Department of Science and Information Technology

Portuguese Sign Language Recognition via Computer Vision and Depth Sensor

Rui Nuno de Almeida

A Dissertation presented in partial fulfillment of the Requirements for the Degree of

Master of Computer Science

Supervisor:
Professor Miguel Sales Dias, Invited Associated Professor,
ISCTE-IUL Lisbon University Institute

October, 2011

Acknowledgements

My First acknowledgment must go to my supervisor Prof. Dr. Miguel Sales Dias. During the journey of this study, he supported me in every aspect. He was the one who presented me to this research area and he inspired me with his enthusiasm on research and showed me ways to go when making science. I learned much from him and this thesis would not have been possible without him.

I owe special gratitude to my mother, father for their continuous, unconditional love and support, and for giving me the inspiration and motivation.

My last, and most heartfelt, acknowledgment must go to my daughter Maria. I stopped counting the times that wanted to give her more attention and couldn't because I was working on this thesis.

Abstract

Sign languages are used worldwide by a multitude of individuals. They are mostly used by the deaf communities and their teachers, or people associated with them by ties of friendship or family. Speakers are a minority of citizens, often segregated, and over the years not much attention has been given to this form of communication, even by the scientific community. In fact, in Computer Science there is some, but limited, research and development in this area. In the particular case of sign Portuguese Sign Language-PSL that fact is more evident and, to our knowledge there isn't yet an efficient system to perform the automatic recognition of PSL signs. With the advent and wide spreading of devices such as depth sensors, there are new possibilities to address this problem.

In this thesis, we have specified, developed, tested and preliminary evaluated, solutions that we think will bring valuable contributions to the problem of Automatic Gesture Recognition, applied to Sign Languages, such as the case of Portuguese Sign Language.

In the context of this work, Computer Vision techniques were adapted to the case of Depth Sensors. A proper gesture taxonomy for this problem was proposed, and techniques for feature extraction, representation, storing and classification were presented. Two novel algorithms to solve the problem of real-time recognition of isolated static poses were specified, developed, tested and evaluated. Two other algorithms for isolated dynamic movements for gesture recognition (one of them novel), have been also specified, developed, tested and evaluated. Analyzed results compare well with the literature.

Keywords:

Sign Language Recognition, Computer Vision, Depth Sensor, Machine Learning, Template Matching, Automatic Gesture Recognition, Hidden Markov Models

Resumo

As Línguas Gestuais são utilizadas em todo o Mundo por uma imensidão de indivíduos. Trata-se na sua grande maioria de surdos e/ou mudos, ou pessoas a eles associados por laços familiares de amizade ou professores de Língua Gestual. Tratando-se de uma minoria, muitas vezes segregada, não tem vindo a ser dada ao longo dos anos pela comunidade científica, a devida atenção a esta forma de comunicação.

Na área das Ciências da Computação existem alguns, mas poucos trabalhos de investigação e desenvolvimento. No caso particular da Língua Gestual Portuguesa - LGP esse facto é ainda mais evidente não sendo nosso conhecimento a existência de um sistema eficaz e efetivo para fazer o reconhecimento automático de gestos da LGP.

Com o aparecimento ou massificação de dispositivos, tais como sensores de profundidade, surgem novas possibilidades para abordar este problema.

Nesta tese, foram especificadas, desenvolvidas, testadas e efectuada a avaliação preliminar de soluções que acreditamos que trarão valiosas contribuições para o problema do Reconhecimento Automático de Gestos, aplicado às Línguas Gestuais, como é o caso da Língua Gestual Portuguesa.

Foram adaptadas técnicas de Visão por Computador ao caso dos Sensores de Profundidade.

Foi proposta uma taxonomia adequada ao problema, e apresentadas técnicas para a extração, representação e armazenamento de características. Foram especificados, desenvolvidos, testados e avaliados dois algoritmos para resolver o problema do reconhecimento em tempo real de poses estáticas isoladas. Foram também especificados, desenvolvidos, testados e avaliados outros dois algoritmos para o Reconhecimento de Movimentos Dinâmicos Isolados de Gestos(um deles novo).Os resultados analisados são comparáveis à literatura.

***Palavras-chave**:*

Reconhecimento de Língua Gestual, Visão por Computador, Sensores de Profundidade, Aprendizagem Automática, Correspondência de Padrões, Reconhecimento Automático de Gestos, Modelos Ocultos de Markov.

Resumen

Las lenguas de Signos se utilizan en todo el Mundo por una multitud de personas. En su mayoría son personas sordas y/o mudas, o personas asociadas con ellos por vínculos de amistad o familiares y profesores de Lengua de Signos. Es una minoría de personas, a menudo segregadas, y no se ha dado en los últimos años por la comunidad científica, la atención debida a esta forma de comunicación.

En el área de Ciencias de la Computación hay alguna pero poca investigación y desarrollo. En el caso particular de la Lengua de Signos Portuguesa - LSP, no es de nuestro conocimiento la existencia de un sistema eficiente y eficaz para el reconocimiento automático.

Con la llegada en masa de dispositivos tales como Sensores de Profundidad, hay nuevas posibilidades para abordar el problema del Reconocimiento de Gestos.

En esta tesis se han especificado, desarrollado, probado y hecha una evaluación preliminar de soluciones, aplicada a las Lenguas de Signos como el caso de la Lengua de Signos Portuguesa - LSP.

Se han adaptado las técnicas de Visión por Ordenador para el caso de los Sensores de Profundidad.

Se propone una taxonomía apropiada para el problema y se presentan técnicas para la extracción, representación y el almacenamiento de características.

Se desarrollaran, probaran, compararan y analizan los resultados de dos nuevos algoritmos para resolver el problema del Reconocimiento Aislado y Estático de Posturas.

Otros dos algoritmos (uno de ellos nuevo) fueran también desarrollados, probados, comparados y analizados los resultados, para el Reconocimiento de Movimientos Dinámicos Aislados de los Gestos.

