13] refer to action research methodology as a cyclic progression from uncertain questions through uncertain methods to uncertain answers to less uncertain questions, methods and answers. The key elements are the cyclic acting and reflecting before acting again in a continuing response to learning outcomes from reflection. Action research provides confidence in outcome through the checking and refining of data and interpretations.

This key element of cyclic activity described above provides rigor. The way in which cyclic activity provides for rigor and thus validity is from:

- the gathering and interpretation of data in each research cycle before testing both data and interpretation in later cycles;
- seeking to disconfirm embryonic interpretations in each cycle;
- refining and analysing methods of reflection and action in each cycle;
- looking for divergent data to challenge other data already collected.

To address the target research questions and thereby implement this particular system, a mixed research strategy is used, where qualitative and quantitative methods were combined in a sequential fashion. The research was implemented in two sequential phases: (1) exploratory qualitative phase, and (2) quantitative phase. While the first phase was carried out throughout the initial stage of my doctoral program, the second phase took place in the second half of the second year and to be finalized in the middle of the third year.

The experimental research contributes to the second phase by initially simulating the trial of the system with some artificial data and then, in a second stage, with a trial of subjects (case study), as describe in Table 2. The system collected data [14] and evaluated the performance of the subject.

Method	Category	Description	Weakness
Simulation	Controlled	Monitor project in depth	Poor controls for later replication
Case study	Observational	Executed with real-time data	Data may not represent reality

Table 2 - Experimental validation models

1.4. Structure of the document

The organization of this document reflects the sequence of the tasks executed to conclude this work. It contains four chapters. The second one is dedicated to the state-of-the-art of the knowledge fields involved in this work. The third chapter focus on presenting the system and its tests, while the last one, points out the main conclusions and future work.

This first chapter summarizes the work, introducing its scope, goals, challenges and potential main contributions. It also presents the research methodology that is used in this thesis.

Chapter two 2 evaluates and describes the actual research in fields like Human Computer Interaction, Face Tracking, Facial expressions and Affective Computing. In this section will be introduced the principles, tools and mechanisms for the study of innovative methods, regarding the implementation of facial analysis systems. It covers the first and second goals listed in table 1.

Section 3 presents a discussion regarding some experimental research done in facial analysis, besides presenting the prototype system. It discusses several experiments that result in the final version of the facial analysis application and the datasets. It covers mainly the third and fourth goals from table 1.

Finally, chapter four is dedicated to discuss and to point out the most relevant conclusions achieved. It also proposes future work, based on the results of this work. This chapter is focused on the last goal in table 1.

2 Literature Review

“As scientists, we step on the shoulders of science, building on the work that has come before us - aiming to inspire a new generation of young scientists to continue once we are gone.” Stephen Hawking

In this literature review, we carried out an analysis of the state of the art of different topics such as Human Computer Interaction, Face Tracking, 3d geometric information in real time, Facial expressions detection and Affective Computing. We want to focus our contributions in a wide-ranging way towards facial evaluation. Thus, we pretend to introduce the principles, tools and mechanisms for the study of innovative methods, regarding the implementation of facial analysis systems. The state of the art reflects what has been done in these areas to solve the problem of using real time systems to detect facial patterns, and their practical application in evaluating and monitoring affective states in several environment places.

2.1. Human Computer Interface

It is the domain of science which consists in the study and application of methods that allow computers to "understand" the contents of an image. The term "understand" means the extraction from an image of the specific information that is useful for a particular purpose, for instance: indicate areas of the human body for surgery (e. g., if cancerous cells are detected in an image of a microscope) or control a process (for example, an industrial robot or an autonomous vehicle).

In the area of Human Computer Interaction (HCI) [60,61,15] researchers is devoted to the study of the interaction between people and computers. It is a multidisciplinary matter that interrelates computer science, arts, design, ergonomics, psychology,

sociology, semiotics, linguistics and related areas. The interaction between man and machine takes place via the end-user interface, which is consisted of software and hardware. It is used, for example, for the manipulation of computer peripherals and large machines such as airplanes and magnetic resonance imaging machines.

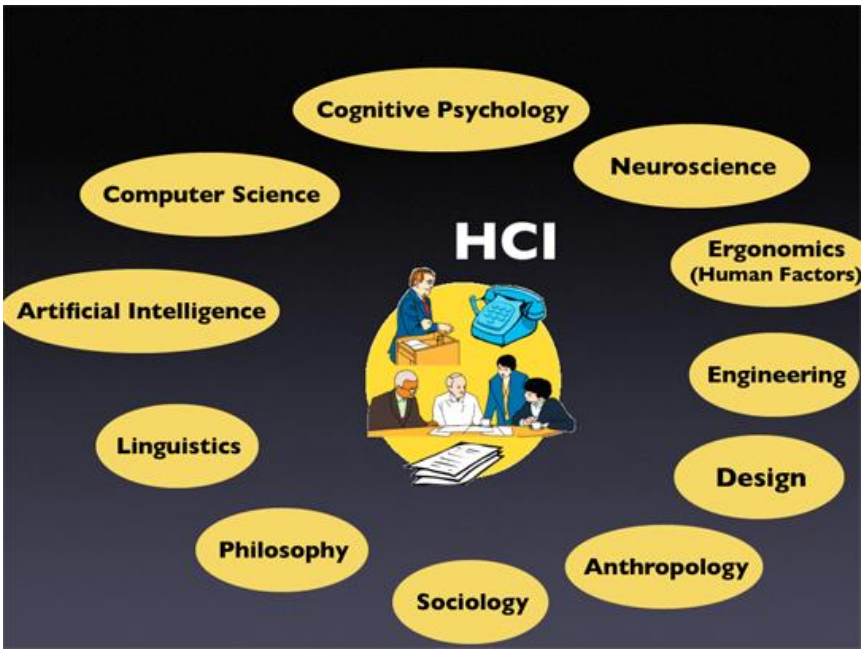
The human performance in the use of computers and information systems has been an area of much research and development that has expanded in recent decades. The studies make use of powerful computational tools for analyzing data collected in accordance with methods of experimental psychology. Other contributions also come from educational psychology, design and graphics, human factors and ergonomics, and most recently, anthropology and sociology. The end-user interfaces have produced significant success stories such as Google, Hotmail, or Yahoo.

The area of HCI has three periods:

- 1º age of facing human factors:
    - Study the user as a set of mechanisms for information processing;
    - Focus on the individual;
    - Creating guidelines for end-user interface development, formal methods and metrics based on systematic testing.
- 2º age of facing human factors:
    - Group focus;
    - Qualitative approaches rather than quantitative, prototyping and design context;
    - Comprehensive analysis and a general understanding of the phenomena relating to a person in a given environment.
- 3º age of facing human factor with the emphasis on cultural and aesthetic aspects:
    - Expanding the cognitive to emotional;

- Pragmatic factors of social experience;
- Pervasive mobile and small technologies;
- The technology goes beyond the limits of the work context and becomes part of the culture, the life and people's homes.

This multidisciplinary domain can be visualized in Figure **1**.



Source: www.deepamehta.de/docs/hci.html

Figure 1 – HCI is a multidisciplinary scientific field [16]

In short, for a user to interact with a given system it is necessary to control it and to know the state of the system. It is the human-computer interface that enables this interaction, and is represented by the layer that lies between the system and human end-user.

The purpose of the interface is fundamentally communicative. This communication is only possible with the existence of a kind of dialogue between the end-user and the machine, basically oriented in the way the end-user can interact with the machine. HCI is an inherent part of any work in the area of facial tracking, since any tool requires an interface with the end-user.

2.2. **Facial Tracking**

Images containing faces of people are very used in human-computer interaction and in human expression analysis. Because of this, several research groups are focusing on facial recognition, face tracking, assessment of posture and expression recognition. On average, the face detection algorithms can accurately detect faces in relation to their position in an image. A survey of the state of the art on the face detection can be found in [17][18].

To develop automated systems that analyze the information in facial images, face detection algorithms have to be efficient and robust. The aim of face detection, given a single image, is to identify all regions in the image that potentially contain a face, regardless of their dimensional position, orientation and lighting conditions. This problem is demanding because the faces are not rigid objects and have a high degree of variability in size, shape, color and texture.

The face detection and tracking is often the first step in applications, such as video surveillance and facial recognition. Location and tracking of human faces is a requirement of tracking facial and / or analysis of facial expressions, although often it is assumed that a face is available in the image in question.

Recently, due to improved processing performance of the personal computers, the massive use of webcams, the lower prices and the small size of devices like projectors or even 3d cameras, we can build new Human-Computer Interaction (HCI) systems, combining the strengths and new methods approaches of facial tracking. However, for a real-time tracking, many problems must be solved.

A major problem lies in the variety of existing features in the human's face, such as the color of the skin or the eyes, the existence or not of beard and glasses. Another problem is the system response time. Whether a system is able to recognize and make the tracking of a face, it is not able to do so, within a tolerable time frame (it can take several minutes). In addition, the tracking of the human's face has to be accurate and robust, for a real and meaningful practical use. These problems increase the difficulty for the recognition and tracking of facial features in real time.

Applications that use face tracking algorithms require a fast tracking and affordable system and most importantly, a very robust one. The confidence in these factors must be high enough to allow any user, the opportuneness and flexibility, to perform natural head movements.

Approaches to facial tracking systems can be separated into two classes: based on the images and based on the characteristics approach [19]. Each of them present different pros and cons.

In based on images, general face features are used such as skin color, head geometry and motion. These evidences are robust to perform rotations, scaling, and do not require high image quality. On the other hand, these approaches lack precision and therefore cannot be used to accurately pin point specific features.

For a smooth and accurate tracking, it is used the characteristics approach [20]. These approaches are based on tracking individual facial features. The tracking characteristics can be obtained accurately through the pixel, which allows a direct and precise mapping. The drawback of these approaches is that they usually require expensive cameras and high resolution. Moreover, these are not robust to head movements, especially rotation and scale.

Recently it was showed that the robustness of tracking based on individual characteristics of the face can be significantly improved, if instead of using features such as edges and corners of eyes, mouth and nostrils, they use features based on curvature of the nose [21]. This creates a new range of interesting possibilities for tracking facial features based on face.

In this context, the facial tracking could help to extract information from the 2d image that could be used to improve the reconstruction or the face analysis.

2.3. 3D geometric information in real time

A number of methods have been developed to obtain 3D geometric information, namely, stereo vision [22], shape from shading [23], shape from focus and defocus [24], laser stripe scanning [25] and time or color-coded structured light [26].

Among optical techniques, stereo vision is probably one of the most studied. However, finding the correspondence is fundamentally a difficult problem. Replacing one camera of a stereo vision system with a projector and projecting structured patterns onto the object can fundamentally solve the matching problem, which is called structured light system [27].

Structured light sensors [28] are composed by one or more cameras and one or more light sources. They have good performance in controlled environment ambient (industrial, medical), in weakly lighted environment (night vision, sub-marine vision) and in weakly textured environment (biometry, anthropometrics).

Regarding micro [29][7][30] and regular facial expressions [31][32], analyzing and studying data requires an accurate and a robust technique of digitally recording the movements of facial muscles. This process is achieved by sampling a person's face several times per second. The outcome of this process is a data set which describes the position of the facial landmarks on the person's face. Several techniques exist for capturing facial motion and it can be done in 2-Dimensional or 3-Dimensional spaces.

Vision-based techniques are used for marker-less feature extraction, where it is based on natural facial feature information, such as the corners of the mouth, eye brows and eyes. A marker-less feature system [33] detects the outer corners of the mouth, outer corners of the eyes and the inner corners of the eyebrows. The system detects certain features, such as the eyes and mouth using anthropometric face information and then uses the anthropometric ratio measurements to reconstruct a facial mesh and to analyze the facial expression. The main advantages of marker-less feature system is its low cost, no facial markers required on the person's face and can run in real-time.

However, issues such as low camera resolution, frame rate and lighting condition can affect the output of the system.

Another 2D Motion capture system is based on tracking facial markers, using a single low cost camera and a vision-based motion tracking software. The main advantages of a marker-based 2D motion capturing system is its low cost, can run in real-time efficiently, and perform an accurate tracking of the markers. However, it is does not support the capturing of 3D motions such as head rotation and tilt.

3D motion captured [34] data is obtained by capturing the facial markers on a person's face using a multi-camera system. To analyze the 3D facial motion data, we capture data sequences containing 3D information of a specific number of land markers placed on the person face. In particular, we capture the main universal expressions, such as happiness (smile), anger, fear, sadness, surprise, and disgust.

Some techniques presented in SIGGRAPH 2010 and 2011 showed high resolution 3d facial reconstructions capable of realistic geometric detail. [35] Present a new technique for passive and marker-less facial performance capture based on anchor frames. This method is based on a multi-camera setup and passive illumination that delivers a single, consistent mesh deforming over time to precisely match an actor's performance. The system is robust to expressive and fast facial motions, reproducing extreme deformations with minimal drift. A disadvantage of this system is not being in real time.

Other new approach is presented by [36] where a marker-based motion capture system is used to capture 3d facial performances and then is performed a facial analysis on the captured data. This facial analysis is able to determine the minimal set of face scans required for a precise facial reconstruction. Unfortunately, despite the use of marker based motion capture, the system is not able to run in real time, but has a very high mesh resolution and quality.

Another technique presented in [37] was non-intrusive, marker-less, with a low cost 3d capture device (e.g., a depth camera), that tracks the facial expression dynamics of the users in real time and maps them to a template model. The contributions of this method are centered on the face tracking that combines 3D geometry and 2D texture

registration in a systematic way with dynamic blend shapes previously generated from existing face animation sequences. Real time processing is facilitated by a reduced facial expression model that can be easily adapted to the specific expression and facial geometry. On the lesser positive mark are the blend shape weights between the raw 3d data and the template models used, which can tamper and misrepresent the micro expressions.

Other technique presented in [38,39] was the Kinect fusion system that is able to reconstruct in real time and in 3d the nearby environment capture by the Kinect device. This system enables a user holding and moving a standard Kinect camera to rapidly create detailed 3D reconstructions of an indoor scene. Only the depth data from Kinect is used to track the 3D pose of the sensor and reconstruct, geometrically precise, 3D models of the physical scene in real-time as seen in Figure **2**. The technique is based on GPU (Graphic Processing Unit) parallelism to obtain the quality and real time capability, and is also low cost. The algorithm is based on the following steps:

1. A depth map conversion will take place to convert the live depth map from image coordinates into 3D points and normals in the coordinate space of the camera;
2. The camera tracking phase, will allow the compute of a rigid 6DOF (Degrees of Freedom) transform to closely align the current oriented points with the previous frame, using a GPU implementation of the Iterative Closest Point (ICP) {Formatting Citation} algorithm;
3. Relative transforms are incrementally applied to a single transform that expresses the global pose of the device;
4. The volumetric integration is a step that instead of fusing point clouds or creating a mesh, a volumetric surface representation is created.
5. Given the global pose of the camera, oriented points are converted into global coordinates, and a single 3D voxel (volume cell) grid is updated;
6. Finally, the volume is raycasted to extract views of the implicit surface, for rendering to the user.

When using the global pose of the camera, this raycasted view of the volume also equates to a synthetic depth map, which can be used as a less noisy and more globally

consistent reference frame for the next iteration of ICP. This allows tracking by aligning the current live depth map with our less noisy raycasted view of the model, as opposed to using only the live depth maps frame-to-frame.

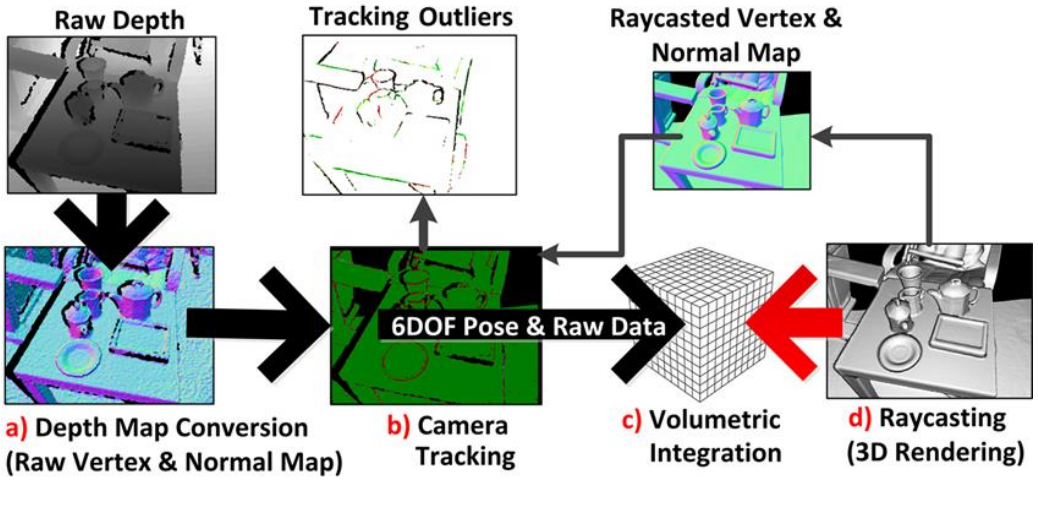


Figure 2 – Kinect fusion algorithm

Each of these steps is executed in parallel on the GPU using the CUDA language. The advantages of this system are: a novel GPU pipeline that achieves 3D tracking, reconstruction, segmentation, rendering, and interaction, all in real-time using only commodity camera and graphics hardware.

2.4. Facial expressions detection

The Facial Action Coding System (FACS), developed by Paul Ekman and Wallice Friesden [40], is a system for describing all visually distinguishable facial movements, based on a selection of facial landmarks. Landmarks are markers placed on a 3D surface that can be used to analyze the deformation [41] of a 3D model, to create and extract a facial expression. Landmarks should be placed at positions with high mobility in the facial expression change, such as areas around the eyes, eyebrows and mouth. Some authors [42] describe an extension to FACS called FACS++, which addresses the lack of emotional spatiotemporal information by using vision-based observations techniques with the dynamics of facial expressions.

This section introduces a brief but relevant resume of the main developments made both in regular and micro facial expressions detection.

2.4.1. Regular expressions

There are authors [43][44] that have based their facial expression analysis in face and features tracking, applying a more subtle following on head, eyebrows, lips and gaze tracking. They use a hierarchical tracking approach, which means that the appearance based trackers combined can achieve a better performance in evaluating some facial expressions.

Finally, to classify the expressions, there are several techniques, such as k-NN or Naive Bayes [67], which produce numeric classification scores regarding predefined goal clusters. However, these techniques don’t correlate with confidence estimations and once the training process is finished, it is difficult to update the knowledge skills with the expressions of new subjects. One alternative is based on [47][48] Case Based Reasoning (CBR).

CBR provides a confidence prediction based on the numerical score of a k-NN classifier and the historical information of previously solved problems. Moreover, the system is able to re-classify false positives and attach confidence predictions to positive classifications.

Some recent state-of-the-art algorithms and methods are related to deep learning techniques to classify facial patterns and/or expressions.