Palabras-clave:

Reconocimiento del Lenguaje de Signos, Visión por Computadora, Sensores de Profundidad, Aprendizaje Automático, Correspondencia de Plantillas, Reconocimiento Automático de Gestos, Modelos Ocultos de Markov

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

[General Mathematical symbols]

$\forall$	Universal qualification
$\in$	Set membership symbol "*is an element of*"
$\wedge$	Logical conjunction "*and*"
$<$	Strict inequality symbol "is less than"
:	Such that
$=$	Equality symbol "is equal to"
$\|C\|$	Cardinality symbol "size of", where C represents a set
$\sum_{k=1}^{S}$	Summation over k from1 to S
$\sqrt{\ }$	Square root
$(n)^2$	Square of n
$\nexists$	Existential quantification "not exists"
$\|\|.\|\|$	Norm of vector

[Section 4.1.1]

A	Intrinsic camera matrix, or matrix of intrinsic parameters
$[R\|t]$	Extrinsic Camera Matrix, or matrix of extrinsic parameters
(X, Y, Z)	3D coordinates of a point in world coordinate space
(u, v)	Coordinates of the projection point in pixels
(c_x, c_y)	Principal point (that is usually at the image center)
(f_x, f_y)	Focal lengths expressed in pixel-related units
k_1, k_2, k_3	Radial distortion coefficients
p_1, p_2	Tangential distortion coefficients
(x_d, y_d)	Pixels of the depth camera
(x_{3D}, y_{3D}, z_{3D})	3D projection of depth camera
f_{xd}, f_{yd}	Intrinsic parameters of the depth camera (focal lengths)
c_{xd}, c_{yd}	Intrinsic parameters of the depth camera (principal point)
R	Rotation parameters estimated from stereo calibration.

T	Translation parameters estimated from stereo calibration.
[Section 4.1.2]	
$I_1, I_2, \dots, I_n$	Sequence of 640x480 depth matrixes(instants 1,2 , ... n)
IA	Time variant depth map matrix
m_i	Instant i
ia_i	Depth map matrix IA on instant i
p(i,j)	Depth values from I_m
MN	8-bit mask of the values l_i from I_n
[General Section 4.2]	
L	Set of *PostureLists*
P	Posture in real time.
[Section 4.2.2.1]	
P	Source posture
T	Target posture
D	Set of dexels from posture one posture
$MIny$	Min y coordinate
$MAxx$	Max x coordinate
$MAyy$	Max y coordinate
$MIxx$	Min x coordinate
(x_i, y_i , z_i)	A dexel
w	Width of a posture
h	Height of a posture
(s, t)	Limit of target
[Section 4.2.2.2]	
D_j	Set of *dexels* in any posture P_j
L	Set of *PostureLists*

A,B,C,D	Sides: top, right, down, left respectively
dA_j	Contour of side A
dB_j	Contour of side B
dC_j	Contour of side C
dD_j	Contour of side D
(x_i, y_i, z_i)	Point from D_j
(y_i, x_k)	Point from contour
FA,FB,FC,FD	Contour functions
S	Size of contours
DA,DB,DC,DD	Distance functions
D	Sum of distances
POLC	Represents the most likely gesture for Pj
[Section 4.2.2.3]	
PL	PostureList
$CAD(P, PL)$	*Average Distance* from P to PL
nPL	Number of *postures* in PL
(x_P, y_P, z_P)	Dexel from P
A	Set of points from P where $x = x_P = x_{PL}$, $y = y_P = y_{PL}$
$CXIO(P, PL)$	Number of points out of PL
$CXOI(P, PL)$	Number of points out of P
B	Points out of PL
C	Points out of P

[Section 4.3.1]

T	Instant in time
$p(T)$	Feature vector on instant T
$x_l(T), y_l(T), z_l(T)$	Feature left hand component functions
$x_r(T), y_r(T), z_r(T)$	Feature right hand component functions
$\mathbb{R}^6$	6-dimensional Euclidean Space

[Section 4.3.2.1]

h	Movement threshold
d	Movement time delay
T, L, R, H	Time, left hand, right hand and head *application stacks*
n	Number of elements on each of the four *stacks*
$rs(d)$	Speed of right hand for last d counts
$ls(d)$	Speed of left hand for last d counts
$\boldsymbol{M}(\ \ )$	A movement

[Section 4.3.2.2]

L	A set of movements from the same class(same gesture)
C	A set of elements from L
M	A movement from L
d	Number of elements of M
$u(t)$	Feature vector
V	Real time movement
$\mathrm{d}(\mathrm{V}, \mathrm{M})$	Distance from V to M
$D(V, L)$	Distance from V to L

[Section 4.3.2.3.1]

N	The number of states of the model
T	length of the observations sequence

$\mathrm{Q} = \{q_1, q_2, \dots q_M\}$	Set of states
$A = \{a_{ij}\}$	State transition probability distribution.
$p\{q_{t+1}\|q_t\}$	Transition probabilities
q_t	Current state
M	number of observation symbols in the alphabet
$\mathrm{V} = \{v_1, v_2, \dots v_M\}$	A discrete set of possible symbol observations.
$B = \{b_j(k)\}$	Observation symbol probability distribution in state j.
v_k	k^{th} observation symbol in the alphabet
o_t	Current parameter vector.
ℵ	Gaussian distribution
c_{jm}	Weighting coefficients
μ_{jm}	Mean vectors
Σ_{jm}	Covariance matrices
$\pi = \{\pi_i\}$	The initial state distribution
λ	An HMM
$O = \{o_1, o_2, \dots, o_T\}$	Observation sequence
$P\{O\|\lambda\}$	Probability that O was generated by the model
$\alpha_t(i)$	Forward variable

[Abbreviations]

API	Application programming interface
ANMM	Average Neighborhood Margin Maximization
BSN	Body Sensor Network
BW	Baum-Welch algorithm
CCD	Charge-coupled devices
CMOS	Complementary metal–oxide–semiconductor
CPI	Camera Pose Initialization
CV	Computer Vision
DARPA	Defense Advanced Research Projects Agency
EMG	Electromyogram
FPS	Frames per second
HCI	Human-Computer Interaction
HMM	Hidden Markov Model
HTK	Hidden Markov models toolkit
IDE	Integrated development environment
IR	Infrared
LED	Light-emitting diode
LIDAR	Light detection and ranging
LR	Left-Right HMM topology
LRB	Left-Right Banded model
NUI	Natural User Interface
PUI	Perceptual and natural user interface
RGB	Color model
PCA	Principal Component Analysis
PSL	Portuguese Sign Language
SDK	Software Development Kit
SGONG	Self-Growing and Self-Organized Neural Gas
ToF	Time-Of-Flight
VGA	Video Graphics Array
WIMP	Window, Icon, Menu, Pointing device

XML	Extensible Markup Language

Knowledge becomes evil if the aim

Plato

1. Introduction

Gestures are part of our everyday life. As we interact with the world, we use gestures to communicate and to deal with physical artifacts around us. Human motion, as a mean to provide input to a computer or output to the user is an increasing research area in Human-Computer Interaction and in Natural User Interfaces. Gesture Recognition systems, in particular, allow a user to perform hand arm and body movements or other motions, like facial expressions, head or eye movements, to interact and communicate with computer systems.

Unfortunately the signals emitted by humans, such as the sound associated with speech or gestures and facial expressions or other movements, associated with communication, are not easy to perceive by computers, and almost always require high throughput. Recent technologies such as *range sensors* [1, p. 179 Ch10] brought better ways to obtain and process those signals and therefore give developers new tools to make more natural communication with machines.

To create a gesture recognition application, for example, a developer must build components to interact with sensors[1], provide mechanisms to save and parse that data[2], build a system capable of interpreting the sensor data as gestures[3] and finally, interpret and utilize the results.

[1] Usually called drivers that make the connection between software and hardware
[2] Depth Sensor Signal Pre-Processing (5.1.)
[3] Gesture Representation(5.2.1, 5.3.1)

1.1 Motivation

In the literature, there is no known effective and permanent solution to the automatic recognition of Portuguese Sign Language (PSL), using computer vision, other than preliminary approaches to automatic recognition of alphabet spelled PSL [38]. It is a complex problem to be solved that requires intense research and development.

With the new emerging technologies, such as depth sensors, now at our disposal given some recent economies of scale, new layers of information and probably more effective tools than those studied and used until quite recently, are now at our hands.

Thus we find that a new approach using these new depth sensor devices, along with concepts from machine learning, stochastic processes and computer vision, can bring promising contributions to the state of the art, in the area of automatic gesture recognition applied to sign languages, including PSL.

New horizons can be open in both the translation of PSL sign language to Portuguese written text, and other forms of textual representation of PSL, yet to appear, as well as systems for teaching or learning sign language.

Although the main motivation for this work is the provision of technology to improve the communication of individuals, deprived of the ability to listen and/or speaking, namely, speakers of Portuguese Sign Language, we aim at developing new foundational techniques and tools for this initial defined purpose, but also to enable the improvement of natural communication via gesture, between individuals and machines. Thus we find that a new approach using these new depth sensing devices, applied to machine learning, stochastic processes and vision, can be an important contribution to the state of the art in this area.

1.2 Problem description

In Sign Language, one of the most difficult challenges is to turn the sensed and acquired gesture raw data into something meaningful, for example, in the context of a gesture control system for an application. The sequences of raw, static, or moving events that comprise a gesture or a sign, must be understood by the application. Most likely, the programmer would use his/her domain knowledge to develop a (complex) set of rules and heuristics to classify those sequences. For a gesture recognition system, this set of rules

could easily become increasing complex and unmanageable. A better solution would be to use some machine learning techniques to classify the gestures.

The problem that this thesis addresses can be stated as follows: "How to perform booth static and moving automatic hand gesture recognition, in the context of Portuguese Sign Language ?".

In this thesis, a set of solutions are presented and tested to tackle the above mentioned problem, which is modeled for basic and simple problems of PSL. A system architecture and a demonstrating application were developed , enabling signal acquisition via a depth sensor and feature processing, from a person issuing two handed static and moving gestures, and then enforces automatic gesture recognition. This application has been tested with success for the particular case of Portuguese Sign Language (PSL), addressing the automatic recognition of signs that spell all letters of the alphabet, as well as PSL signs that correspond to a limited set of Portuguese words.

1.3 Thesis

1.3.1. Thesis hypothesis

- **Portuguese Sign Language Adoption**: In our first hypothesis, we state that it is possible to extend and adapt state-of-the-at work in automatic gesture recognition, previously done for other Sign Languages, to limited and simple, yet fundamental problems of the specific case of PSL.
- **Gesture Modeling and Recognition in 3D:** A second hypothesis is that for the case of limited and simple classes of gestures that are part of Portuguese Sign Language, these can be modeled and automatically recognized in 3D.
- **Depth Sensor Application**: Our third hypothesis is that a Depth Sensor can be exclusively used to perform and improve automatic gesture recognition in specific cases, such as isolated static gesture and isolated moving gesture.

1.3.2. Objectives

To address the above mentioned hypothesis, we have stated the following thesis objectives, per each Hypothesis:

- **Hypothesis 1 - Portuguese Sign Language Adoption** - The description of Portuguese Sign Language has its specific meanings and symbols, which differs from others. In this sense, it is important to verify if the work and results reported in the literature, regarding other Sign Languages, are also valid and possible to achieve in the case of PSL. We will show that with our research, we are pairing with our peers in the literature, for simple and limited problems of Automatic Sign Language Recognition, with a specific application to PSL.
- **Hypothesis 2 – Gesture Modeling and Recognition in 3D** – Our objective is to specify, develop and test a system architecture that uses fully 3D data structures to define describe, record and classify the depth data and use that data in an efficient and effective manner, for real-time automatic gesture recognition.
- **Hypothesis 3 - Depth Sensor Application** – Differently from traditional Computer Vision-based approaches, that require one or more video cameras, our objective is to use only depth information from a single sensor to address some of the simple, yet fundamental, problems of gesture recognition in real-time. It is our intention to build a solution entirely based on depth information, able to recognize the basic classes of gesture, in real-time.

1.3.3 Scientific Contribution

This dissertation presents the following contributions:

- Identifies proper gesture taxonomy and presents a structure for static[4] and moving[5] gesture with appropriate feature representation and information logging.
- Introduces novel techniques and algorithms[5,6] for 3D depth feature extraction and classification and Sign Language recognition with little learning effort.
- Presents good experimental results in real-time that compare very well with the literature, which improve the Sign Language recognition process and, in particular, the PSL recognition process.

[4] See 5.2 Isolated Static Gestures
[5] See 5.3. Isolated Moving Gestures and Hand Kinematics

1.4. Document structure

The remaining portions of this document are structured as follows:

Chapter 2: In this chapter we will present an overview of introductory concepts that we found as being foundational for a good understanding of the context of the research presented in this thesis. Firstly, we will present the concept of "gesture" according to literature. Next we will define the role of gestures in human communication, in particular Sign Languages. Then we will propose a gesture taxonomy that seems to be most appropriate for PSL Finally, we will provide a description of Gesture Recognition, namely using Range Sensors that include Depth Sensors.

Chapter 3: This chapter lists some of the critical related works, taken from the state-of-the-art, regarding static, moving, isolated and continuous gesture recognition.

Chapter 4: In this chapter, we will introduce the details of our proposed system architecture, its design and implementation, and will describe the components of the developed applications.

Chapter 5: In this chapter we will explain the overall development methodology used to address the problems. Firstly, we will give some theoretical concepts on *Signal Processing* in Depth Sensors. The following subsection will present the work done with *Isolated Static Gestures* modeling. We will present the mathematical formalization of two novel algorithms as well as setup steps and final results. The third subsection covers the work done with *Isolated Moving Gestures*. We will present the mathematical formalization of two proposed algorithms (one of them novel), to tackle the problem, as well as setup steps and a theoretical definition of the Hidden Markov Model solution. Finally, the experimental results are presented.