These authors [53] present an implementation of a facial recognition system using the Microsoft Kinect v2 sensor for the purpose of patient identification in a radiotherapy setting. The system was developed with the Microsoft Kinect v2 using a facial mapping library distributed with the Kinect v2 SDK as a basis for the algorithm. The system extracts 31 fiducial points representing various facial landmarks which are used in both the creation of a reference data set and subsequent evaluations of real-time sensor data in the matching algorithm. To test the algorithm, a database of 39 faces was created, each with 465 vectors derived from the fiducial points, and a one-to-one matching procedure was performed to obtain sensitivity and specificity data of the facial identification system. Receiver operating characteristic (ROC) curves were plotted to display system performance and identify thresholds for match determination. The results indicate a fairly robust methodology for verifying, in real-time, a specific face through comparison from a pre-collected reference data set. In its current implementation, the process of data collection for each face and subsequent matching session averaged approximately 30 seconds, which may be too onerous to provide a realistic supplement to patient identification in a clinical setting.

In summary, a facial recognition system can be implemented for patient identification using the Microsoft Kinect v2 sensor and the distributed SDK. In its present form, the system is accurate-if time consuming-and further iterations of the method could provide a robust, easy to implement, and cost-effective supplement to traditional patient identification methods.

The facial expression is an important part in human interaction, along with other forms of nonverbal communication, such as postures and gestures. The human facial expression conveys a remarkable amount of information that can reflect emotional

feelings. Perceiving people's facial expressions can help a person comprehend their emotions.

The new technology provided today for capturing the facial expressions, with rapid, high resolution image acquisition, helps us to analyze and recognize in real time the facial emotions. This can be useful in many real time applications, like entertainment, airport security, trading (the customer's feeling about a product) and patient monitoring.

In this research [54], the authors focus on the emotion recognition from facial expressions by using Microsoft Kinect for Windows sensor V2 and the face tracking SDK to recognize eight expressions. This approach obtained a very good result and the identification rate was about 96%. They have used a simple Neural Network which made the system simple and efficient.

Some authors use facial expressions because they are good examples of nonverbal cues used in inter-human interaction. This research [55] presents a facial expression recognition approach using deep learning. The approach is based on the analysis of subtle changes in facial features of human face. The detected facial features, designed as action units, are mapped to two psychological measurements, arousal and valence, using support vector regression. This approach uses deep learning to process and analyze each action unit. For each of these actions units and combinations, a two-class classifier has been trained. The facial expression is recognized by finding the minimum Euclidean distance from the detected point (arousal and valence) and the basic emotions. The proposed approach has shown a recognition rate of more than 90%.

3D face was recently investigated for various applications, including biometrics recognition and diagnosis support. Describing how a face bends and which kinds of patches it is composed by, is the aim of studies in Face Analysis, which ultimate goal is to identify which features could be extracted from three-dimensional faces depending on the objective of its end application.

In this research [56], the authors' proposed 54 novel geometrical descriptors for Face Analysis. They are generated by composing primary geometrical descriptors such as mean, Gaussian, principal curvatures, distance, shape index, curvedness, and the

coefficients of the fundamental forms. The new descriptors were mapped on 217 facial depth maps and analyzed in terms of descriptiveness of facial shape and exploitability for localizing landmark points. Automatic landmark extraction stands as the final aim of this analysis. Results showed that the newly generated descriptors are suitable to 3D face description and to support landmark localization procedures.

Other research points out that higher level features can represent the abstract semantics of original data. In [57] a multiple scales combined deep learning network to learn a set of high-level feature representations through each stage of convolutional neural network for face recognition is proposed. There are two main differences between their model and the traditional deep learning network. The first, is the way they get the prefixed filter kernels by learning the principal component of images' patches using Principal Component Analysis (PCA) [64] , nonlinearly process the convolutional results by using simple binary hashing, and pool them using spatial pyramid pooling method. The second, in their model, the output features of several stages are fed to the classifier.

The purpose of combining feature representations from multiple stages is to provide multiscale features to the classifier, since the features in the latter stage are more global and invariant than those in the early stage. Therefore, the system encodes both holistic abstract information and local specific information. With extensive experimental results the system shows that can efficiently extract high-level feature presentations and outperform state-of-the-art face/expression recognition methods on multiple modalities benchmark face-related datasets.

Over the last few years, deep learning has produced breakthrough results in many application fields including speech recognition, image understanding and so on. These authors [58] try to apply deep learning techniques for real-time facial expression recognition instead of hand-crafted feature-based methods. The proposed system can recognize human emotions based on facial expressions using a webcam. It can detect faces and recognize users with a distance of 2∼3m for Television set environments. It can determine whether a user is expressing happiness, sadness, surprise, anger, disgust, neutral or any combination of those six emotions. The experimental results show that

the proposed method achieves high accuracy. It can be used for various services such as consumer behavior research, usability studies, psychology, educational research and market research.

2.4.2. Micro expressions

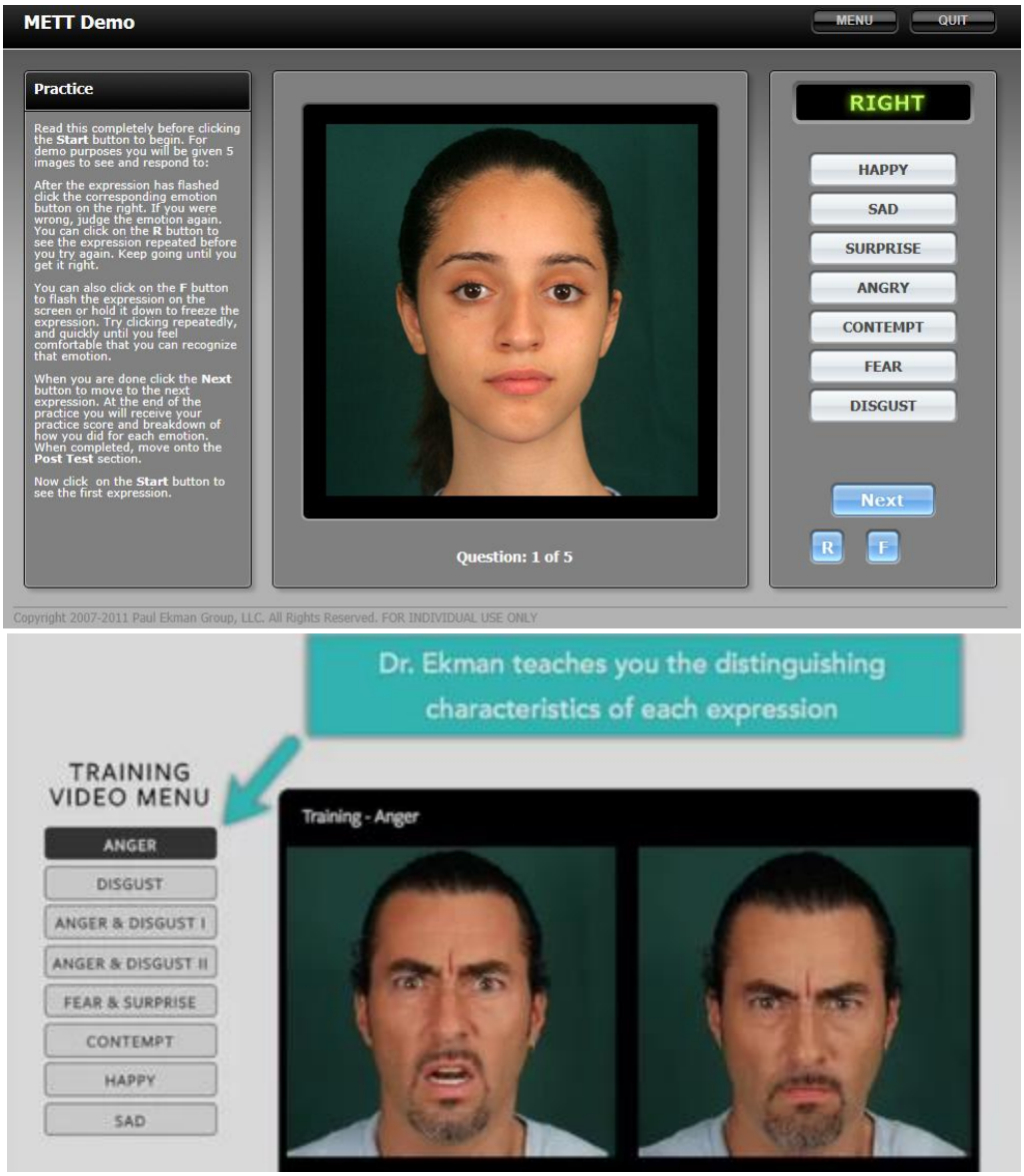
Micro expressions are a brief, involuntary facial expression shown on the face of humans according to emotions experienced. Unlike regular facial expressions, it is difficult to fake micro expressions.

Micro expressions express the seven universal emotions as shown in Figure **3**: disgust, anger, fear, sadness, happiness, surprise, and contempt. They can occur as fast as 1/25 of a second [45].Using the Ekman method built on over 40 years of published research on micro expressions and deception, an application was designed and implemented to train people to improve the abilities to understand the world around us. Some US government agencies, educational and medical professionals are using Dr. Ekman's training to enhance their ability to better evaluate people as we can see in Figure 4.

Microexpressions were first revealed in 1966 by Haggard and Isaacs. These "micromomentary" expressions where outlined by Haggard and Isaacs in their study while "scanning motion picture films of psychotherapy hours, searching for indications of non-verbal communication between therapist and patient".

Source: https://en.wikipedia.org/wiki/Microexpression

Figure 3 – Universal Facial expressions



Source:https://www.paulekman.com/micro-expressions-training-tools/

Figure 4 – Paul Ekman training application for micro expressions

This training program can evaluate the user ability to successfully identify the micro expression that appears in the video image. To better understand the difficulty to evaluate these expressions we advise to experiment the online training application available in [65].

2.5. **Affective Computing**

Affective computing is the development and analysis of systems and devices that can identify, understand, process, and simulate human emotions. The inspiration for the study and research of this area is the ability to simulate empathy. The system should infer the emotional state of humans and adapt its actions to them. The term Affective Computing, was devised by Dr. Rosalind W Picard, who is the Director of Affective Computing Research at the MIT Affective Computing Research Group. In 1997 she published a book named Affective Computing, becoming the common term used for that field of computing.

The way people participate with an activity has been studied from several perspectives in HCI and psychology. The term “engagement” involves attentional and emotional involvement with a task. Engagement is also not stable, but fluctuates throughout an interaction experience.

Detecting emotional information begins with passive sensors, which capture data about the user's physical state or behaviour, without interpreting the input. The data gathered is analogous to the cues that human beings use to perceive emotions in others. For example, a video camera may capture facial expressions, body posture, and gestures, while a microphone may capture speech. Other sensors detect emotional cues by directly measuring physiological data, such as skin temperature and galvanic resistance. [51]

Recognizing emotional information requires the extraction of substantial patterns from the gathered data. This is obtained using machine learning techniques. In [66] a fusion of different modalities was used, such as facial expression detection, natural language processing, or speech recognition, to produce labels (i.e. 'awe') to identify what emotion was being expressed.

The vast majority of present systems are data-dependent. This creates one of the biggest challenges in detecting emotions based on speech, as it implicates choosing an

appropriate database to train the classifier. Most of the currently possessed data was obtained from actors and is thus a representation of archetypal emotions. Those so-called acted databases are usually based on the Basic Emotions theory (by Paul Ekman), which assumes the existence of six basic emotions (anger, fear, disgust, surprise, joy, sadness), the others simply being a mix of the former ones. [52] Nevertheless, these still offer high audio quality and balanced classes, which contribute to high success rates in recognizing emotions.

Some authors [50] explored how computer vision techniques can be used to detect engagement while students complete a task. Students provided engagement annotations both concurrently during the writing activity and retrospectively from videos of their faces after the activity as shown in Figure 5.

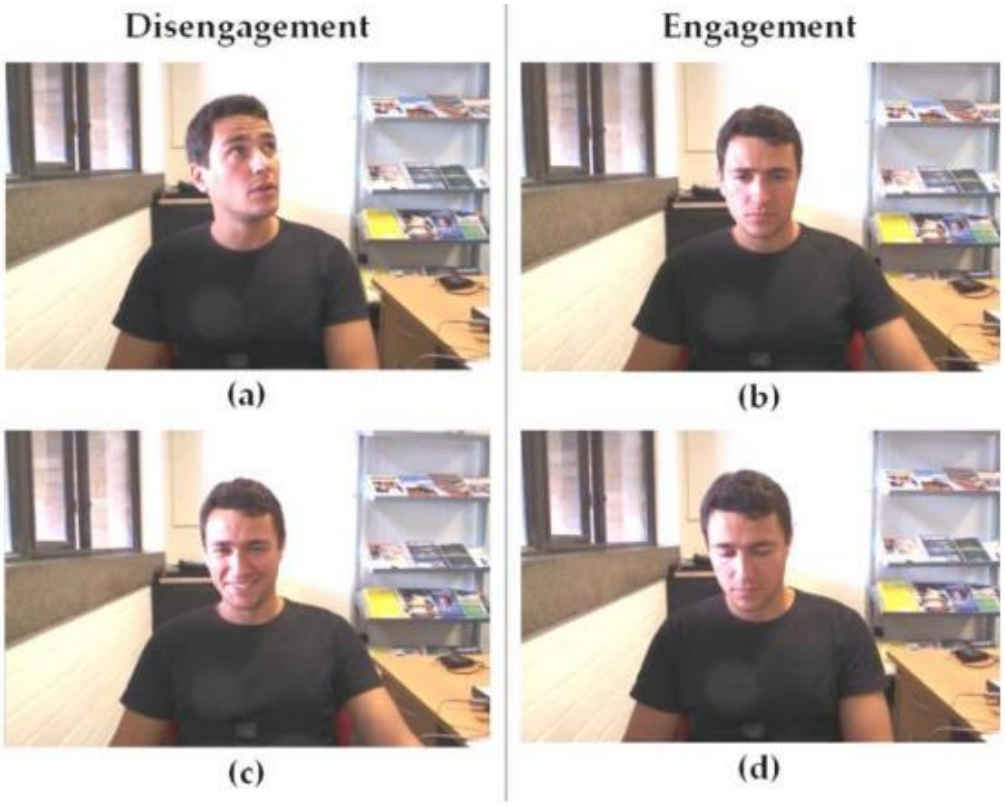


Figure 5 – Detect engagement through computer vision techniques

Source: Monkaresi, Hamed & Bosch, Nigel & Calvo, Rafael & D'Mello, Sidney. (2016). Automated Detection of Engagement Using Video-Based Estimation of Facial Expressions and Heart Rate. IEEE Transactions on Affective Computing. 8. 1-1. 10.1109/TAFFC.2016.2515084

They used computer vision techniques to extract three sets of features from videos, heart rate, Animation Units (from Microsoft Kinect Face Tracker), and local binary patterns in three orthogonal planes (LBP-TOP). These features were used in supervised learning for detection of engagement. The Kinect Face Tracker features produced the best results among the individual channels, but the overall best results were found using a fusion of channels.

Facial recognition is an area in affective computing with applications in e-learning systems [69] as shown in Figure 6, machine vision to robots, detecting angry car drivers, analysing shoppers to get feedback on the product and profiling criminals by facial analysis.

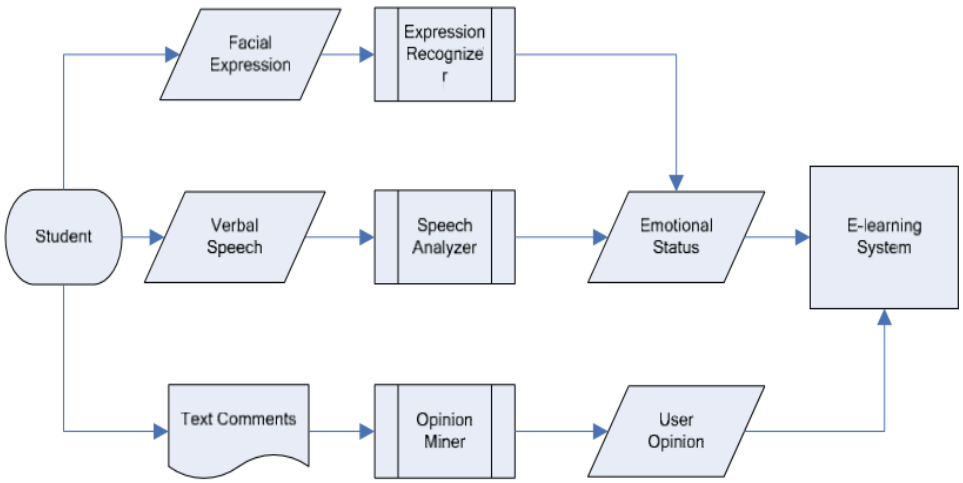


Figure 6 – Application of Affective Computing in e-learning

Source: Hongfei Lin, Fengming Pan, Yuxuan Wang, ShaohuaLv and Shichang Sun (2010). Affective Computing in Elearning, E-learning

2.6. **Conclusions**

This chapter introduced techniques for 3d facial geometry and facial expression analysis. It also described what micro/regular facial expressions are and explained the role of computer vision and human computer interface in this study. We also highlighted and summarized relevant aspects to this research.

These techniques and fields referred here aid in identifying a graphical user interface, application functions requirements and an algorithm model system. The goal is to detect components, techniques and methods that could be improved or aggregate to design a new approach to solve the problem of multimodal data sensor for facial patterns.

In conclusion, based on the results of the research executed to this chapter we might say that both deep learning algorithms using Kinect sensor will be especially interesting to the development of this work, because they introduce a low cost and high efficient possible solution to the facial analysis.

Another aspect will be the assertion that will be very hard to capture micro expressions with depth sensors because of the current frame rate of these sensors.

3 Experimental Results

"Science is beautiful when it makes simple explanations of phenomena or connections between different observations. Examples include the double helix in biology and the fundamental equations of physics." Stephen Hawking

The main goal of this work is to investigate a set of systems and techniques that make it possible to answer the question regarding how to use multimodal sensor data to obtain a classification system in order to identify facial patterns. To accomplish this, a prototype was developed using Microsoft Kinect for Windows sensor V2. This chapter presents the experimental tests and results conducted with the prototype, on various types of algorithms and techniques regarding 3D geometric information in real time in order to help to achieve this goal.

3.1. Facial analysis exploratory research

In our system for facial pattern recognition, we used Microsoft Kinect for Windows sensor V2, which simplifies the facial feature extraction. Kinect for Windows v2 sensor is a product from Microsoft.

It is a device with depth sensing technology, a built-in color camera, an infrared (IR) emitter, and a microphone array, enabling it to sense the location and movements of people. Microsoft provides, with this new Kinect sensor, a development kit (SDK 2.0) with novel facilities, drivers, tools, APIs and device interface. As compared to its predecessor, it offers enhanced color depth, image fidelity, video definition, depth perception, skeletal tracking and improved range of high quality operation (.5 meters near, 4.5 meters far).

The sensor additionally includes full-HD video and wider field of view, improved skeletal tracking and new active infra-reduction for better tracking in low light. The sensor is also

known to connect to a Windows 10 computer via a power supply and computer interface hub that features a USB3.0 port.