Chapter 6: This chapter presents the results and discussion about the algorithms for both *Isolated Static Gestures* and *Isolated Moving Gestures* presented in the previous chapter.

Chapter 7: This chapter has the final conclusions and considerations about hypothesis coverage, goals and results achieved, and recommendations for future Work.

Appendix A: Presents the theoretical introduction of Hidden Markov Models and the three canonical problems on which we have based one of the algorithms for *Isolated Moving Gestures* Recognition.

1.5 Summary

In Chapter 1 the motivation and the overall contextualization of the addressed problems have been defined, along with the thesis hypotheses and respective objectives to be achieved with this thesis work.

If knowledge can create problems,
it is not through ignorance that we
can solve them.

Isaac Asimov

2. Background Information

In this chapter we present some fundamental concepts for understanding the scope of this thesis. From the literature review, we defined the concept of "Gesture" in the context of "*Sign Language Recognition*". Then we contextualized "*Sign Language*" in the framework of "*Gestures in Human Communication*". We also define what we mean by "*Gesture Recognition*" and we propose an appropriate *taxonomy* to classify and define the subject of our work – Gesture in the context of Sign Language Recognition. We then define the role of "*Range Sensing*", in particular *Depth Sensors* and, more precisely, our adoption of *Microsoft Kinect* for *Gesture Recognition.* Finally we detail the scope of the thesis according to the presented concepts.

2.1 Gesture

According to the Oxford English dictionary a gesture is "*A movement of part of the body, especially a hand or the head, to express an idea or meaning*" [2]. This definition expresses in some way the main challenges of computer processing of gestures. The first challenge is defining what exactly a gesture is. To capture record and recognize a gesture, we need to know what signals and what "parts of the body" exactly we want to capture. Without this, we can't have a computational digital representation of those gestures. The second challenge respects to "express an idea or a meaning". This second challenge is truly a complex problem. We should not only computationally represent these "ideas and meanings" but also, to make them match the gestures captured in the first challenge. This is the problem of *recognition and classification*, which usually requires mathematical methods, statistical and machine learning sciences.

2.1.1 Gesture in Human Communication

As we can observe in everyday life, speech and gesture seem to be one of the most natural and common ways human being use to communicate with each other. However, the vast majority of current human-computer interfaces still follow the WIMP metaphor, which force the user to adapt to the machine requirements. Mainstream Human-Computer Interaction (HCI) devices, like keyboards or mice are non-natural communication for us humans and many researchers have dedicated efforts for years to create more natural human-machine interfaces.

Speech recognition is being widely researched for decades. Its fundamentals and theoretical background is well studied and many commercial products have been developed [6]. However, gesture recognition research is more recent and therefore remains a more unexplored field. This lack of research is due mainly to the very high computational load needed to process efficiently video signals [6]. Nowadays powerful enough processors are able to process video signals in real time and/or handle with the huge amount of information necessary to store these kinds of signals. Additionally, with the advance of computer hardware devices such as Depth Sensors, more sophisticated applications can be though and more natural interfaces can be built. Nowadays, Virtual Reality, Remote Operation, Robot Command and Sign Language Recognition, are the applications which could take more benefit of the development of gesture recognition techniques [6].

2.1.2 Sign Language

A Sign Language is a natural language which, instead of speech, uses gesture to convey meaning, combining hand and finger movements, including hand-shapes, their position, orientation and flexion and also, arms and body movements as well as facial expressions and lip-patterns. Sign languages are usually developed in deaf communities, which include, deaf or hearing-impaired as well as their families, friends and interpreters.

Contrary to popular belief, sign language has not an international standard and is also not completely delimited by country boundaries. Wherever communities of deaf people exist, sign languages have naturally developed and evolved. As with spoken languages, sign languages vary from region to region. They are not totally based on the spoken language in the country of origin. Alike spoken languages, there are always some regional

diversities and dialects and linguistic evolution phenomena, even in small communities of countries like Portugal. Portuguese Sign Language – PSL, has no standards yet developed but a descriptive set of gestures that characterizes the most common used can be found in [10]. Other simple forms of signed communication have been developed in situations where speech is not practical to speak, such as between scuba divers, in loud workplaces, while hunting or in television recording studios.

Sign language is a visual language and consists of 3 major components:

- *finger-spelling*: used to spell words letter by letter of the alphabet.
- *word level sign vocabulary*: used for the majority of communication.
- *non-manual features*: facial expressions and tongue, mouth and body position.

2.1.3 Gesture Recognition

Gesture recognition is the mathematical interpretation of a human motion by a computing device. Gesture recognition, along with facial expression recognition, speech recognition, eye tracking and lip movement recognition, are components of what the community refers to as a Perceptual and Natural User Interface (PUI) [11]. The goal of PUI is to enhance the efficiency and ease of use for the underlying logical design of a given HCI technique, a design discipline of human-computer interaction, known as usability. In personal computing, gestures are most often used for input commands, but gesture output via 3D virtual avatars mimicking sign language signs, is also available in the scientific community [12, 13].

Recognizing gesture an HCI input modality, allows computers to be more accessible for the physically-impaired and to citizens in general and makes interaction more natural in a gaming or 3D virtual world environment. Hand and body gestures can be sensed by a device that contains accelerometers and gyroscopes to detect tilting, rotation and acceleration of movement -- or the computing device can be outfitted with a camera (video or depth camera), so that software logic in the device can recognize and interpret specific gestures. A wave of the hand, for instance, might be recognized to terminate a program.

We defined a moving gesture as " a sequence of postures connected by motions over a short time span." We are talking about voluntary motions of hands and arms to express some action or meaning.

2.1.4 Proposed Hand Gesture Taxonomy

In literature we can find several classifications associating gesture to its function and meaning [3]. Gestures can be *independent* or *complementary to speech*, *conscious* or *unconscious* [4, pp. 383-409 Ch.15].

Based on referenced works from the literature, namely [5] and [7], we propose an adapted taxonomy that we find to be the most suitable for automatic hand gesture recognition. In this context, we define the following three basic concepts associated to hand gesture:

Movement – the atomic element of motion.

Activity – a sequence of ***Movements***.

Action – a high-level description of what is happening in the context of hand gesture ***Activity*** [7].

To characterize ***Movement,*** we further classify hand gesture as:

- ***Static Gesture*** - defined by a single hand posture.

In this thesis, *static hand gestures* will be referred to as *hand postures*, adopting the following ***posture*** definition:

"*Posture is a specific combination of hand position, orientation and flexion observed at some time instance*" [5].