The Microsoft Kinect Studio [20], together with the Kinect for Windows Software Development Kit (Kinect for Windows SDK), enables us to create applications that can track human faces in real time. The Microsoft.Kinect.Face API provides capability for tracking facial feature locations, tracking facial animations, and capturing 3D representations of faces in real time.

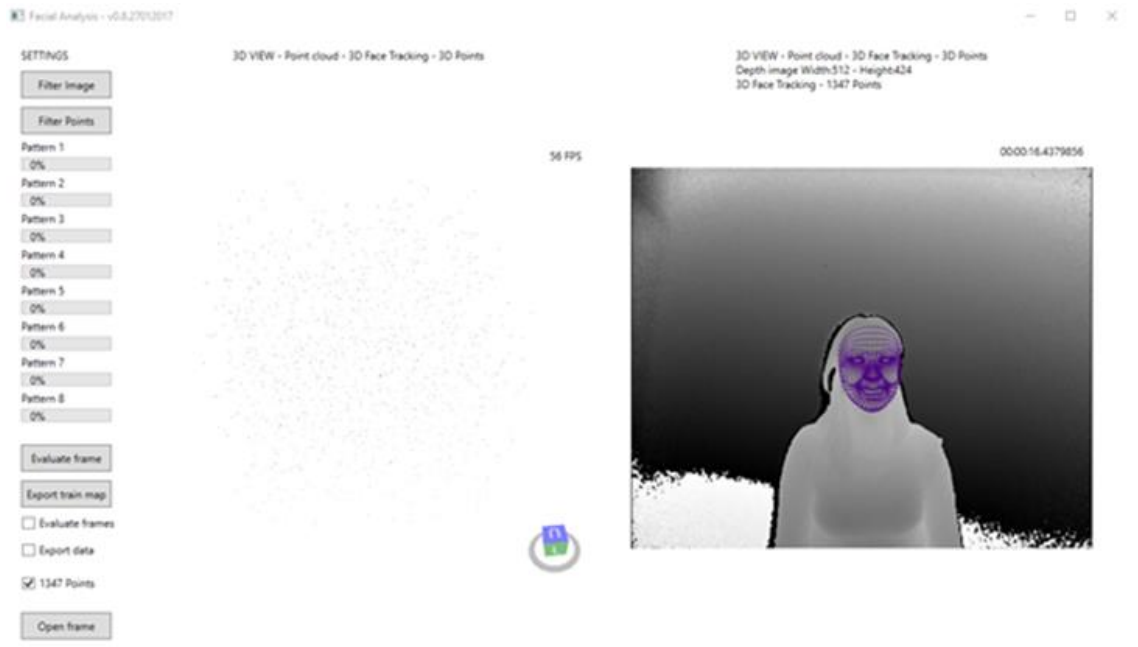


Figure 7 – Prototype system interface

With this technology, we built an application that is able to track up to a max of 1347 3D points of the face that we can map to the 2D image. In addition, the application can evaluate still images and video for facial pattern depending on the input requisites, and also export and filter 3D data. User annotation capability is also possible to improve evaluation systems. Figure 7 and Figure 8 illustrate its interface and the system capturing process. A YouTube channel playlist is also available where can be seen videos of the application running in real time [62].

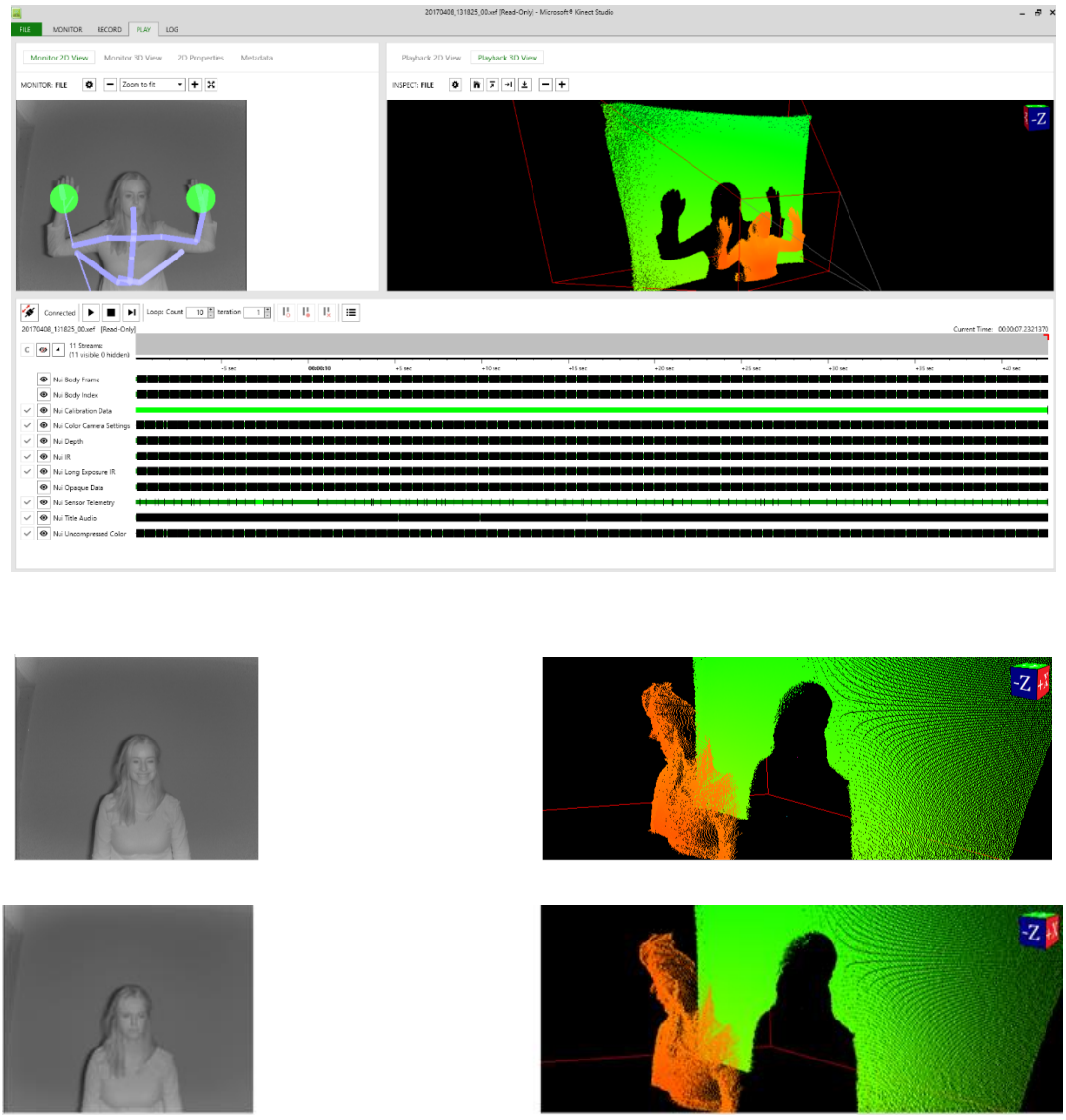


Figure 8 – System capturing process in Kinect studio v2.0

The export data option is able to save to the disk the 3D points being filtered and showed in the image in Figure 9. In Figure 10 and Figure 11 we can see all the 3D points produced by the Kinect sensor. The application has also a neural network that evaluates facial patterns in real time with the computed values from the Euclidean distance. Then exports the values used to evaluate facial patterns enabling the creation of charts. Using a long short-term memory (LSTM) [59, 60] – the deep learning neural network of the application is also able to classify in real time the facial patterns.

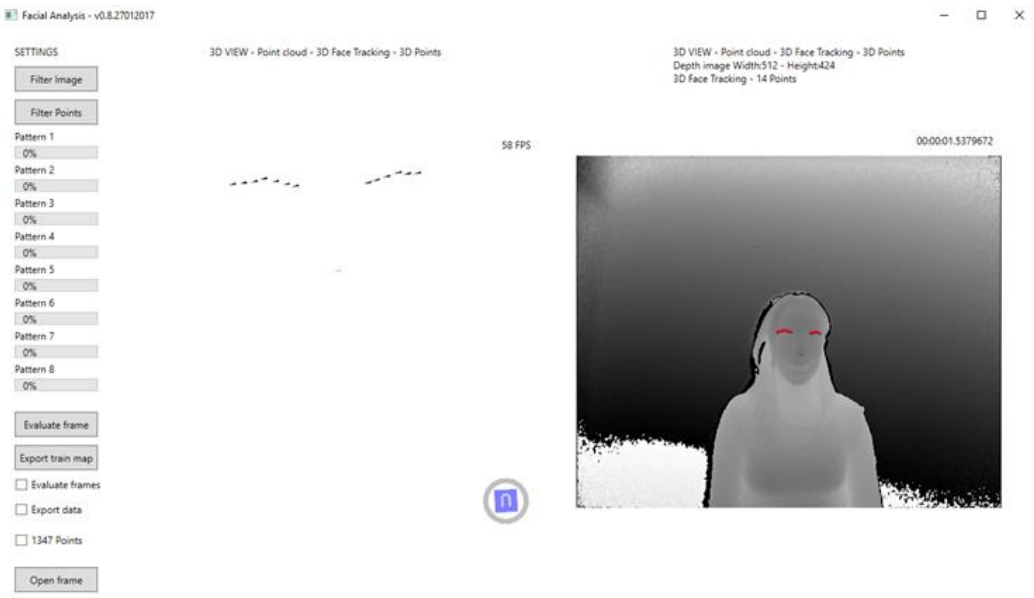


Figure 9 – Prototype 3D point visualization and depth map with 3D eyebrows and nose points mapped into

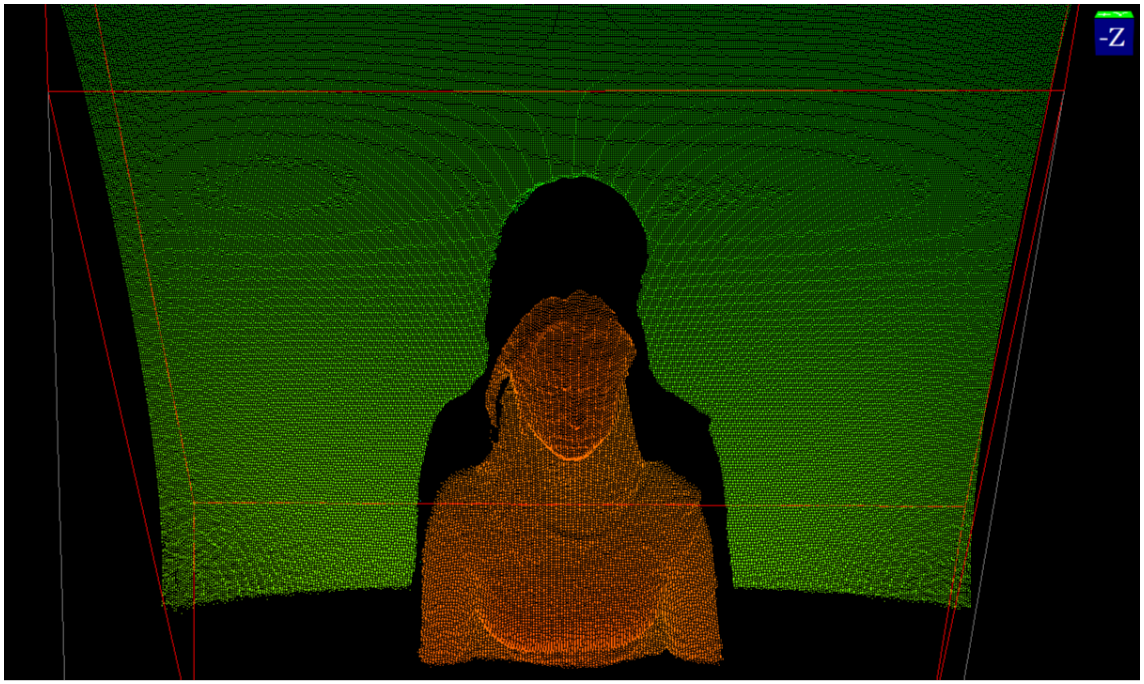


Figure 10 – Front view of 3d cloud point from sensor

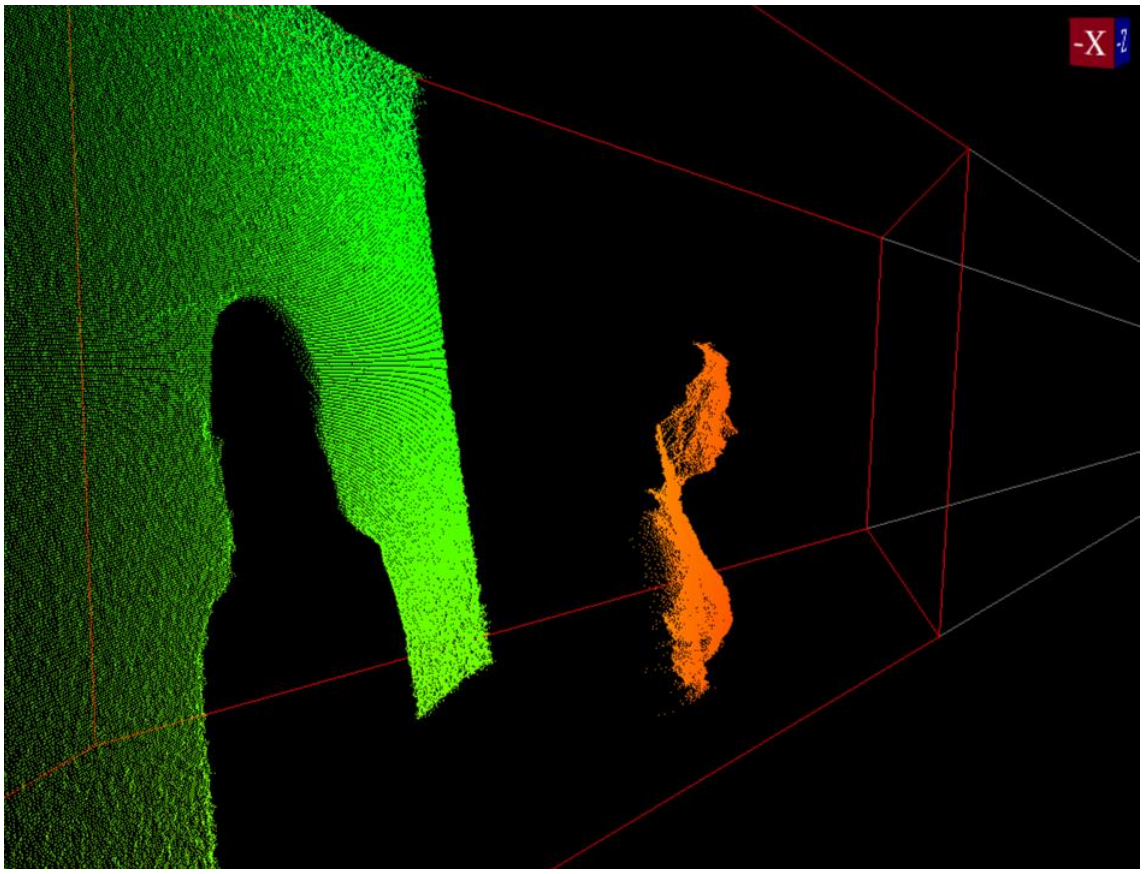


Figure 11 – Side view of 3d cloud point from sensor

3.2. **Proposed method**

The ultimate purpose of our research is to recognize facial patterns using just 3d geometry information. Currently we have made experiments with our system using 30 subjects, 15 male and 15 female ranging from 12 to 73. In order to establish its accuracy in recognizing patterns of different individuals we use in real time, a deep learning neural networks available in the prototype and also data visualization with external tools.

The phases of our method illustrated in the flowchart in Figure 12 is similar to those proposed in the method used in [49] with the innovation regarding the combination of the Euclidean distance and vector direction movement between the 3d tracked points, the visualization process and the LSTM neural network.

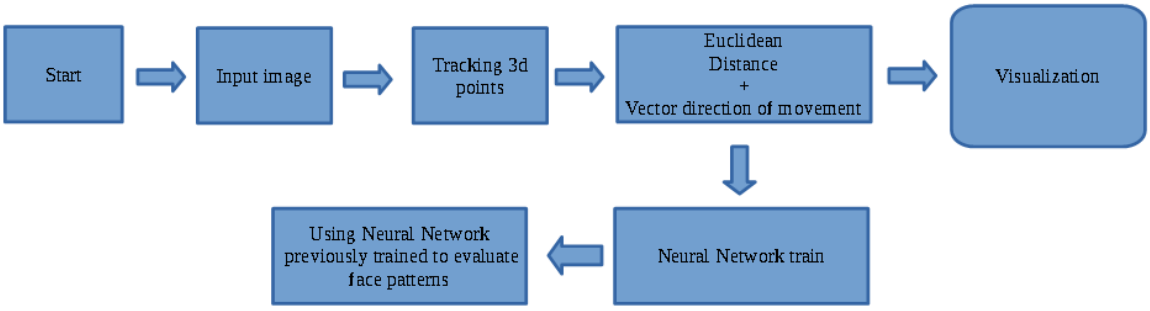


Figure 12 – Flowchart of our proposed method (see for more detail Appendix 5)

We construct our own ground truth data, which is used for quantitatively training the neural network. The ground truth data is based on capturing, with the Kinect v2, a subject making a specific facial pattern during several seconds. The accuracy of the ground truth must be good enough to identify facial features, like the eyes, the nose, the mouth, the eyebrows' and the face edge. To test the accuracy, we use Kinect studio playback capabilities and the prototype to see if the desired points are captured. Our setup can be seen in Figure 13. The distance between the subject and the sensor is on average 60cm, otherwise the quality in capturing data may be compromised. To ensure

that there is no incorrect record session, we use a distance of at least 45cm between the subject and the Kinect sensor and two initial protocols:

a) Moving the arms to a surrender position and then backing down, and;

b) Walking to the setup and seat in the chair.

The required validation can easily be asserted by using the Kinect studio record/play capabilities.

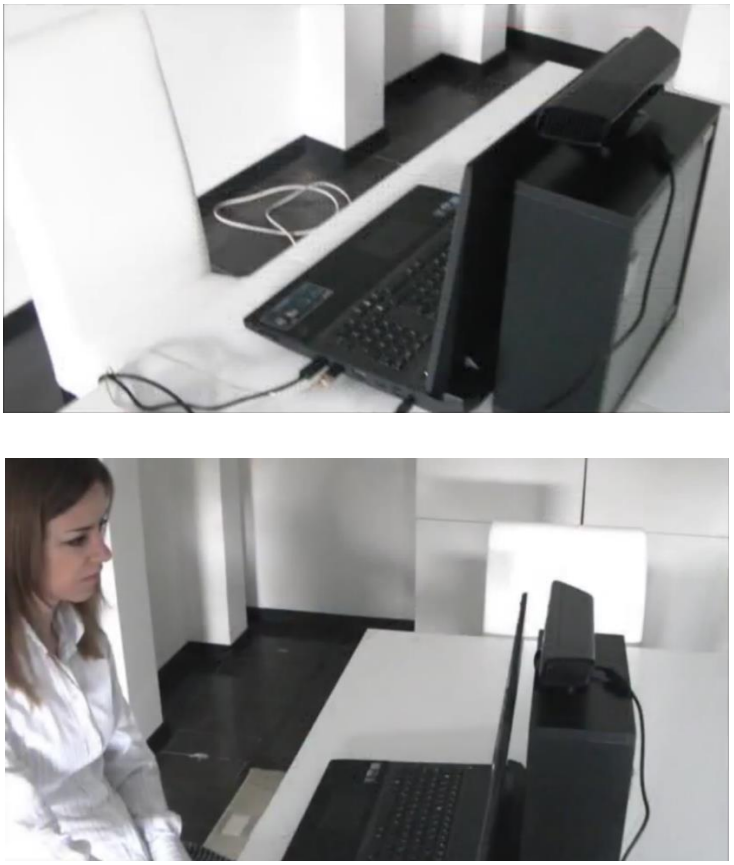


Figure 13 – Setup scenario with the Kinect sensor

We have used the setup above to collect two datasets with ground truth facial points, shown in Figure 7 and Figure 8. The preliminary dataset has 6 different subjects, 4 facial patterns, 50 seconds of record data in each subject with 11/10 streams (Nui Body Frame, Nui Body Index, Nui Depth, Nui IR, and Nui Uncompressed Color). It is used for parameter tuning and to test the Kinect v2 in Figure 14, under the minimal distance possible between the subject and the sensor, i.e. 45 cm.

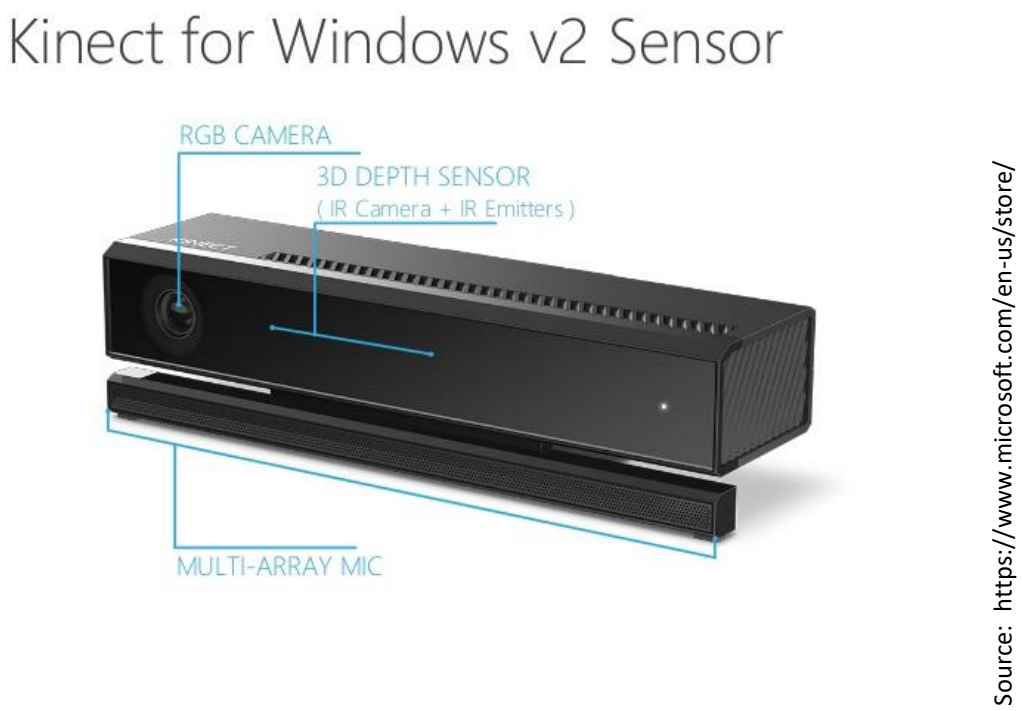


Figure 14 – Kinect v2 sensor

The main dataset has 30 different subjects and is used to evaluate 8 facial patterns as seen in Figure 15 without imposing the 45cm distance mark. The reason for not enforcing the 45 cm is because the captures are made in different locations. For each dataset, we have an average of 6GB and 10GB per file size respectively.

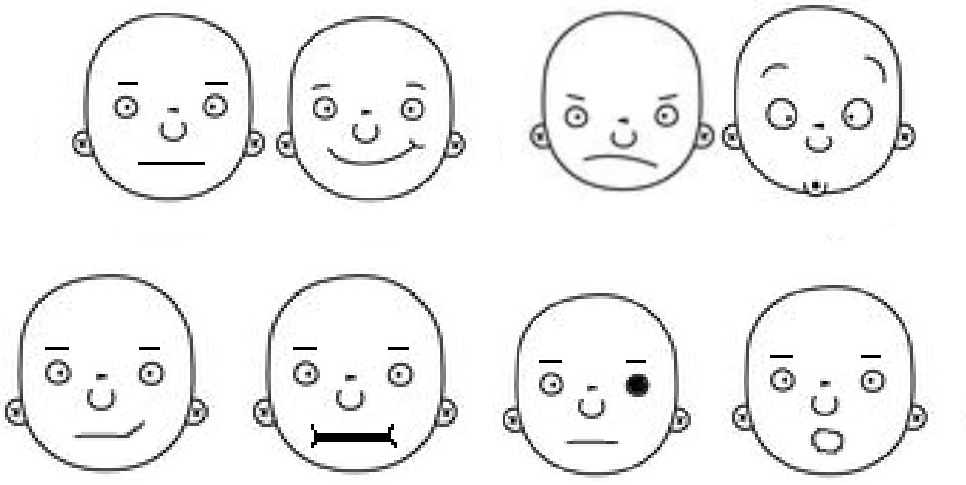


Figure 15 – Facial patterns that users had to mimic (see for more detail Appendix 1 and 2)

In result, each subject is asked to perform 8 different tasks. Each task is actually a distinct facial expression: neutral, smile, angry, surprised, mouth to the side, lip bite, close eye and open mouth.

These two datasets give the possibility, through Kinect studio v2.0, to be played and injected to the application. Further, the application generates a text file with the calculated Euclidean distances between each face control point and the reference point, besides the direction of the 3D movement between them. The Euclidean distance is used to train the neural network.

The reference point can change according the area of the face that is being targeted to be captured. Presently, we consider the eyebrows or the mouth area. If the eyebrows area is considered, the reference point is located at the upper middle part of the nose. If the mouth area is considered, then, the reference point is positioned at the upper center lip.

A recurrent neural network (RNN) [59, 60] is a class of artificial neural network where connections between units form a directed cycle. This allows it to exhibit dynamic temporal behavior. RNNs can use their internal memory to process arbitrary sequences of inputs, making them good option for handwriting or speech recognition. In this case a specific RNN is used, a LSTM.

This kind of neural network is a deep learning system that avoids the vanishing gradient problem. LSTM is normally augmented by recurrent gates called "forget" gates. LSTM prevents back propagated errors from vanishing or exploding. Instead, errors can flow backwards through unlimited numbers of virtual layers unfolded in space. That is, LSTM can learn tasks that require memories of events that happened thousands or even millions of discrete time steps earlier. Problem-specific LSTM-like topologies can be evolved. LSTM works even given long delays between significant events and can handle signals that mix low and high frequency components.

The platform for deep learning used is the Microsoft Cognitive Toolkit (CNTK) [63]. It is an open-source toolkit for commercial-grade distributed deep learning. It describes neural networks as a series of computational steps via a directed graph. CNTK allows the

user to easily realize and combine popular model types such as feed-forward deep neural network (DNNs), convolutional neural networks (CNNs) and recurrent neural networks (RNNs/LSTMs). CNTK implements stochastic gradient descent (SGD, error backpropagation) learning with automatic differentiation and parallelization across multiple GPUs and servers. Figure 16 illustrates CNTK architecture.

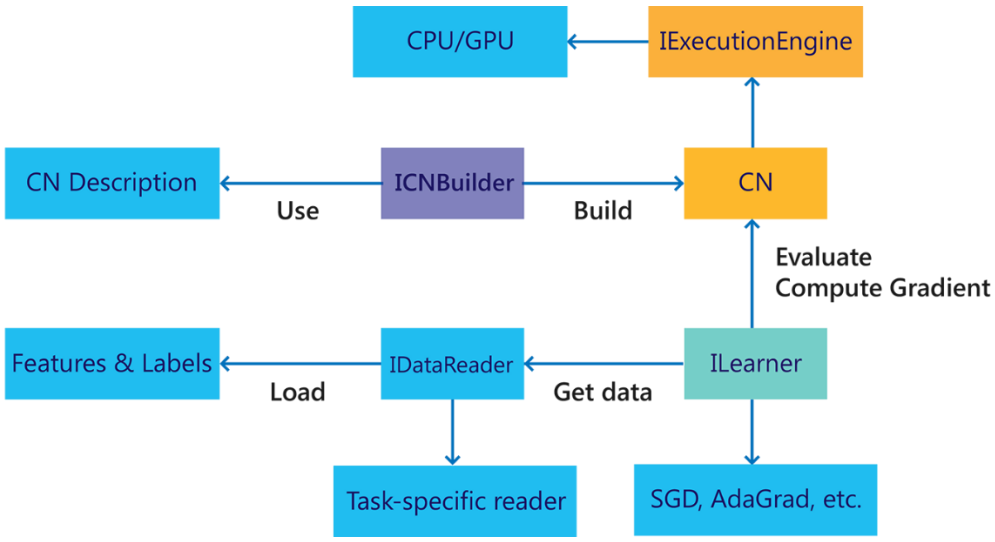


Figure 16 – CNTK architecture

The task of the neural network is to look at the query (column S0) and predict the intent of the sequence (column S1) as seen in table 3. The column S2 predicts a value individually for each query.

The model we use is a recurrent model consisting of an embedding layer, a recurrent LSTM cell, and a dense layer to compute the posterior probabilities as seen in Figure 17.

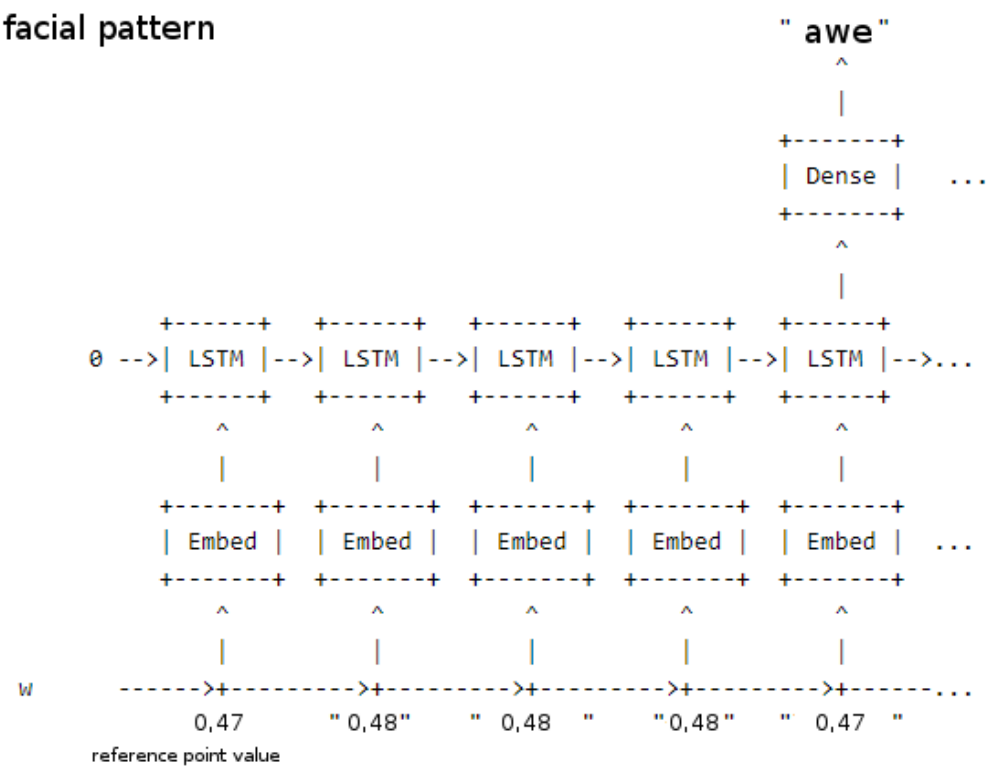


Figure 17 – LSTM model from CNTK used to train data [68]

The training data has the following line format:

<Field1>; |SO <Field2>;|#<Field3>;|S1 <Field4>;|#<Field5>;|S2 <Field6>;|#<Field7>

- Field1 = id;
- Field2 = numeric value that is a representation of the sum of the Euclidean distance between the face control points and the reference point or the direction of the 3D movement of each control point;
- Field3 = comment with the actual value of the evaluated Euclidean distance or the 3D direction;
- Field 4 = general class for sequence;
- Field 5 = comment with class name;
- Field 6 = class value;
- Field 7 = comment with subject id and frame number.

Table 3 illustrates a file containing several records according the format described above. This data is used to train the LSTM network for the CNTK engine.

0	|S0	7:1	|# 0.47	|S1	0:1	|# awe	|S2	0:1	|# frame 1 - subject002
0	|S0	8:1	|# 0.48		|S2	0:1	|#	frame 2	
0	|S0	8:1	|# 0.48		|S2	0:1	|#	frame 3	
0	|S0	8:1	|# 0.48		|S2	0:1	|#	frame 4	
0	|S0	7:1	|# 0.47		|S2	0:1	|#	frame 5	
0	|S0	8:1	|# 0.48		|S2	0:1	|#	frame 6	
0	|S0	7:1	|# 0.47		|S2	0:1	|#	frame 7	
0	|S0	7:1	|# 0.47		|S2	0:1	|#	frame 8	
0	|S0	8:1	|# 0.48		|S2	0:1	|#	frame 9	
0	|S0	8:1	|# 0.48		|S2	0:1	|#	frame 10	
0	|S0	7:1	|# 0.47		|S2	0:1	|#	frame 11	
0	|S0	7:1	|# 0.47		|S2	0:1	|#	frame 12	
0	|S0	7:1	|# 0.47		|S2	0:1	|#	frame 13	
0	|S0	7:1	|# 0.47		|S2	0:1	|#	frame 14	
0	|S0	7:1	|# 0.47		|S2	0:1	|#	frame 15	

Table 3 – Sequence with 15 frames - training data for CNTK engine LSTM network

Each sequence is composed by 15 frames values captured from Kinect. For each frame the sum of the Euclidean distance between each control point and the reference one is calculated. An alternative to the Euclidean distance is to use the 3d vector representing the control point displacement in the space.

These sequences are feed to the training process. After training, we use the model created to evaluate in real time the sequences. The total number of frames in each sequence can be altered in run time to train or evaluate another sequences. This means that we can have a model created by a sequence of 15 frames and then, evaluate a sequence of 30 or 20 frames with this model as seen in Figure 18. Also the number of points can be changed as we can see in Figure 18.

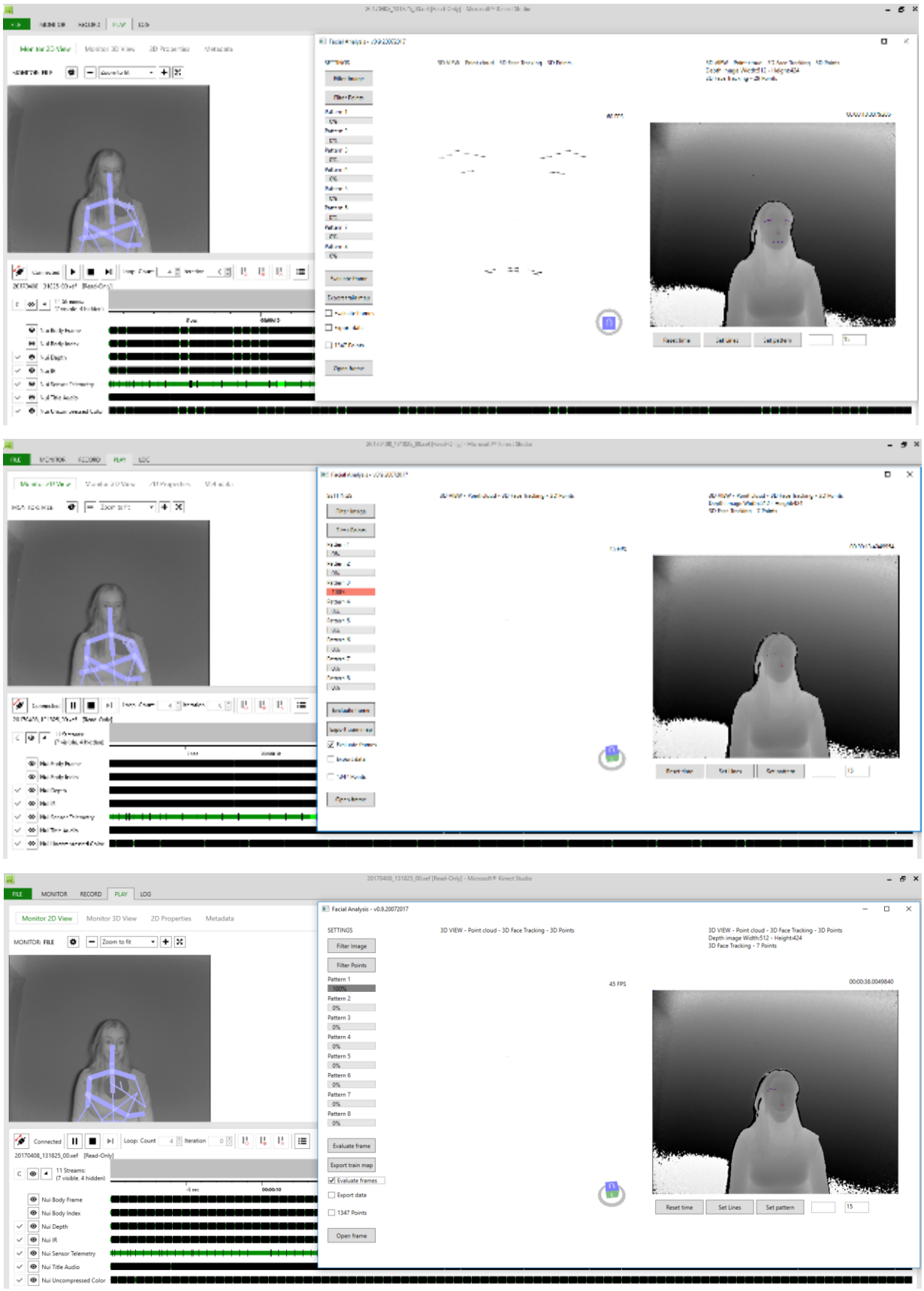
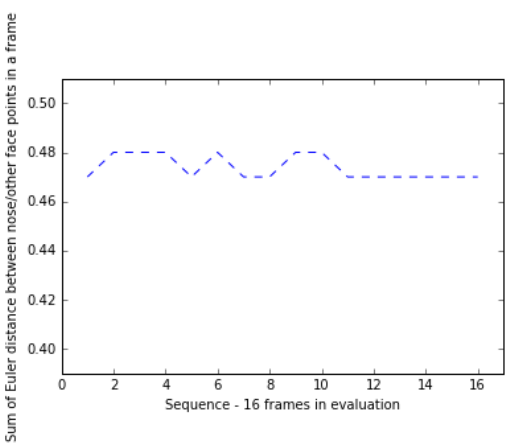
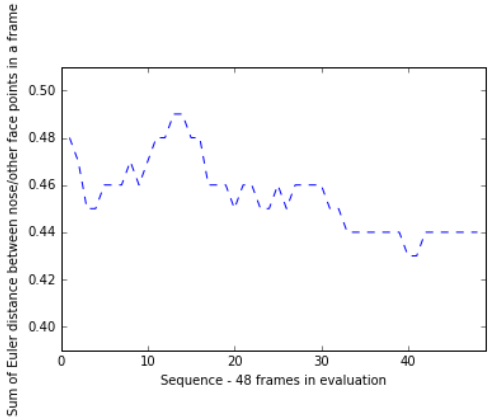