Postures are signals whose time variance can be neglected, so that they can be completely analyzed using only a small set of frames of the discretized hand posture signal, in a given time period. Other examples of postures, are facial expressions like a smile or an angry face and the hand postures for finger-spelling used to spell alphabet letters in Sign Languages, in which a limited set of frames should be almost enough for complete understanding.

- ***Moving Gesture*** - defined by a time sequence of *postures* and positions.

We adopt the description of a moving hand gesture from [5], as follows:

"*A sequence of postures connected by motions over a short time span.*"

A moving gesture can be thought as a sequence of varying postures. In an input video or depth camera signal, the individual frames define the postures and the video or depth sequence defines the gesture. The word level signs like "cat", "dog" or "hello" in Sign Languages, good examples of moving gestures [6], are gestures that can only be recognized taking the temporal context information into account.

In what concerns ***Activity***, we further classify hand gestures as:

- ***Isolated Gesture***

Isolated sign recognition deals with the recognition of signs that are performed alone, without any signs before or after the performed sign. Thus, the sign is performed without being affected by the preceding or succeeding signs. The importance of the research on isolated sign recognition is that it enables the finding of better mathematical models and features that represent the performed sign.

- ***Continuous Gesture***

Recognizing unconstrained continuous sign sentences is another challenging problem in Sign Language Recognition. During continuous signing, individual signs can be affected by the preceding or succeeding signs. This effect is similar to co-articulation in speech. Additional movements or shapes may occur during the transition between signs. This movement phenomenon is called movement epenthesis [8]. This effect complicates the explicit or implicit segmentation of the signs during continuous signing. To solve this problem, the movements during transitions can be modeled explicitly and used as a transition model between the sign models [9].

In *Figure 1* we depict a hierarchical view of our proposed hand gesture taxonomy.

When gestural *activity*, gains significance in some Sign Language context, it turns into a set of ***Actions***. An ***Action*** always implies that the Signer conveys something related to the meanings of Gestures in Sign Language. A proper *syntax* and *grammar* give a meaning to the gesture *activity*, and it becomes something meaningful as a sentence or any message in terms of human communication[6].

[6] See 2.1.1 Gesture in Human Communication and 2.1.2 Sign Language

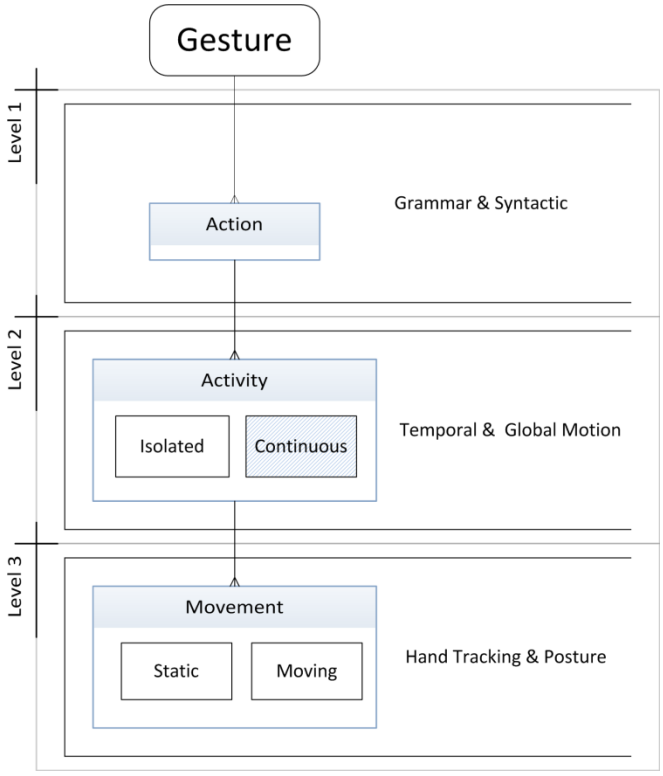

*Figure 1 – A hierarchical view of **o**ur proposed Hand Gesture taxonomy*

The figure above illustrates the gesture classification described in the preceding paragraphs. Hand gestures, as used in Sign Language production, are composed of a set of ***Actions*** related to its own syntax, grammar, semantics and pragmatics, of human sign communication. A particular ***Action*** is defined by sets of ***isolated* or *continuous*** gestures over time, which composes the ***Activity*** in terms of Sign Language. In turn, an ***Activity*** is composed of one or more ***static*** or ***moving*** hand gestures that characterize the concept of ***Movement*** in our proposed taxonomy.

2.2 Range Sensing

A first approach to calculating range is based on measuring the time-of-flight of an emitted signal [14]. Time-of-flight (ToF) technology, based on measuring the time that light emitted by an illumination unit requires to travel to an object and back to a detector, is used in light detection and ranging (LIDAR) [15] scanners for high-precision distance measurements. Recently, this principle has been the basis for the development of new range-sensing devices (***Figure 2***), so called *ToF cameras*, which are realized in standard CMOS or CCD technology [16].

Depth Sensors calculate the distance to a given object, and are also useful in motion and gesture driven interfaces. Many range and depth sensors calculate the position of the nearest object, using stereoscopic triangulation computed using epipolar geometry. For example, the Sharp IR Ranger [17] emits a controlled burst of near infrared light from a light emitting diode (LED). This light is reflected by any object within a few feet and is focused onto a small linear charge-coupled devices (CCD) array that is displaced slightly from the emitter. The position of the reflection on the sensor can be related the distance to the object by epipolar geometry. Similar approaches can be used over a longer effective distance with the use of lasers rather than LEDs [1, p. 179].

Stereo computer vision systems similarly use triangulation to calculate depth. If the same object is detected in two displaced views, the difference in their sensed 2D positions, called "disparity," can be related to the depth of the object [18, 19]. Stereo vision techniques may be used to determine the depth of a discrete object in the scene, or to compute depth at each point in the image to arrive at a full range image.

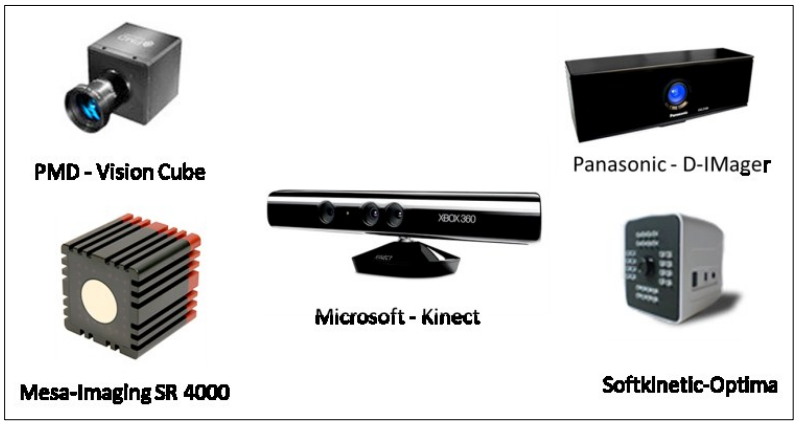

Figure 2 - Devices that use some range sensing technology

In this work we have used Microsoft® XBOX 360 Kinect Sensor that has an optical system sending information streams (video, depth and audio) through a USB connection (***Figure 3***), based on a 3D sensor by PrimeSense [20]. Kinect reconstructs 3D scene information from a continuously-projected infrared structured light (***Figure 4***).