Figure 18 – 3 screenshots of the application running with different configurations and evaluating sequences

As a test case and for fast prototype an evaluation script was developed in CNTK engine. This platform allowed evaluating the change of frames in a sequence as we can see in Figure 19.




```
[('0.48', 'awe'),
 ('0.47', 'awe'),
 ('0.48', 'awe'),
 ('0.48', 'awe'),
 ('0.48', 'awe'),
 ('0.48', 'awe'),
 ('0.48', 'awe'),
 ('0.48', 'awe'),
 ('0.47', 'neutral'),
 ('0.47', 'neutral'),
 ('0.47', 'neutral'),
 ('0.47', 'neutral'),
 ('0.47', 'neutral'),
 ('0.47', 'neutral'),
 ('0.47', 'neutral'),
 ('0.47', 'neutral')]
```




```
[('0.48', 'neutral'),  ('0.46', 'neutral'),
 ('0.47', 'neutral'),  ('0.45', 'neutral'),
 ('0.45', 'neutral'),  ('0.46', 'neutral'),
 ('0.45', 'neutral'),  ('0.46', 'neutral'),
 ('0.46', 'neutral'),  ('0.46', 'neutral'),
 ('0.46', 'neutral'),  ('0.46', 'neutral'),
 ('0.46', 'neutral'),  ('0.45', 'neutral'),
 ('0.47', 'neutral'),  ('0.45', 'neutral'),
 ('0.46', 'neutral'),  ('0.44', 'neutral'),
 ('0.47', 'neutral'),  ('0.44', 'neutral'),
 ('0.48', 'neutral'),  ('0.44', 'neutral'),
 ('0.48', 'neutral'),  ('0.44', 'neutral'),
 ('0.49', 'neutral'),  ('0.44', 'angry'),
 ('0.49', 'neutral'),  ('0.44', 'angry'),
 ('0.48', 'neutral'),  ('0.44', 'angry'),
 ('0.48', 'neutral'),  ('0.43', 'angry'),
 ('0.46', 'neutral'),  ('0.43', 'angry'),
 ('0.46', 'neutral'),  ('0.44', 'angry'),
 ('0.46', 'neutral'),  ('0.44', 'angry'),
 ('0.45', 'neutral'),  ('0.44', 'angry'),
 ('0.46', 'neutral'),  ('0.44', 'angry'),
 ('0.46', 'neutral'),  ('0.44', 'angry'),
 ('0.45', 'neutral'),  ('0.44', 'angry'),
 ('0.45', 'neutral'),  ('0.44', 'angry')]
```

Figure 19 – Line charts showing different numbers of sequences being evaluated

3.3. Data Analysis and tests

To have a rough and straightforward validation of the proposed method, we considered initially only 4 facial patterns, before taking into account all the 8 ones. So, our data analysis can be divided into two distinct phases.

The first one was a preliminary evaluation of the potential of the proposed method. This phase was actually used to decide if a second phase should be performed, or not (depending on the positivity of the results). The second phase was an extension of the first one, now taking into account, all the 8 facial expressions and a number of subjects much higher than in the first phase.

3.3.1 First phase

In the first phase, six subjects were invited to participate in our recordings, considering only 4 facial expressions. We asked each one of them to express in their faces the following patterns: neutral, smile, angriness and surprise. The subjects were 2 male and 4 female, with age ranging from 27 to 50.

Each subject repeated three times each expression individually and the result was recorded in different files. The recordings were done at a 30 frame rate per second. The subjects had to clearly initiate and terminate each facial expression, starting always from a neutral one and ending at the peak of the intended expression. Only the neutral expression should not cause any significant change.

The Kinect's output can give several facial control points. In this phase, we considered only the control points that map the subject's eyebrows and the top of the nose (in the middle of the two eyebrows). We assumed as starting point to consider only these control points to significantly and potentially infer some facial patterns of any individual.

While the eyebrows normally move when the expression occur, the point in the top of the nose stays steady. This is relevant to us, because it is used as a reference point to the others. A total of six control points per eyebrow plus one reference point in the top of the nose were tracked between frames.

Although each facial expression recordings took more than one second, we wanted to work with only the first fifteen captured frames. We assumed that one second was enough to the subject fully complete a facial pattern. So, the files were filtered, and the first fifteen frames were extracted.

To evaluate the behavior of the 3D control points and attempt to find out patterns, we assumed that the change of their spatial position should reveal something. Because human face has different shapes (for instance, the distance between the eyes or the size of the face itself) and it was almost impossible to guarantee that every subject has the same facial proportions or do not move the head while recording, we optioned to work with relative values.

Euclidean distance is a measure of the true straight line distance between two points in Euclidean space. If we take the reference control point (r_x, r_y, r_z) and an eyebrow point (e_x, e_y, e_z), we have the Euclidean distance for each point given by:

$$d = \sqrt{(r_x - e_x)^2 + (r_y - e_y)^2 + (r_z - e_z)^2}$$

Equation 1 - Euclidean distance (see for more detail Appendix 3)

We used this formula to find out the distance between each eyebrow control point and the reference one. Because of the symmetry of the face, only one of the eyebrows was actually considered. After calculating the distance between each pair of points, the sum per frame was also computed.

Besides using the Euclidean distance, we wanted also to evaluate the direction of the movement of each eyebrow control point. In analytic geometry, the direction cosines (or directional cosines) of a vector are the cosines of the angles (α, β and γ) between the vector and the three coordinate axes. Equivalently, they are the contributions of each component of the basis to a unit vector in that direction. It can be used to evaluate the

direction of the 3D movement of each control point in relation to the reference point. It is calculated using the following formulas:

$$\alpha = cos^{-1}\left(\frac{(e_x - r_x)}{\sqrt{(r_x - e_x)^2 + (r_y - e_y)^2 + (r_z - e_z)^2}}\right)$$

$$\beta = cos^{-1}\left(\frac{(e_y - r_y)}{\sqrt{(r_x - e_x)^2 + (r_y - e_y)^2 + (r_z - e_z)^2}}\right)$$

$$\gamma = cos^{-1}\left(\frac{(e_z - r_z)}{\sqrt{(r_x - e_x)^2 + (r_y - e_y)^2 + (r_z - e_z)^2}}\right)$$

Equation 2 - direction of the 3D movement of each control point (see for more detail Appendix 4)

We calculated the sum of the angles between each control point and the reference one per frame. After calculating it for each pair of points, they were summed up by frame.

Having all these calculation executed for each expression file of each subject, we started plotting Sparkline charts in order to quickly compare and detect any possible existing pattern between subjects' facial expressions. We assumed that the data visualization would help us in this task.

The first sequence of charts considered how the sum of the 3D vector direction angles per each eyebrow control point per all the fifteen frames changed. The second sequence took into account how the sum of the Euclidean distance per each control point and per all the fifteen frames vary along the facial expressions.

Although the values came from different subjects' captured sequences, the resulting charts were very similar in all cases. Figure 20 and Figure 21 show this respectively, illustrating just one of the sequences.

The charts used always the data from the same captured sequence of each subject, so the first sequence of the subject one was compared against the first sequence of the others, while the second, against the second, and so forth. Because of the extreme similarity between the results, we show only the result of one of the captured sequences. Each Sparkline has six values in the series (one for each subject).

	3D Vector Direction Totals (Eyebrown)					
	Subjects					
	1	2	3	4	5	6
Awe						
Neutral						
Angry						
Smile						

Figure 20 – Sum of the 3D vector direction angles per all the 15 frames.

	Euclidean Distance Totals (Eyebrown)					
	Subjects					
	1	2	3	4	5	6
Awe						
Neutral						
Angry						
Smile						

Figure 21 – Sparkline charts of the sum of the Euclidean distance per all the 15 frames.

We also considered how the sum of the 3D vector direction angles per all eyebrow control points per each of the fifteen frames changed, besides how the sum of the Euclidean distance per all the control points and per each of the fifteen frames vary along the facial patterns. Figure 22 illustrates the different results achieved according subject, captured sequence and facial expression in the case of the Euclidean distance and the 3D vector direction. Each Sparkline chart has 15 values (one for each frame in the sequence).

	Euclidean Distance				3D Vector Direction			
Subjects	Awe	Neutral	Angry	Smile	Awe	Neutral	Angry	Smile
1								
2								
3								
4								
5								
6								

	Euclidean Distance				3D Vector Direction			
Subjects	Awe	Neutral	Angry	Smile	Awe	Neutral	Angry	Smile
1								
2								
3								
4								
5								
6								

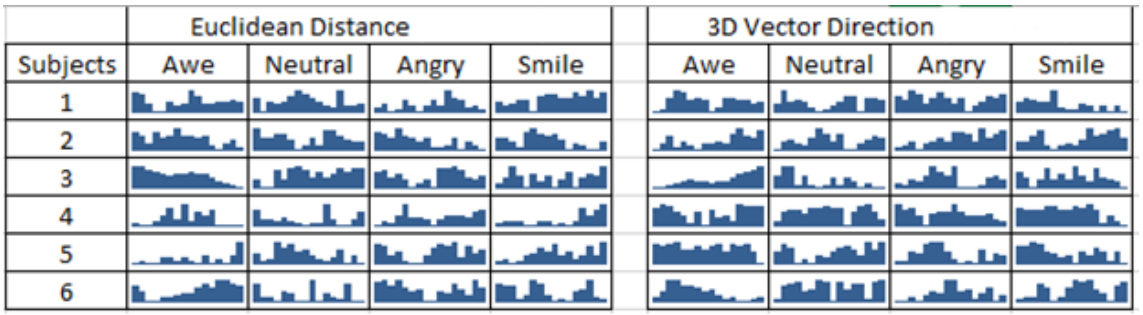
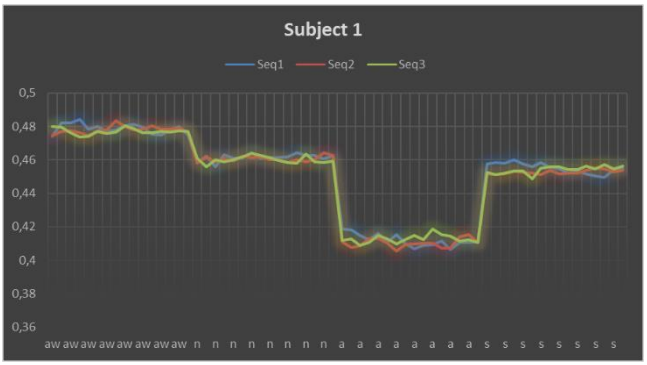


Figure 22 – The Euclidean distance and the 3D vector direction per frame and per subject in captured sequences 1, 2 and 3

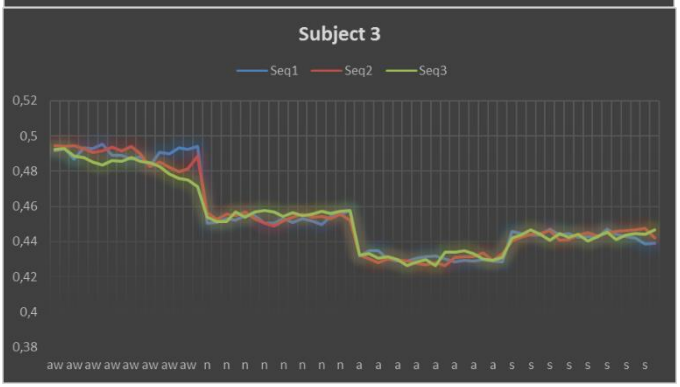
Finally, we plotted line charts for each subject. These line charts were built concatenating the results used in the previous Sparkline charts. In other words, for each captured sequence we took each 15 values of each facial expression and concatenate them, resulting in a total of 15 x 4 values (15 frames and 4 facial expressions).

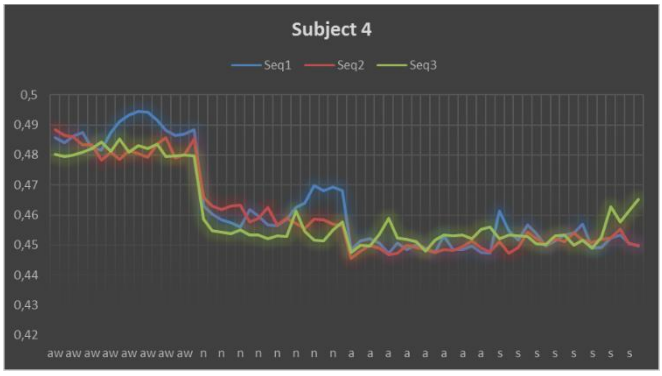
We wanted to know if there were significant changes of the expression behavior for each subject according the captured sequences. We also wanted to find out any similarities between the way these changes occur between subjects and between facial patterns.

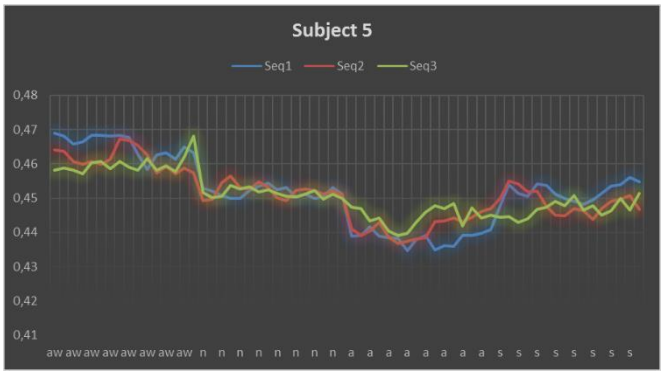
The following charts in Figure 23 illustrate the results. Each sequence is represented with a different color. Besides this, we used abbreviations to represent the expression behind the data: aw = awe, a = angry, n = neutral and s = smile. We considered only the results from the Euclidean distance.




Subject 2
Seq1
Seq2
Seq3
0,52
0,5
0,48
0,46
0,44
0,42
0,4
0,38
aw aw aw aw aw aw aw aw aw n n n n n n n n a a a a a a a a s s s s s s s s


Subject 3
Seq1
Seq2
Seq3
0,52
0,5
0,48
0,46
0,44
0,42
0,4
0,38
aw aw aw aw aw aw aw aw aw n n n n n n n n a a a a a a a a s s s s s s s s


Subject 4
Seq1
Seq2
Seq3
0,5
0,49
0,48
0,47
0,46
0,45
0,44
0,43
0,42
aw aw aw aw aw aw aw aw aw n n n n n n n n a a a a a a a a s s s s s s s s


Subject 5
Seq1
Seq2
Seq3
0,48
0,47
0,46
0,45
0,44
0,43
0,42
0,41
aw aw aw aw aw aw aw aw aw n n n n n n n n a a a a a a a a s s s s s s s s

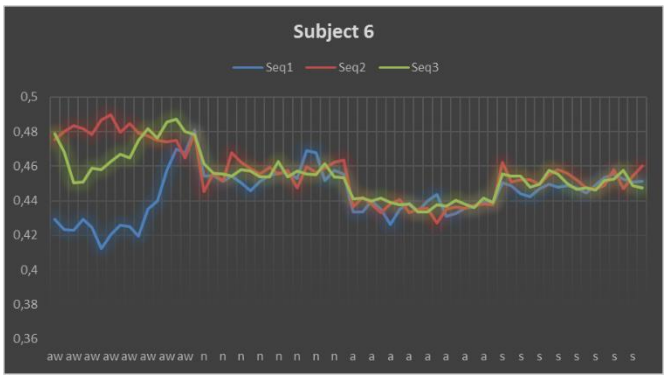


Figure 23 – Line charts per subject concatenating all four patterns

3.3.2 Findings

The results in Figure 20 and Figure 21 showed clearly that different subjects present the same pattern in their facial expression when we consider total values per frames for each control point. The same does not happened when we considered how the sum of the 3D vector direction angles per all eyebrow control points per each of the fifteen frames changed. Neither when we took into account how the sum of the Euclidean distance per all the control points and per each of the fifteen frames vary along the facial patterns. This can be observed in Figure 22. As we can notice, the same subject with the same facial expression presents very distinct results in these Sparkline charts.

It is interesting to observe that although each subject showed different responses when we summarized its facial expression per each frame (summing up all the results per control points), the same does not happen when we summarized the results per control point per all frames. This means that the facial expressions have the same pattern in all the subjects if we consider the global result per control point, but how it is built frame by frame, it is distinct to each subject.

Another relevant output is that each subject changes significantly the way that its facial expression is built along the 15 frames. It is not only distinct, but also vary changeable. We can see that each captured sequence (1, 2 and 3) in charts illustrated in Figure 22, present different profiles for each face expression and each subject. So, the way a subject constructs a facial expression is always irregular and probably unique, each time, although the final result is the same.

Both Euclidean distance and 3D vector direction showed that evaluating the direction of the movement or the displacement of the eyebrow control points did not result in something different. Actually, they both confirm the same conclusions.

Finally, the individual analysis of the line charts illustrated in Figure 23 indicates that each facial expression implies in a very distinct different range of values. For instance, if

we consider subjects 1, 2 and 3 we can see that the awe expression presents a range of data with the highest values, while the angry one has the lowest. The neutral and smile facial expressions have very proximal values and in a range between the other two. The other 3 subjects do not present a so clear distinction, but it still can be noticed. This happened particularly because these subjects were using glasses or had not very well defined and detectable eyebrows.

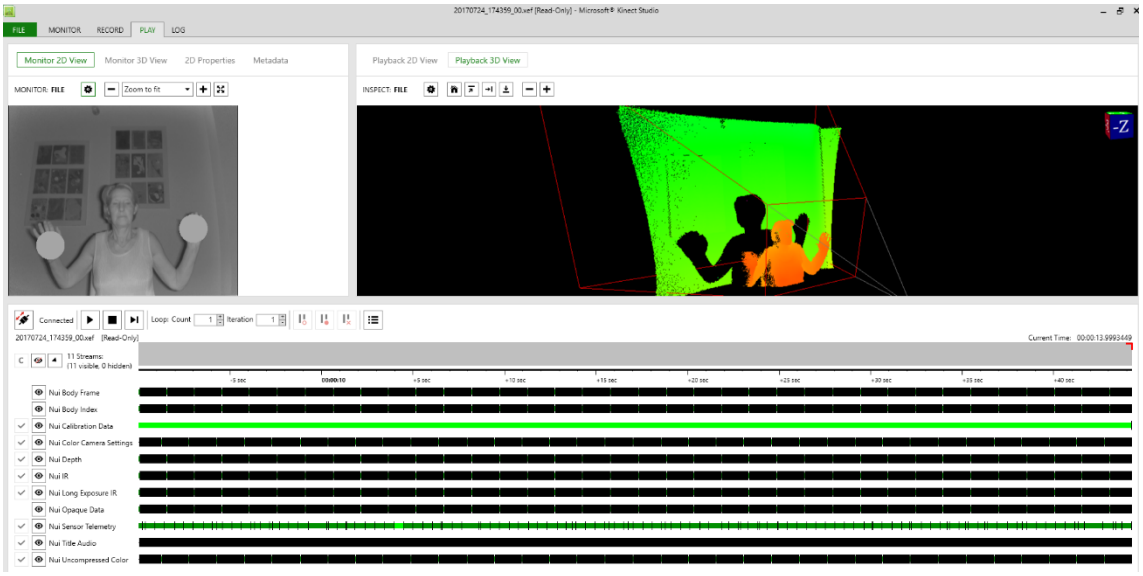
The pattern observed in this phase in 3d the vector direction was: Awe < Neutral <= Smile < Angry. While in Euclidean distance was: Angry <= Neutral = Smile < Awe. Based on the outcomes of this phase, we concluded that working with the Euclidean distance is enough and a good approach to evaluate and detect a facial expression. It is not necessary to calculate also the 3d vector direction angles, because it leads to a redundant conclusion and has an impact on real time evaluation of facial expressions.

3.3.3 Second phase

The second phase included 8 facial patterns, 30 subjects and used more control points to process and analyze data. The subjects were 15 women and 15 men, with age ranging from 12 to 73 as shown in Figure 24.

The number of control points considered per face area were:

- 8 - mouth area;
- 7 - right eyebrow and nose;
- 7 - left eyebrow and nose;
- 21 - the sum up of all the control points as can be seen in Figure 25.



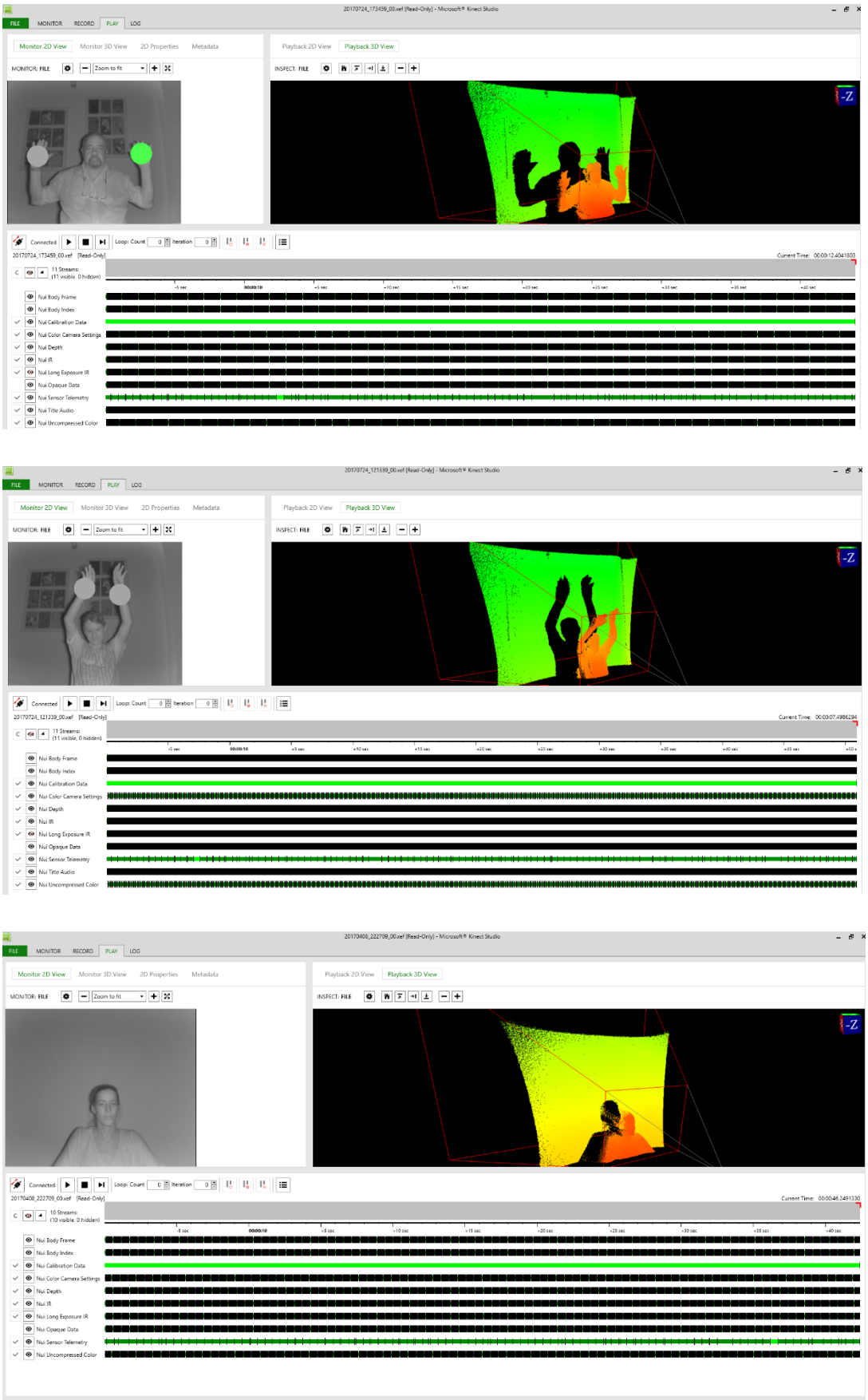


Figure 24 – Range of ages and genders in dataset

We avoided the wearing of glasses and guaranteed that the subjects had their eyebrows clearly detectable.

We asked each one of them to express in their faces the following patterns: neutral, smile, angriness, surprise, mouth to the side, lip bite, close eye and open mouth. The subjects stayed during 5 seconds in each expression individually and the result was recorded in 2 files.

The first file had neutral, smile, angriness and surprise and the second file had the neutral, mouth to the side, lip bite, close eye and open mouth. The recordings were done at a 30 frame rate per second. The subjects had to clearly initiate and terminate each facial expression, starting always from a neutral one.

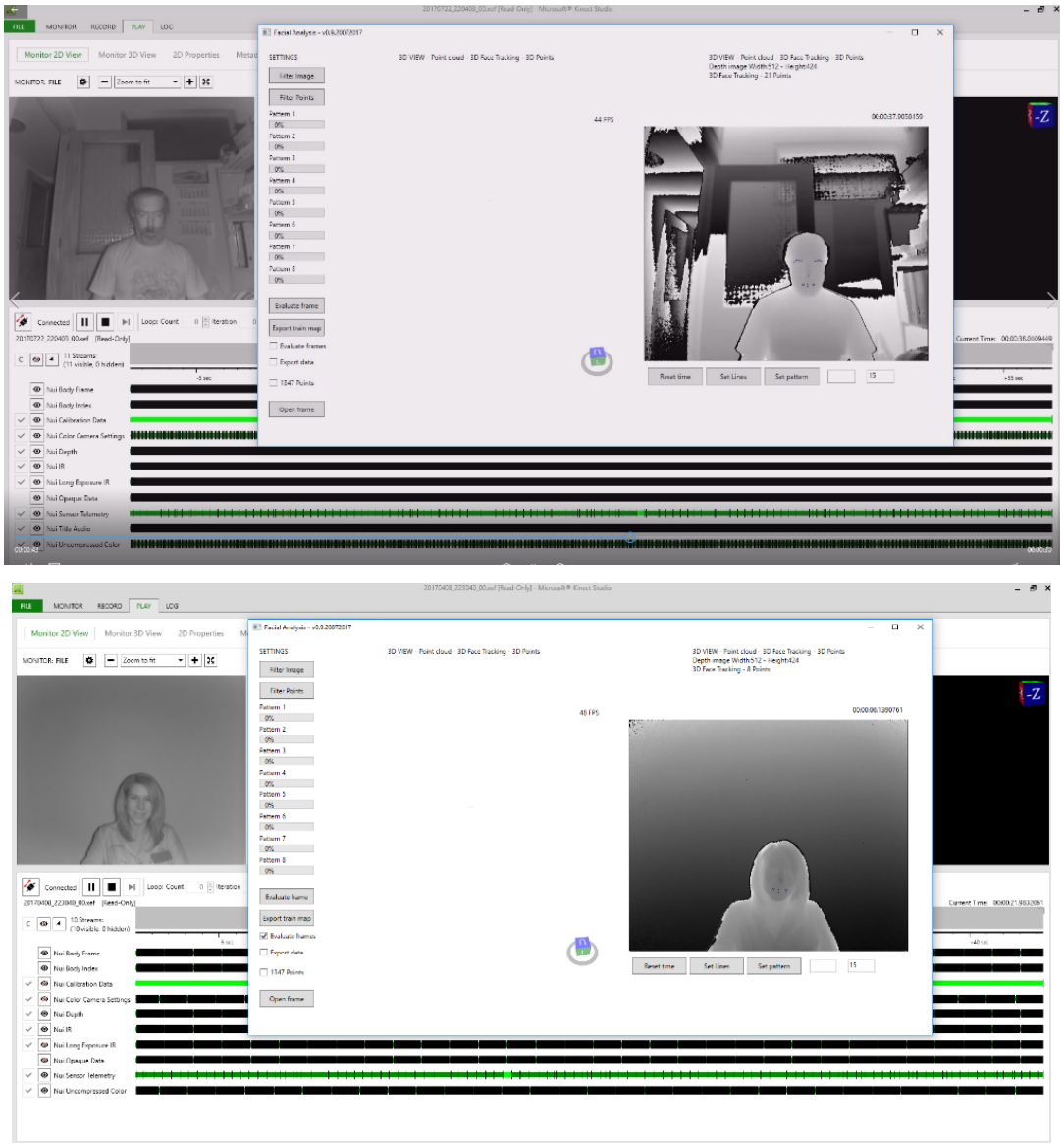


Figure 25 – Control points: 7 –left eyebrow + nose; 21 – eye brows + nose + mouth; 8 - mouth

In this phase only the Euclidean distances where calculated and feed to the neural network. There was also an improvement regarding the evaluation of the neural network thanks to the increase of the amount of data available to train. The LSTM trained model with 30 users was able to classify with an accuracy of 71% a test case with 6 users in the group of right eyebrow and nose

One problem encountered when using a group of control points, for example the right eyebrow and the nose (7 control points), was the classification error. This error occurred when the values of the Euclidean distance for one facial expression were equal to another one.

To solve this issue a different combination of 3 groups of control points was used. The groups were: the right eyebrow and the nose, the left eyebrow and the nose and the mouth, as shown in Figure 25. The classification of a facial expressions, when using the 21 points, is made by the three subgroups independently. When we combined the different control points groups, the accuracy goes to 90%. In addition, you can see in the YouTube channel playlist videos of the application running in real time and classifying the expressions [62].

Another issue was the diversity of the facial geometry of each subject. It generated, in some cases, values that are different from subject to subject. This issue was partially overcame with the use of the neural network that is able to detect patterns in big chunks of data.

3.4. **Conclusions**

In this, chapter several experimental tests were performed using our prototype. They were realized in two phases, considering different both the number of subjects and facial expressions. Some important inferences were possible to achieve in each of the phases and in the overall experiment.

Phase 1:

The main conclusions of phase 1 are summarized in table 4 and 5. Table 4 considers the facial patterns comparison between the six subjects while in table 5, each subject is analyzed alone, considering the average gap values between each facial expression.

Patterns	Euclidean distance or 3D vector direction	
	per each control point per all frames	per all control points per each frame
Awe	same pattern	different and varying each time
Angry		
Neutral		
Smile		

Table 4 - Comparison of results between subjects (charts in Figure 20 and Figure 21)

These results indicated that people present the same final pattern in their facial expressions although the facial points are completely different and does not present any kind of pattern neither between subjects nor to the same one. This means that to infer with precision what facial expression is being presented by the subject, it should be detected only when it meets its peak and is fully completed. In between, it is not possible or does not offer an acceptable level of certainty.

Patterns	Neutral	Angry	Smile
Awe	+6%	+13%	+8%
Neutral		+6%	+2%
Angry			-4%

Table 5 - Comparison between facial patterns - average gap values (Euclidean distance chart in Figure 21)

On the other hand, when analysing individually each subject, we can detect a pattern in the amplitude change of the control point values. Although two of the subjects did not show this change so clearly, the other four did.

The awe expression presented always higher values than the other expressions, while the angry one, the lowest. The smile and neutral expression had almost the same level, although some difference can be noted.

Looking at table 5 we can see that the awe expression had in average 13% higher values in comparison to the angry one, while in comparison to the smile one, only 8%. The angry expression has less 6% than the neutral, 4% than the smile and 13% than the awe one.

Phase 2:

In phase 2, the main results pointed out that even with occlusions and glasses the application can classify expressions due to the LSTM neural network model that makes possible some "remembering" of previously values over arbitrary time intervals. Also some people with long beard can induce errors in the feature estimation process by Kinect v2. This increases the error rate of the application.

Tables 6, 7 and 8 compares the average results between facial expressions considering Euclidean distance. In each table is considered different group of control points. As we

can see, it is possible to detect a pattern between the comparisons of the achieved results. For instance, the smile pattern in table one, is always similar to the neutral, mouth side, lip bite or close eye patterns, lower than the awe and the open mouth one and greater than the angry one.

Patterns	Neutral	Angry	Smile	Awe	Mouth side	Lip bite	Close eye (left)	Open mouth
Neutral		Neutral>	Neutral=	Neutral<	Neutral=	Neutral=	Neutral=	Neutral<
Angry	Angry<		Angry<	Angry<	Angry<	Angry<	Angry<	Angry<
Smile	Smile=	Smile>		Smile<	Smile=	Smile=	Smile=	Smile<
Awe	Awe>	Awe>	Awe>		Awe>	Awe>	Awe>	Awe>=
Mouth side	Mouth side=	Mouth side>	Mouth side=	Mouth side<		Mouth side=	Mouth side=	Mouth side<
Lip bite	Lip bite=	Lip bite>	Lip bite=	Lip bite<	Lip bite=		Lip bite=	Lip bite<
Close eye (left)	Close eye=	Close eye>	Close eye=	Close eye<	Close eye=	Close eye=		Close eye<
Open mouth	Open mouth>	Open mouth>	Open mouth>	Open mouth=	Open mouth>	Open mouth>	Open mouth>	

Table 6 – Average comparison between facial patterns – Euclidean distance in a frame with 7 points: right eyebrow and nose

Patterns	Neutral	Angry	Smile	Awe	Mouth side	Lip bite	Close eye (left)	Open mouth
Neutral		Neutral>=	Neutral<	Neutral<	Neutral<	Neutral=	Neutral=	Neutral<
Angry	Angry<=		Angry<	Angry<	Angry<	Angry<	Angry<	Angry<
Smile	Smile>	Smile>		Smile>	Smile>	Smile>	Smile>	Smile>=

Awe	Awe>	Awe>	Awe<		Awe>	Awe>	Awe>	Awe<=
Mouth side	Mouth side>	Mouth side>	Mouth side<	Mouth side<		Mouth side>	Mouth side>	Mouth side<
Lip bite	Lip bite=	Lip bite>	Lip bite<	Lip bite<	Lip bite<		Lip bite=	Lip bite<
Close eye (left)	Close eye=	Close eye>	Close eye<	Close eye<	Close eye<	Close eye=		Close eye<
Open mouth	Open mouth>	Open mouth>	Open mouth<=	Open mouth<=	Open mouth>	Open mouth>	Open mouth>	

Table 7 – Average comparison between facial patterns – Euclidean distance in a frame with 8 points: mouth

Patterns	Neutral	Angry	Smile	Awe	Mouth side	Lip bite	Close eye (left)	Open mouth
Neutral		Neutral>	Neutral=	Neutral<	Neutral=	Neutral=	Neutral>	Neutral<
Angry	Angry<		Angry<	Angry<	Angry<	Angry<	Angry<=	Angry<
Smile	Smile=	Smile>		Smile<	Smile=	Smile=	Smile>	Smile<
Awe	Awe>	Awe>	Awe>		Awe>	Awe>	Awe>	Awe>=
Mouth side	Mouth side=	Mouth side>	Mouth side=	Mouth side<		Mouth side=	Mouth side>	Mouth side<
Lip bite	Lip bite=	Lip bite>	Lip bite=	Lip bite<	Lip bite=		Lip bite>	Lip bite<
Close eye (left)	Close eye=	Close eye>	Close eye=	Close eye<	Close eye=	Close eye=		Close eye<

Open mouth	Open mouth>	Open mouth>	Open mouth>	Open mouth<=	Open mouth>	Open mouth>	Open mouth>	

Table 8 – Average ccomparison between facial patterns – Euclidean distance in a frame with 7 points: left eyebrow plus nose

In overall, the phase 2 validated the assumptions achieved in phase 1. Depth image and the detection of the face features, using control points and the calculation of the total values per frames for each control point, applying the Euclidean distance, is useful and supports well faces expression recognition.

4 Main results and future work

“I am just a child who has never grown up. I still keep asking these 'how' and 'why' questions. Occasionally, I find an answer.” Stephen Hawking

It is presented in this chapter, the main conclusions reached during this research, as well as a summary of the main aspects regarding the research activities, besides the potential future work to be developed

The Exploratory Qualitative Phase (lasted approximately 18 months), was the main focus in the initial phase of this PhD. In this phase, data was collected from secondary sources (through literature review) and from the exploratory (action research type) testing of some techniques, such as 3d face geometry tracking and detection.

The purpose of these exploratory testing was to illustrate the problem field, getting a good level of knowledge of it, thus shaping the potential future system prototype. This was absolutely crucial for the formulation of the research question and for the definition of the overall research design, providing the basis for the quantitative work. Although the bulk of this phase was carried out in the initial part of this PhD, the execution of additional testing was not discarded. Some tests were needed for the preparation of the instruments of the quantitative phase and for the clarification of the final results.

The action research contributed to this phase and was based on trying to answer this question: which is the most useful algorithm or/and sensor for analyzing 3d data in real time and identify facial expressions? In addition, it also contributed to develop a base system of the prototype centered on the best answer for this question. Of course, that the initial stage of this phase was based on a literature survey that gave the foundations of our system. Figure 26 illustrates the research cycle performed along this work.