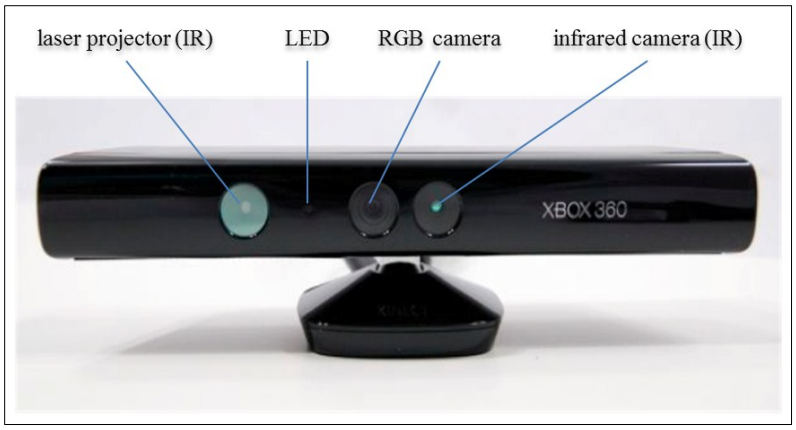

Figure 3- Kinect Sensor - From left to right, a laser projector (IR), an RGB camera and an infrared camera (IR).Among the RGB camera and the laser there is an IR LED just for status indication and has no role in the optical system.

It implements a natural user interface (NUI) and provides two separate video streams - one VGA 640x480, 30FPS originated by RGB camera, and another 11-bit 640x480, 30FPS output corresponding to the 3D depth after processing. The drivers can get 1280x1024streams although limited to 10 fps.

For image processing, the sensor includes an IR Class-1 Laser projector (harmless to the human eye), a RGB camera and an IR camera. The IR laser projects a known (hardcoded) image made of static pseudorandom diffraction patterns (a computer-generated hologram), on the scene. A standard off-the shelf monochrome CMOS IR sensor horizontally aligned with the IR projector (incapable of extracting time of return from modulated light), senses the IR spots in the scene environment (***Figure 4***). Through stereoscopic triangulation [21] , between corresponding points of the hardcoded image and of the IR camera image (which are easily calculated, since corresponding points lie in the same horizontal scan lines), the system is able to calculate their disparity, that is, the horizontal offset of the point on the IR camera image relative to corresponding hardcoded position and, from that calculation, reconstruct the 3D position of each scene point, which is inversely proportional to the found disparity.

Figure 4 - The projection of a computer-generated hologram on the environment by Kinect.

The Kinect sensor outputs video at operational a frame rate of 30 Hz (although the sensor is capable of 60 Hz) and the available image resolutions are respectively, 8-bit VGA resolution (640 × 480 pixels) for the RGB camera and the same VGA resolution for the IR camera, but with 11-bit depth, which provides 2,048 levels of depth resolution[7]. The area required to play Kinect is roughly 6m² and the sensor can maintain tracking through an extended range of approximately 0.8–3.5 m. This range is automatically adjustable, based on gameplay and the player's physical environment, such as the presence of obstacles. The sensor has an angular field of view of 57° horizontally and 43° vertically and a depth resolution of around 1.3 mm per pixel at the minimum range and 1 cm at 2 m. Together with the Kinect sensor's microphone array, featuring four microphones which captures 16-bit audio at a sampling rate of 16 kHz and enabling acoustic source localization and ambient noise suppression, this sensor allows full-body 3D motion capture, facial recognition and robust voice recognition, thus implementing in fact a paradigm of a natural user interface (NUI). The Kinect is capable of simultaneously tracking up to six people, including two active players for motion analysis with a feature extraction of 20 joints per player.

2.3 Detailed Scope of the Thesis

The simple, yet fundamental problems studied in this thesis include only *isolated static* and *isolated moving hand gestures* for Sign Languages, with a particular focus in Portuguese Sign Language. Therefore, the thesis only covers isolated gestures that have a

[7] The drivers can get 1280x1024streams although limited to 10 fps.

meaning by themselves and do not depend on any previous or subsequent static or moving gestures.

We hope, in future approaches, to be covering the remaining classes: *continuous static* and *continuous moving hand* gesture.

2.4 Conclusions

In this chapter we have started to present some fundamental concepts for the reader to understand the scope of this work. We have then provided the literature view about the concept of Gesture in the context of Sign Language Recognition and in Human Communication. As a result, we have proposed an appropriate taxonomy to classify Hand Gesture in the context of Sign Language Recognition. Finally, we have defined the role of Range Sensing, in particular Depth Sensors, like the adopted Microsoft Kinect, in the context of Gesture Recognition.

Knowledge has to be improved, challenged, and increased constantly, or it vanishes.

Peter F. Drucker

3. State of the Art Review

In literature there are several approaches and methods for capturing, representing and recognizing gestures, other than by using Depth Sensors [22, 23], Different hardware and devices, methods of detection, segmentation and tracking, feature extraction and representation and also several recognition methods algorithms and techniques are available.

In this chapter we present an overview of related works that make the state of the art regarding Gesture and Sign Language recognition.

This section is divided in four different stages on which our problem is also subdivided. Eventually some woks may cover the subjects from more than one of these subdivisions and we refer to it in the appropriate section as "State of the Art" on that subject.

First, in subsection 3.1, we have the "State of the Art" on hardware and devices used by researchers over Gesture and Sign Language Recognition (see 4.1)

The second subsection 3.2, is about works related to static hand pose recognition and finger-spelling (see 4.2).

In third subsection 3.3 we present the works related to Isolated Moving hand gesture (see 4.3).

Finally in subsection 3.4, we will refer works on continuous gesture recognition that we will not cover in this thesis as earlier written in this document, but we found it as important to formulate the architecture in section 5. and propose further work.

3.1 Sensors and Devices

Hand Gesture recognition has been used in several studies as a mean of human-computer interaction (HCI) or for robot control, as an alternative to the most common HCI devices such as mouse, keyboard or joystick.

There are several works using all kinds of devices for gesture data acquisition, and processing. Among them we highlight the following:

A) Computer Vision systems, based on color image and video

"Non-rigid Shape Recognition for Sign Language Understanding" [24]. In this work Vladutu L. et al. present their method designed to identify human gestures for Sign Language recognition on a standard personal computer connected to a color video camera. The recognition is based on Video-stream analysis of gestures using *Principal Component Analysis* on *Singular Value Decomposition* to extract the representative frames using *Fuzzy-Logic*. Feature extraction is based on *Color Layout Descriptors* and the *Region shape*.

Jyh-Ming Lien et al in their *work "**Skeleton-Based Data Compression for Multi-Camera Tele-immersion System**"* [25] present a 3D stereo multi-camera video Image-based system where they make full body 3D reconstruction for tele-immersive applications. The system generates large amount of data points, which have to be sent through the network in real-time. They introduce a skeleton-based compression method using motion estimation where kinematic parameters of the human body are extracted from the point cloud data in each frame.

B) Bio signals and body sensor based devices

Kim, Mastnik and André presented ***"EMG-based Hand Gesture Recognition for Realtime Biosignal Interfacing"*** [26]. They developed an electromyogram (EMG) based interface for hand gesture recognition using a single channel EMG sensor positioned on the inside of the forearm. In addition to common statistical features such as variance, mean value, and standard deviation, they also calculated features from the time and frequency domain including Fourier variance, region length, zero crosses, occurrences, etc. For realizing real-time classification, they combined two simple linear classifiers (k-NN and Bayes) in decision level fusion.

"Hand Gesture Recognition with Body Sensor Networks" [27], was a work of King, Lo, Darzi, and Yang , where they present a pervasive skills assessment tool based on the *Body Sensor Network* (BSN). By integrating an optical sensor with the BSN node, it provides a wireless gesture sensing platform for assessing basic laparoscopic skills.

.

C) Wearable devices

"Using Multiple Sensors for Mobile Sign Language Recognition" [28] from Brashear et al. presented a lab-based *Sign Language Recognition* system with the goal of making it a mobile assistive technology. They examine using multiple sensors for disambiguation of noisy data to improve recognition accuracy. Their experiment compares the results of training a small gesture vocabulary using noisy vision data, accelerometer data and both data sets combined.

Westeyn, Brashear, Atrash, and Starner on their work "***Georgia Tech Gesture Toolkit: Supporting Experiments in Gesture Recognition***" [29] presented the *Georgia Tech Gesture Toolkit GT2k*, a Gesture Recognition tool using a *hat mounted camera* with the image defocused to improve tracking. They use Cambridge University's Hidden Markov Models Toolkit (HTK), to provide tools that support gesture recognition and capabilities for training models.

Zhu and Sheng from Oklahoma State University on their work "***Online Hand Gesture Recognition Using Neural Network Based Segmentation*** [30]," ,implemented algorithms using an inertial sensor worn on a finger of the human subject. They propose an online hand gesture recognition algorithm for a robot assisted living system. A neural network-based gesture spotting method is combined with Hidden Markov Model (HMM) to recognize hand gestures. For segmentation they used neural network and in the recognition module, Bayesian filtering was applied to update the results considering the context constraints.

McGuire et al. worked on "***Towards a One-Way American Sign Language Translator*** [31] ", inspired by the *Defense Advanced Research Projects Agency's* (DARPA) in speech recognition, and introduced a new task for Sign Language Recognition: a mobile

one-way american Sign Language translator using a one-handed glove-based system and Hidden Markov Models (HMM).

"***A Simple Wearable Hand Gesture Recognition Device using iMEMS***" [32], was a work from Pandit, Dand, Mehta, Sabesan and Daftery, where they proposed a "Human Computer Interfacing Device" mapping Hand Gestures to communicate with computers, robots and other household electronic devices. An accelerometer is utilized to get moving/static profile of movement to navigate the mouse or gyroscope to rotate 3-D virtual objects. Accelerometer profiles are converted into wireless interactivity. The device involves non-tactile interaction with computer screen where mouse pointer reacts to and manipulates screen content in accordance with hand gestures.

3.2 Static Hand Posture Recognition and Finger-spelling

Some works have been done regarding to static configuration of the human hand which is called hand posture and its applications for human-computer interaction. 3D approaches are our main focus but there is still some lack of work in this area for *Static Posture Recognition*.

The researchers have been using a variety of procedures and techniques acquiring these configurations and behaviors, among which the following:

A) **Appearance-based approaches**

"Appearance Based Recognition of American Sign Language Using Gesture Segmentation" [33] ,a work by Kulkarni and Lokhande where they propose a system for automatic translation of static gestures of alphabets in American Sign Language. It uses three feature extraction methods and a neural network to recognize signs. The system deals with images of bare hands, which allows the user to interact in a natural way. An image is processed and converted to a feature vector that will be compared with the feature vectors of a training set of signs.

"Haarlet-based Hand Gesture Recognition for 3D Interaction" [34] . Bergh et al. present a system to manipulate 3D objects or navigate through 3D models by detecting

hand gestures and movements of a user in front of a camera mounted on top of a screen. They introduce an improved skin color segmentation algorithm which combines an online and an offline model and a Haarlet-based hand gesture recognition system, where the Haarlets are trained based on *Average Neighborhood Margin Maximization* (ANMM).

"Hand gesture recognition using a neural network shape fitting technique" [35]. Stergiopoulou and Papamarkos present a method for hand gesture recognition based on a fitting procedure via a Self-Growing and Self-Organized Neural Gas (SGONG) network proposed. Initially, the region of the hand is detected by applying a color segmentation technique based on a skin color filtering procedure in the YCbCr color space. Then, the SGONG network is applied on the hand area so as to approach it's shape. Based on the output grid of neurons produced by the neural network, palm morphologic features are extracted. These features, in accordance with finger features, allow the identification of the raised fingers. Finally, the hand gesture recognition is accomplished through a likelihood-based classification technique.

B) **2D *Computer Vision* methods**

"***Hand gesture recognition for man-machine interaction***" from Kapuscinski and Wysocki, addresses some basic problems of hand gesture recognition. Three steps important in building gestural vision-based interfaces are discussed. The first step is the hand location through detection of skin colored regions. Representative methods based on 2D color histograms are described and compared. The next step is the hand shape (posture) recognition. An approach that uses the morphological hit-miss operation is presented. Finally, application of Hidden Markov Models (HMM) in recognition of *Moving Gestures* is explained. An experimental system that uses the discussed methods in real time is described and Recognition results are presented [36].