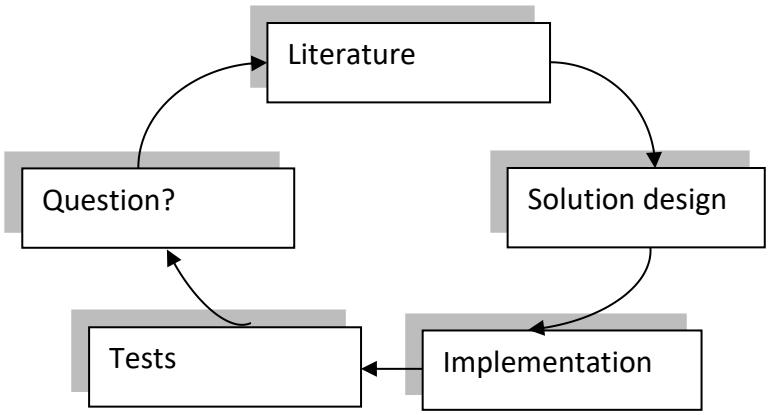


Figure 26 – Research cycle

The Quantitative Phase (lasted approximately 18 months) of the research (experimental research type). It was based on the implementation of a prototype system and a case study. Simply put, the idea was to implement a system based on the exploratory testing that allowed us to identify the facial expressions/patterns. The system was based on the exploratory phase, combined with a review of the literature. Once the system was implemented the case study was carried out with some test subjects. The goal of this case study was to validate the system by applying it, in a series of tests to evaluate the detection of facial patterns with 3d geometry data.

Our proposed method was simple. The main application is in Visual Studio 2015 (C#) with Kinect Studio v2 for record/play capabilities. Although we use Kinect v2 there are several equipment's that are based on the same technology and can be used with some reworking (Intel® RealSense™ Depth Camera D415 and D435 for example).

The face of the person was tracked and the recorded features coordinate values provided to the CNTK engine LSTM neural network. In the CNTK engine with the trained model, the neural network classifies and recognizes the facial expression and passes back the result to the main application, which displays it on the screen.

We used in the first phase of the facial analysis exploratory research 6 different subjects and a minimum of 7 features to recognize 4 facial patterns to reduce time and storage.

In the second phase, 21 features were considered to recognize 8 facial patterns with 30 subjects.

Some difficulties appeared when building the dataset; the persons with glasses and with hair in front of the face make some patterns hard to detect. The main dataset in the second phase achieved a result of 71% with the LSTM neural network with the group of nose and right eyebrows. Globally, the main dataset achieved a result of 90% in the second phase with all 3 groups aggregated.

Finally, it is important to highlight that the results achieved in this research, fully support that it is possible to classify a fusion of 3d and 2d data with deep learning and combine it to detect facial patterns. So, the answers to the 2 initial questions proposed in this study are:

- A fusion of 3d and 2d data **can be classified using deep learning networks (LSTM) and data visualization graphs**.
- 3d and 2d data **can be combined using the Euclidean distance/3d vector direction and facial features to detect facial patterns**.

For the future work, we consider that is necessary to build a bigger database, using people with different ages and ethnicities. Adding an unsupervised training feature to the application, to make the recognition more general, is also an interesting approach.

In addition, other areas of deployment could be used, such as the entertainment industry, by improving movies and games production pipeline, the human computer interaction and the security area, in face recognition and expression analysis, and in the automobile industry, regarding driver drowsiness detection.

Also, the use of this technology on a E-learning environment could be achieved as an evaluation of attention/distraction regarding facial patterns. The professor could estimate the engagement of the student’s thru the online lesson.

5 Bibliography

[1] Craig S. D., D'Mello S., Witherspoon A., Graesser A.: "Emote aloud during learning with auto tutor: Applying the facial action coding system to cognitive–affective states during learning". Cognition and Emotion 22, 5, 777– 788 (2008).

[2] Dinges D. F., Rider R. L., Dorrian J., Mcglinchey E. L., Rogers N. L., Cizman Z., Goldenstein S. K., Vogler C., Venkataraman S., Metaxas D. N.: "Optical computer recognition of facial expressions associated with stress induced by performance demands". Aviation, space, and environmental medicine 76, Supplement 1, B172–B182 (2005).

[3] Sharma N., Gedeon T.: "Objective measures, sensors and computational techniques for stress recognition and classification: A survey". Computer methods and programs in biomedicine 108, 3, 1287–1301 (2012).

[4] X. Fan, Q. Peng, and M. Zhong : "3D Face Reconstruction from Single 2D Image Based on Robust Facial Feature Points Extraction and Generic Wire Frame Model". *International Conference on Communications and Mobile Computing*, pp. 396-400, (2010).

[5] Amel Aissaoui, Jean Martinet, Chaabane Djeraba : "Rapid and accurate face depth estimation in passive stereo systems". Multimedia Tools and Applications. 72. 10.1007/s11042-013-1556-z. (2014).

[6] P A Tjahyaningtijas, H & Puspitasari, P & Yamasari, Y & Anifah, L & A Buditjahyanto : "Method Comparison of 3D Facial Reconstruction Coresponding to 2D Image". IOP Conference Series: Materials Science and Engineering. 288. 012073. 10.1088/1757-899X/288/1/012073 (2018).

[7] J. Endres and A. Laidlaw : "Micro-expression recognition training in medical students: a pilot study". *BMC Medical Education*, vol. 9, p. 47, (2009).

[8] R. W. Picard : "Future affective technology for autism and emotion communication". *Philosophical Transactions of the Royal Society of London - Series B: Biological Sciences*, vol. 364, no. 1535, pp. 3575-3584, (2009).

[9] V. Vaishnavi and W. Kuechler : "*Design Science Research Methods and Patterns*". Auerbach Publications, (2008).

[10] J. W. Creswell : "*Research Design: Qualitative, Quantitative, and Mixed Methods Approaches".* vol. 7, no. 1. Sage Publications, p. 260 (2008).

[11] K. Srinagesh : "*The Principles of Experimental Research".* no. December. Butterworth-Heinemann, pp. 11-12 (2005).

[12] D. E. Avison, F. Lau, M. D. Myers, and P. A. Nielsen : “Action research”. *Communications of the ACM*, vol. 42, no. 1, pp. 94-97, (1999).

[13] M. Brydon-Miller, D. Greenwood, and P. Maguire : “Why Action Research?”. *Action Research*, vol. 1, no. 1, pp. 9-28, (2003).

[14] M. V. Zelkowitz and D. R. Wallace : “Experimental models for validating technology”. *Computer*, vol. 31, no. 5, pp. 23-31, (1998).

[15] A. Dix, J. Finlay, G. D. Abowd, and R. Beale : "*Human-computer interaction".* vol. Third, no. 3. Prentice Hall, p. 834 (2004).

[16] Jorg Richter, Jurij Poelchau : “DeepaMehta -- Another Computer is Possible”. in *Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications*, I. Global, Ed. (2008).

[17] E. Hjelmås : “Face Detection: A Survey”. *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 236-274, (2001).

[18] C. Zhang and Z. Zhang : “A Survey of Recent Advances in Face Detection”. *Learning*, no. June, p. 17, (2010).

[19] D. J. Kriegman : “Detecting faces in images: a survey”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34-58, (2002).

[20] K. Toyama : “Look, ma-no hands! hands-free cursor control with real-time 3d face tracking”. in *Proc Workshop on Perceptual User Interfaces PUI98*, pp. 49–54 (1998).

[21] D. O. Gorodnichy : “On importance of nose for face tracking”. *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, no. May, pp. 188-193, (2002).

[22] U. R. Dhond and J. K. Aggarwal : “Structure from stereo-a review”. *Ieee Transactions On Systems Man And Cybernetics*, vol. 19, no. 6, pp. 1489-1510, (1989).

[23] R. Zhang, P. S. Tsai, J. E. Cryer, and M. Shah : “Shape-from-shading: a survey”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 690-706, (1999).

[24] M. Subbarao, T. S. Choi, and A. Nikzad : “Focusing Techniques”. *Optical Engineering*, vol. 32, no. 11, pp. 2824-2836, (1993).

[25] M. Levoy et al.: “The Digital Michelangelo Project”. in *Proceedings of SIGGRAPH 2000*, vol. 18, pp. 131-144 (2000).

[26] J. Batlle, E. Mouaddib, and J. Salvi : “Recent progress in coded structured light as a technique to solve the correspondence problem: a survey”. *Pattern Recognition*, vol. 31, no. 7, pp. 963-982, (1998).

[27] D. Fofi, T. Sliwa, and Y. Voisin : “A comparative survey on invisible structured light”. in *Society of PhotoOptical Instrumentation Engineers SPIE Conference Series*, vol. 5303, pp. 90-98 (2004).

[28] J. Salvi : “Pattern codification strategies in structured light systems”. *Pattern Recognition*, vol. 37, no. 4, pp. 827-849, (2004).

[29] P. Ekman : “Basic emotions”. in *Handbook of cognition and emotion*, vol. 14, no. 1992, T. Dalgleish and M. Power, Eds. John Wiley & Sons, pp. 45-60 (1999).

[30] K. S. Haggard, E. A., & Isaacs : “Micro-momentary facial expressions as indicators of ego mechanisms in psychotherapy”. *Methods of Research in Psychotherapy*, pp. 154-165, (1996).

[31] M. Obaid, M., Mukundan, R., Billinghurst : “3D Facial Expression Analysis and Representations”. in *In NZCSRSC 2009: New Zealand Computer Science Research Student Conference. Auckland*, (2009).

[32] P. Branco : “Computer-based facial expression analysis for assessing user experience”. University of Minho, (2006).

[33] K. Zakharov : “Affect Recognition and Support in Intelligent Tutoring Systems”. (2007).

[34] M. Breidt, C. Wallraven, D. Cunningham, and H. Buelthoff : “Facial Animation Based on 3D Scans and Motion Capture”. *ACM SIGGRAPH 2003 Sketches Applications*, p. 1, (2003).

[35] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer : “High resolution passive facial performance capture”. *ACM Transactions on Graphics*, vol. 29, no. 4, p. 1, (2010).

[36] H. Huang : “Leveraging Motion Capture and 3D Scanning for High-fidelity Facial Performance Acquisition”. *Scanning*, (2011).

[37] T. Weise, S. Bouaziz, and H. Li : “Realtime Performance-Based Facial Animation”. *lggepflch*, (2011).

[38] S. Izadi et al.: “KinectFusion : Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera”. *interactions*, pp. 559–568, (2011).

[39] R. A. Newcombe et al.: “KinectFusion : Real-Time Dense Surface Mapping and Tracking”. *ISMAR*, pp. 127-136, (2011).

[40] P. Ekman, W. V. Friesen, and J. C. Hager : “Facial Action Coding System: The manual”. Research Nexus division of Network Information Research Corporation, (2002).

[41] J.-L. Minoi and D. Gillies : “3D facial expression analysis and deformation”. *Proceedings of the 4th symposium on Applied perception in graphics and visualization APGV 07*, vol. 1, no. 212, p. 138, (2007).

[42] I. A. Essa and A. P. Pentland : “Coding, analysis, interpretation, and recognition of facial expressions”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 757-763, (1997).

[43] M. Al Haj, J. Orozco, J. Gonzalez, and J. J. Villanueva : “Automatic face and facial features initialization for robust and accurate tracking”. *2008 19th International Conference on Pattern Recognition*, pp. 1-4, (2008).

[44] F. Dornaika and J. Orozco : “Real time 3D face and facial feature tracking”. *Journal of RealTime Image Processing*, vol. 2, no. 1, pp. 35-44, (2007).

[45] P. Ekman : “Facial expressions of emotion: an old controversy and new findings”. *Philosophical Transactions of the Royal Society of London - Series B: Biological Sciences*, vol. 335, no. 1273, pp. 63-69, (1992).

[46] B. Amberg, S. Romdani, and T. Vetter : “Optimal Step Nonrigid ICP Algorithms for Surface Registration”. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 14, no. 1-3, pp. 1-8, (2007).

[47] B. Hu , S. You : “Semantic and case-based reasoning for facial expression recognition”. Journal of Information and Computational Science 7(9):1868-1877 (2010).

[48] Aasia Khanum, Muid Mufti, M. Younus Javed, M. Zubair Shafiq : “Fuzzy case-based reasoning for facial expression recognition
”. Journal Fuzzy Sets and Systems,Volume 160 Issue 2, January, Pages 231-250 (2009).

[49] Hesham A. Alabbasi, Prof.Florica Moldoveanu, Prof. Alin Moldoveanu : "Real Time Facial Emotion Recognition using Kinect V2 Sensor". IOSR Journal of Computer Engineering (IOSR-JCE), Volume 17, Issue 3, Ver. II, PP 61-68 (2015).

[50] Monkaresi, Hamed & Bosch, Nigel & Calvo, Rafael & D'Mello, Sidney : "Automated Detection of Engagement Using Video-Based Estimation of Facial Expressions and Heart Rate". IEEE Transactions on Affective Computing. 8. 1-1. 10.1109/TAFFC.2016.2515084 (2016).

[51] Garay, Nestor; Idoia Cearreta; Juan Miguel López; Inmaculada Fajardo : "Assistive Technology and Affective Mediation". Human Technology. 2 (1): 55–83 (2006).

[52] Ekman, P. & Friesen, W. V. : "The repertoire of nonverbal behavior: Categories, origins, usage, and coding". Semiotica, 1, 49–98. (1969).

[53] Silverstein, Evan & Snyder, Michael. : "Implementation of Facial Recognition with Microsoft Kinect v2 Sensor for Patient Verification". Medical physics. 44. . 10.1002/mp.12241 (2017).

[54] Hesham A. Alabbasi, Prof.Florica Moldoveanu, Prof. Alin Moldoveanu : " Real Time Facial Emotion Recognition using Kinect V2 Sensor". IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. II, PP 61-68 (2015).

[55] Al-Darraji, Salah & Berns, Karsten & Rodić, Aleksandar. : "Action Unit Based Facial Expression Recognition Using Deep Learning". 540. 413-420. 10.1007/978-3-319-49058-8_45 (2017).

[56] Marcolin, Federica & Violante, Maria Grazia & Moos, Sandro & Vezzetti, Enrico & Tornincasa, Stefano & Dagnes, Nicole & Speranza, Domenico : "Three-dimensional face analysis via new geometrical descriptors". Lecture Notes in Mechanical Engineering. 747-756. 10.1007/978-3-319-45781-9_75 (2017).

[57] Tian, Lei & Fan, Chunxiao & Ming, Yue : "Multiple scales combined principle component analysis deep learning network for face recognition". Journal of Electronic Imaging. 25. 023025. 10.1117/1.JEI.25.2.023025 (2016).

[58] Lee, Injae & Jung, Heechul & Hyun Ahn, Chung & Seo, Jeongil & Kim, Junmo & Kwon, Ohseok : "Real-time personalized facial expression recognition system based on deep learning". 267-268. 10.1109/ICCE.2016.7430609 (2016).

[59] Corrêa, Débora & Salvadeo, Denis & Levada, Alexandre & Saito, José : "Using LSTM Network in Face Classification Problems". (2008)

[60] Tolosana, Ruben & Vera-Rodriguez, Ruben & Fierrez, Julian & Ortega-Garcia, Javier : "Exploring Recurrent Neural Networks for On-Line Handwritten Signature Biometrics". IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2793966 (2018).

[61] Gachie, Wanjiru & Govender, Desmond : "The evaluation of human computer interface design of learning management systems: problems and perspectives. Problems and Perspectives in Management". 15. 394-410. 10.21511/ppm.15(3-2). (2017).

[62] Videos of the prototype http://www.dailymotion.com/playlist/x57hzi and https://www.youtube.com/playlist?list=PLPIdYBlZfCu69Qq092cGqFRUzZ4MZK3TZ (visited on 14/02/1018) .

[63] CNTK https://www.microsoft.com/en-us/cognitive-toolkit/ (visited on 14/02/1018).

[64] H. El-Hori, Inas & K. El-Momen, Zahraa & Ganoun, Ali : "PCA facial expression recognition". 9067. . 10.1117/12.2051196 (2013).

[65] Micro Expressions trainings tools https://www.paulekman.com/micro-expressions-training-tools/ (visited on 14/02/1018).

[66] Deshmukh, Renuka & Paygude, Shilpa & Jagtap, Vandana : "Facial Emotion Recognition System through Machine Learning approach". 10.1109/ICCONS.2017.8250725 (2017).

[67] Datta, Navita & Kumar, Rajeev & Bhardwaj, Reeta : "Reduction of False Alarm Rate by using K-NN and Naive Bayes: A Review". International Journal of Computer Applications. 180. 3-6. 10.5120/ijca2017915985 (2017).

[68] Understanding LSTM Networks https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 14/02/1018).

[69] Hongfei Lin, Fengming Pan, Yuxuan Wang, ShaohuaLv and Shichang Sun "Affective Computing in Elearning". E-learning, MajaJakobovic (Ed.), ISBN: 978953-7619-95-4, InTech (2010).

6 Appendix

6.1. **Appendix 1**

Presentation 1

Task oriented sequence for gathering user data: neutral, smile, angry and awe.

Tarefas

Faça uma expressão zangada

Tarefas

Faça uma expressão neutra

Tarefas

Faça uma expressão de admiração

Tarefas

Faça uma expressão neutra

FIM

Tarefas terminadas

Obrigado

6.2. **Appendix 2**

Presentation 2

Task oriented sequence for gathering user data: neutral, mouth side, lip bite, close eye and open mouth.

Tarefas

Faça a seguinte expressão : morda os lábios

Tarefas

Faça uma expressão neutra

Tarefas

Faça a seguinte expressão : feche um olho

Tarefas

Faça uma expressão neutra

Tarefas

Faça a seguinte expressão : abrir boca

Tarefas

Faça uma expressão neutra

FIM

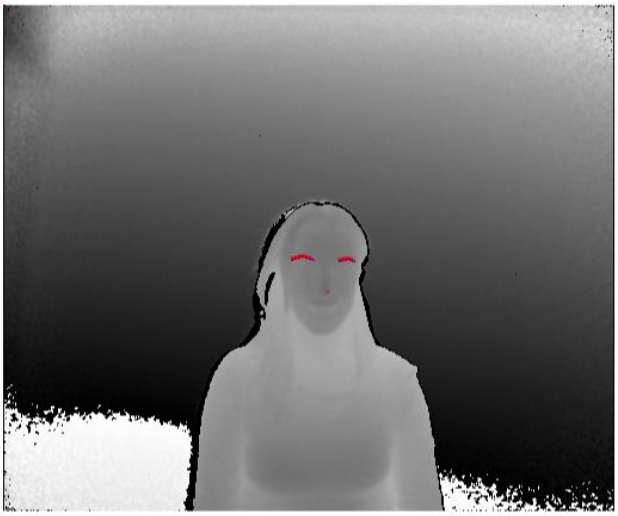
Tarefas terminadas

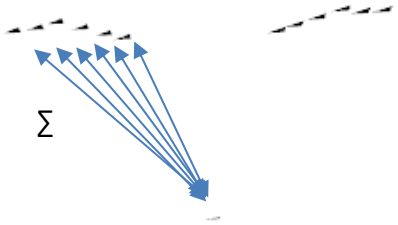
Obrigado

6.3. **Appendix 3**

Calculation of Euclidean distance

Image with facial features with annotations explaining the calculation of Euclidean distance.





⟷	Euclidean distance between the reference control point (located on the nose) and the control points of the right eyebrows.
∑	All the distances are summed in a single frame resulting in a reference value for a facial expression in a frame.

Related code in C# :

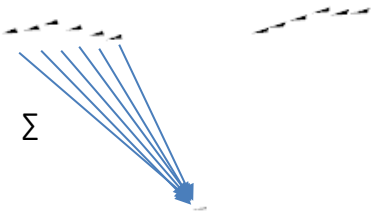
…

```
// Euclidean distance
if (flagEvaluatePointsContinuous)
{
//store the distance of individual points in a list
List<double> VectorFrame = new List<double>() { };
angMedTot = 0;
        for (int v = 1; v < FilteredPointCloud.Count; v++)
        // for each point calculate the distance and add sum that value to angMedTot
        {
        // FilteredPointCloud[0] – reference control point ex: nose or upper lip point
        double deltaX = Math.Abs(FilteredPointCloud[0].X - FilteredPointCloud[v].X);
        double deltaY = Math.Abs(FilteredPointCloud[0].Y - FilteredPointCloud[v].Y);
        double deltaZ = Math.Abs(FilteredPointCloud[0].Z - FilteredPointCloud[v].Z);
        double t = Math.Round(Math.Sqrt((deltaX * deltaX) + (deltaY * deltaY) + (deltaZ *
deltaZ)), 2, MidpointRounding.AwayFromZero);
        VectorFrame.Add(t);
        angMedTot = angMedTot + t;
        }
}
```

6.4. **Appendix 4**

Calculation of the 3d vector direction

Image with facial features with annotations explaining the calculation of the 3d vector direction.



→	Directions of the 3d vectors between the reference control point (nose) and each control point of the right eyebrows.
∑	All the values are summed in a single frame resulting in a reference value for a facial expression in a frame.

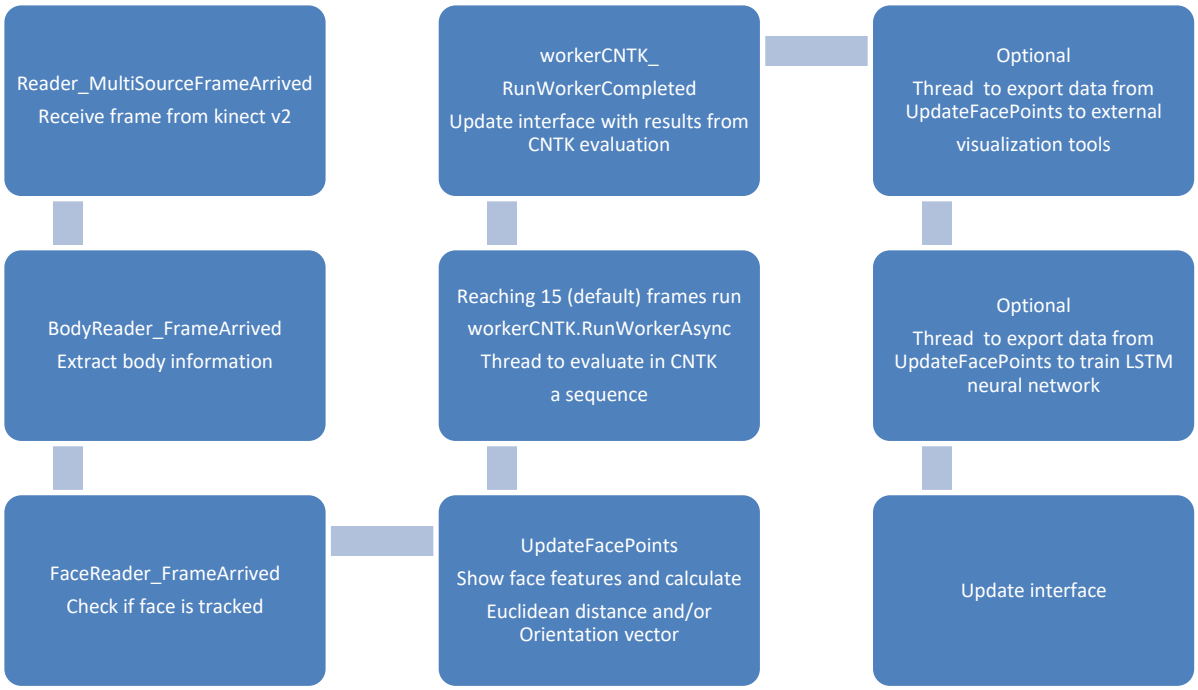
Related code in C# :

```
…

// 3d vector direction

if (flagEvaluatePointsContinuous)

{

//store the vector direction of individual points in a list

List<double> VectorFrame = new List<double>() { };

angMedTot = 0;

        for (int v = 1; v < FilteredPointCloud.Count; v++)

        // for each point calculate the orientation vector and add sum that value to angMedTot

        {

        // FilteredPointCloud[0] – reference control point ex: nose or upper lip point

        double deltaX = Math.Abs(FilteredPointCloud[0].X - FilteredPointCloud[v].X);

        double deltaY = Math.Abs(FilteredPointCloud[0].Y - FilteredPointCloud[v].Y);

        double deltaZ = Math.Abs(FilteredPointCloud[0].Z - FilteredPointCloud[v].Z);

        double t = Math.Round(Math.Sqrt((deltaX * deltaX) + (deltaY * deltaY) + (deltaZ * deltaZ)), 2, MidpointRounding.AwayFromZero);

        double vd = ( (Math.Acos(deltaX/t) * 180) / Math.PI) + ((Math.Acos(deltaY/t)*180)/Math.PI) +((Math.Acos(deltaZ/t)*180)/Math.PI)

        VectorFrame.Add(vd);

        angMedTot = angMedTot + vd;

        }

}
```

6.5. Appendix 5

Detailed algorithm flowchart

Flowchart with code steps representing the algorithm.



6.6. **Appendix 6**

Code related to improving real time facial analysis and kinect sensor.

This section is to illustrate some of the code used to improve real time facial analysis and Kinect sensor.

```
//Initialize sensor and setup events to receive infomation
private void Window_Loaded(object sender, RoutedEventArgs e)
 {
    _sensor = KinectSensor.GetDefault();
    if (_sensor != null)
    {
        _bodySource = _sensor.BodyFrameSource;
        _bodyReader = _bodySource.OpenReader();
        _bodyReader.FrameArrived += BodyReader_FrameArrived;
        _faceSource = new HighDefinitionFaceFrameSource(_sensor);
        _faceReader = _faceSource.OpenReader();
        _faceReader.FrameArrived += FaceReader_FrameArrived;
        _faceModel = new FaceModel();
        _faceAlignment = new FaceAlignment();
        _sensor.Open();
        _reader = _sensor.OpenMultiSourceFrameReader(FrameSourceTypes.Color | FrameSourceTypes.Depth | FrameSourceTypes.Infrared | FrameSourceTypes.Body);

        _reader.MultiSourceFrameArrived += Reader_MultiSourceFrameArrived;
    }
 }
```

```
// Event that receive depth frame from Kinect sensor
void Reader_MultiSourceFrameArrived(object sender, MultiSourceFrameArrivedEventArgs e)
    {
      VideoStampTime.Start();
      var reference = e.FrameReference.AcquireFrame();
      // Depth
      using (var frame = reference.DepthFrameReference.AcquireFrame())
       {
        if (frame != null)
        {
          TimeSpan frameID = frame.RelativeTime;
          TimeSpan aux_timespan = frameID.Subtract(StampTime2);
          if (aux_timespan < new TimeSpan(0, 0, 5))
          {
            TotalStampTime = TotalStampTime + aux_timespan ;
            VisualFrameID.Content = TotalStampTime + new TimeSpan(0, 0, 0, 0, 105);
          }
          StampTime2 = frameID;
          if (_mode == Mode.Depth)
          {
              w = frame.FrameDescription.Width;
              h = frame.FrameDescription.Height;
              camera.Source = frame.ToBitmap();
          }
        }
      }
      if ((frameFlag) && (points_tracking > 600))
```

```
        {
            if (p1347)
                facetrack.Text = "3D VIEW - Point cloud - 3D Face Tracking - 3D Points\r\nDepth image Width:" + w + " - Height:" + h + System.Environment.NewLine + "3D Face Tracking - " + points_tracking + " Points";
            else if (p35)
                facetrack.Text = "3D VIEW - Point cloud - 3D Face Tracking - 3D Points\r\nDepth image Width:" + w + " - Height:" + h + System.Environment.NewLine + "3D Face Tracking - " + List_Tracking_Points.Count + " Points";
            frameFlag = false;
        }
         }
```

// Event that receive body information frame from Kinect sensor

```
private void BodyReader_FrameArrived(object sender, BodyFrameArrivedEventArgs e)
    {
        using (var frame = e.FrameReference.AcquireFrame())
        {
            if (frame != null)
            {
                Body[] bodies = new Body[frame.BodyCount];
                frame.GetAndRefreshBodyData(bodies);
                Body body = bodies.Where(b => b.IsTracked).FirstOrDefault();
                if (!_faceSource.IsTrackingIdValid)
                {
                    if (body != null)
                    {
                        _faceSource.TrackingId = body.TrackingId;
```

```
            }
        }
    }
  }
}
```

// Event that receive face information frame from Kinect sensor

```
private void FaceReader_FrameArrived (object sender, HighDefinitionFaceFrameArrivedEventArgs e)
  {
    using (var frame = e.FrameReference.AcquireFrame())
    {
      if (frame != null && frame.IsFaceTracked)
      {
        frame.GetAndRefreshFaceAlignmentResult(_faceAlignment);
        UpdateFacePoints(frame.RelativeTime);
      }
    }
  }
```

// Main method that evaluates the face points of frame from Kinect sensor

```
  private void UpdateFacePoints(TimeSpan frameID)
  {
    if (_faceModel == null) return;
```

```
var vertices = _faceModel.CalculateVerticesForAlignment(_faceAlignment);
points_tracking = vertices.Count;
if (vertices.Count > 0)
{
   if (_points.Count == 0) // only execute once to fill the canvas with points and lines
   {
      for (int index = 0; index < vertices.Count; index++)
      {
         var cor = new SolidColorBrush(Colors.Indigo);
         if (index == 18) //nose
            cor = new SolidColorBrush(Colors.Red);
         Ellipse ellipse = new Ellipse
         {
            Width = 2.0,
            Height = 2.0,
            Fill = cor,
            Tag = index
         };
         Line redLine = new Line();
         redLine.X1 = 0;
         redLine.Y1 = 0;
         redLine.X2 = 0;
         redLine.Y2 = 0;
         // Create a red Brush
         SolidColorBrush redBrush = new SolidColorBrush();
         redBrush.Color = Colors.Red;
         redLine.StrokeThickness = 3;
```

```
                redLine.Stroke = redBrush;
                _linhas.Add(redLine);
                _points.Add(ellipse);
            }
            foreach (Line linha in _linhas)
            {
                canvas.Children.Add(linha);
            }
            foreach (Ellipse ellipse in _points)
            {
                canvas.Children.Add(ellipse);
            }
        }
        strBuffer.Clear();
        TimeSpan addMili = new TimeSpan(0, 0, 0, 0, 105);
        var str = " " + frameID + " ";
        var TempoMili = TotalStampTime + addMili;
        strBuffer.Append("\r\n" + TempoMili.ToString() + " ");
        for (int index = 0; index < vertices.Count; index++)
        {
            if ((List_Tracking_Points.Contains(index)) && (p35))
            {
                CameraSpacePoint vertice = vertices[index];
                DepthSpacePoint point =
_sensor.CoordinateMapper.MapCameraPointToDepthSpace(vertice);
                if (float.IsInfinity(point.X) || float.IsInfinity(point.Y)) return;
                if (index == 18)
                        { FaceCenter.X = point.X; FaceCenter.Y = point.Y; }
```

```
                Ellipse ellipse = _points[index];
                Canvas.SetLeft(ellipse, point.X);
                Canvas.SetTop(ellipse, point.Y);

                Line redLine = _linhas[index];
                if(setlines_flag) UpdateLines(index,redLine,vertices,point);
                else ResetLines(index, redLine, vertices, point);
                LastPoint.X = point.X;
                LastPoint.Y = point.Y;

                strBuffer.Append(index + " " + vertices[index].X + " " + vertices[index].Y + " " +
vertices[index].Z + " ");
                if (List_Tracking_Points_LeyeNose.Contains(index))
                  FilteredPointCloud.Add(new Point3D(-vertices[index].X, -vertices[index].Y,
vertices[index].Z));
            }
            else if (p1347)
            {
                CameraSpacePoint vertice = vertices[index];
                DepthSpacePoint point =
_sensor.CoordinateMapper.MapCameraPointToDepthSpace(vertice);

                if (float.IsInfinity(point.X) || float.IsInfinity(point.Y)) return;

                Ellipse ellipse = _points[index];

                Canvas.SetLeft(ellipse, point.X);
                Canvas.SetTop(ellipse, point.Y);
```

```
            strBuffer.Append(index + " " + vertices[index].X + " " + vertices[index].Y + " " +
vertices[index].Z + " ");

            FilteredPointCloud.Add(new Point3D(-vertices[index].X, -vertices[index].Y,
vertices[index].Z));

          }

        }

        if(flagExportPoints)

          AddText(fs, strBuffer.ToString()); // Save to file points


        // Euclidean/ 3d vector direction

        if (flagEvaluatePointsContinuous)

        {

            List<double> VectorFrame = new List<double>() { };

            angMedTot = 0;

            double bce = 0;

            for (int v = 1; v < FilteredPointCloud.Count; v++)

            {

              double deltaX = Math.Abs(FilteredPointCloud[0].X - FilteredPointCloud[v].X);

              double deltaY = Math.Abs(FilteredPointCloud[0].Y - FilteredPointCloud[v].Y);

              double deltaZ = Math.Abs(FilteredPointCloud[0].Z - FilteredPointCloud[v].Z);

              double t = Math.Round(Math.Sqrt((deltaX * deltaX) + (deltaY * deltaY) +
(deltaZ * deltaZ)), 2, MidpointRounding.AwayFromZero);

              VectorFrame.Add(t);

              angMedTot = angMedTot + t;

            }

            angMedTot = angMedTot + (5*bce);

            PSequence.Add(angMedTot.ToString("0.00"));
```

```
            PVector.Add(VectorFrame);

        if (PVector.Count >= NFramesEvaluation) // classify only the sequence with n frames
        {
            sequence = String.Join(" ", PSequence.ToArray());
            stringVectores.Append(VisualFrameID.Content + " " + sequence + "\r\n");
            List<string> args = new List<string>() { };
            args.Add(sequence);
            args.Add(VisualFrameID.Content.ToString());
             // parallel task that classify, to improve real time face analysis
            if (!(workerCNTK2.IsBusy)) workerCNTK2.RunWorkerAsync(args);
            using (StreamWriter w = File.AppendText(@"C:\tmp\train.txt")) { Log(stringVectores.ToString(), w); }

            stringVectores.Clear();
            angMedTot = 0;
            PVector.Clear();
            PSequence.Clear();
        }
    }


    if (p1347)
        {
        if (!(refreshFrameRate < 10))
        {
            refreshFrameRate = 0;
```

```
            }
            else
               refreshFrameRate++;
            }
          else
            t.UpdateModel(FilteredPointCloud);
          FilteredPointCloud.Clear();
        }
        LastPoint.X = 0;
        LastPoint.Y = 0;
         }
```

// Method that improves real time face analysis by creating parallel processing

```
private void workerCNTK2_DoWork(object sender, DoWorkEventArgs e)
     {
        // run all background tasks here
        List<string> seq = (List < string >) e.Argument;
       // evaluateLSTM_server_fast.py must be running
        e.Result = ExecuteConsoleApplication(@"C:\local\Anaconda3-4.1.1-Windows-x86_64\python.exe", "C:/local/cntk/Tutorials/evaluateLSTM_client_fast.py \"" + seq[0] + "\"");
     }
```

// Method that process information to UI when background task are completed

```
private void workerCNTK2_RunWorkerCompleted(object sender, RunWorkerCompletedEventArgs e)
```

```
{
    //update ui once worker complete his work
    string resultCNTK = e.Result.ToString();
    int classResult = 99;
    if (resultCNTK.Length>40)
        classResult = Int32.Parse(resultCNTK.Substring(1, 1));
    else if ((resultCNTK.Length < 40) && (resultCNTK.Length >0))
        classResult = Int32.Parse(resultCNTK.Substring(0,1));
    switch (classResult)
    {
        case 0: //AWE
            Pt0.Value = 100; Pt1.Value = 0; Pt2.Value = 0; Pt3.Value = 0; Pt4.Value = 0; Pt5.Value = 0; Pt6.Value = 0; Pt7.Value = 0;
            textBox3.Content = (Int32.Parse(textBox3.Content.ToString()) + 1).ToString();
            break;
        case 1: //ANGRY
            Pt0.Value = 0; Pt1.Value = 100; Pt2.Value = 0; Pt3.Value = 0; Pt4.Value = 0; Pt5.Value = 0; Pt6.Value = 0; Pt7.Value = 0;
            textBox4.Content = (Int32.Parse(textBox4.Content.ToString()) + 1).ToString();
            break;
        case 2: //NEUTRAL
            Pt0.Value = 0; Pt1.Value = 0; Pt2.Value = 100; Pt3.Value = 0; Pt4.Value = 0; Pt5.Value = 0; Pt6.Value = 0; Pt7.Value = 0;
            textBox5.Content = (Int32.Parse(textBox5.Content.ToString()) + 1).ToString();
            break;
        case 3://SMILE
            Pt0.Value = 0; Pt1.Value = 0; Pt2.Value = 0; Pt3.Value = 100; Pt4.Value = 0; Pt5.Value = 0; Pt6.Value = 0; Pt7.Value = 0;
            break;
```

```
            case 4: //lipbite

                Pt0.Value = 0; Pt1.Value = 0; Pt2.Value = 0; Pt3.Value = 0; Pt4.Value = 100; Pt5.Value = 0; Pt6.Value = 0; Pt7.Value = 0;

                textBox3.Content = (Int32.Parse(textBox3.Content.ToString()) + 1).ToString();

                break;

            case 4: //mouthside

                Pt0.Value = 0; Pt1.Value = 0; Pt2.Value = 0; Pt3.Value = 0; Pt4.Value = 0; Pt5.Value = 100; Pt6.Value = 0; Pt7.Value = 0;

                textBox4.Content = (Int32.Parse(textBox4.Content.ToString()) + 1).ToString();

                break;

            case 5://openmouth

                Pt0.Value = 0; Pt1.Value = 0; Pt2.Value = 0; Pt3.Value = 0; Pt4.Value = 0; Pt5.Value = 0; Pt6.Value = 100; Pt7.Value = 0;

                break;

            case 6:// closeeye

                Pt0.Value = 0; Pt1.Value = 0; Pt2.Value = 0; Pt3.Value = 0; Pt4.Value = 0; Pt5.Value = 0; Pt6.Value = 0; Pt7.Value = 100;

                break;

            default:

                Pt0.Value = 0; Pt1.Value = 0; Pt2.Value = 0; Pt3.Value = 0; Pt4.Value = 0; Pt5.Value = 0; Pt6.Value = 0; Pt7.Value = 0;

                break;

        }

        textBox2.Content = (Int32.Parse(textBox2.Content.ToString()) + 1).ToString();

    }
```