"Hand Gesture Recognition in Natural State Based on Rotation Invariance and OpenCV Realization" from Zhang, Yun, and Qiu propose a gesture recognition solution with color segmentation, density distribution feature and an improved algorithm based on

artificial markers in the hand. The hand gesture recognition method is based on rotation invariance. This method can realize the multi-angle or posture changing hand in gesture recognition [37] .

A hand gesture recognition engine presented by Dias, Nande,. Barata and Correia called ***"O.G.R.E. – Open Gestures Recognition Engine"*** [38], based on Computer Vision (CV), as a computing platform to support gesture interaction. This approach uses image processing techniques and a single video camera, to address the problem of generic hand gestures recognition, especially of alphabet spelled Portuguese Sign Language hand poses. The system initially removes the background of captured images, eliminating irrelevant pixel information and the human hand is then detected, segmented and its contours localized. From these contours significant metrics are derived, allowing a search in a pre-defined hand poses library of Portuguese Sign Language signs, where each pose is previously converted into a set of metric values.

C) **Other machine learning techniques**

"Hand Region Extraction and Gesture Recognition using entropy analysis" [39]. Shin et al., propose a gesture recognition system using motion information from extracted hand region in complex background image. First, they measure entropy for the difference image between continuous frames. Using color information that is similar to a skin color in candidate region which has high value, they extract hand region only from background image. Chain code has been applied to acquire outline detection from the extracted hand region and hand gestures recognition is carried out by improved centroidal profile.

Liu, Gan and Sun on their work "***Static Hand Gesture Recognition and Its Application based on Support Vector Machines***" [40], propose a novel hand gesture recognition algorithm to identify whether the hand can meet the requirements of driving license test or not. The algorithm is mainly based on Hu moments and *Support Vector Machines* (SVM). Firstly, Hu invariant moments are extracted into a vector of 7 dimensions. Then, a *Support Vector Machines* (SVM) is used to find a decision border between the

integrated hand and the defected hand. At testing time, a decision is taken using the previously learned SVMs.

"Skin Color Profile Capture for Scale and Rotation Invariant Hand Gesture Recognition" [41]. Bastos and Dias presented in 2007 a new approach to real-time scale and rotation invariant hand pose recognition , which was a technique for computing the best hand skin color segmentation map - "skin profile", used to enable correct classification of skin regions and construct reliable scale and rotation invariant hand pose gesture descriptors, by a technique, referred to as "oriented gesture descriptors", based on a scale, rotation and luminance invariant image feature descriptor (referred to as FIRST [42]), with real-time extraction and tracking characteristics.

D) **3D approaches**

Ueda et al. presented ***"Hand Pose Estimation for Vision-based Human Interface"*** [43] , where a novel method is proposed for hand pose estimation that can be used for vision-based human interfaces. The aim of the method is to estimate all joint angles to manipulate an object in the virtual space. In this method, the hand regions are extracted from multiple images obtained by the multi-viewpoint camera system. By integrating these multi-viewpoint silhouette images, a hand pose is reconstructed as a "voxel model". Then all joint angles are estimated using three dimensional model fitting between hand model and voxel model.

Eghbalet al., presented ***"Real Time Hand Based Robot Control Using Multimodal Images"*** [44] . A new solution using 2D/3D images for real time hand detection, tracking and classification which is used as an interface for sending the commands to an industrial robot. 2D/3D images, including low resolution range data and high resolution color information, are provided by a novel monocular hybrid vision system, called MultiCam, at video frame rates. After region extraction and applying some preprocessing techniques, the range data is segmented using an unsupervised clustering approach. The segmented range image is then mapped to the corresponding 2D color image. Haar-like

features are then extracted from the segmented color image and used as the input features for an AdaBoost classifier to find the region of the hand in the image and track it in each frame. The hand region found by AdaBoost is improved through post processing techniques and finally the hand posture is classified based on a heuristic method.

Kollorz, Penne, and Barke, presented "***Gesture recognition with a time-of-flight camera***" [23] with their new approach for gesture classification using x- and y-projections of the image and optional depth features. The system uses a 3-D time-of-flight (ToF) sensor which has the big advantage of simplifying hand segmentation. For the presented system, a Photonic-Mixer-Device (PMD) camera with a resolution of 160 × 120 pixels and a frame rate of 15 frames per second is used. The goal of their system was to recognise 12 different static hand gestures. They concluded that the x- and y-projections and the depth features of the captured image are good enough to use a simple nearest neighbour classifier, resulting in a fast classification.

3.3 Isolated Moving Hand Gesture and Sign Language Recognition

Moving hand gestures comprise a sequence of hand shapes that describe the hand trajectory, with spatial transformation parameters, such as rotation, translation, scaling, depth variations etc. [8].
We highlight some *"State of the Art"* works covering this subject:

"O.G.R.E. – Open Gestures Recognition Engine" [38], presents the recognition of hand trajectories - *Simple Paths* defined as sketched user defined simple primitive shapes, such as triangle, circle, square using a recursive algorithm that uses calligraphic interfaces based on fuzzy logic, to determine the real bounding box. This engine also Hybrid gestures called *Staged Paths*, composed of both static poses and hand trajectories. The recognition method is based on vector analysis and localized hand pose identification. The user hand describes a trajectory which consists of a limited point set (dependent on the video capture rate and the gesturing speed) representing segments of high curvature.

[8] "Hand Gesture Taxonomy" (2.1.4)

Ionescu *et al.* , present ***"Dynamic Hand Gesture Recognition Using the Skeleton of the Hand"*** [45]. They use user Computer Vision (CV) in the interpretation of human gestures proposing novel moving hand gesture recognition technique based on the 2D skeleton representation of the hand. For each gesture, the hand skeletons of each posture are superposed providing a single image which is the moving signature of the gesture. The recognition is performed by comparing this signature with the ones from a gesture alphabet, using Baddeley's distance as a measure of dissimilarities between model parameters.

"Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models" [46] . Elmezain, et al. propose a method to recognize the alphabets from a single hand motion using Hidden Markov Models (HMM). Gesture recognition is based on three main stages; preprocessing, feature extraction and classification. In preprocessing stage, color and depth information are used to detect both hands and face in connection with morphological operation. After the detection of the hand, the tracking will take place in further step in order to determine the motion trajectory so-called gesture path. The second stage, feature extraction enhances the gesture path which gives us a pure path and also determines the orientation between the center of gravity and each point in a pure path. Thereby, the orientation is quantized to give a discrete vector that used as input to HMM. In the final stage, the gesture of alphabets is recognized by using Left-Right Banded model (LRB) in conjunction with Baum-Welch algorithm (BW) for training the parameters of HMM. Therefore, the best path is obtained by Viterbi algorithm using a gesture database.

" Gesture Recognition In Flow based on PCA Analysis using Multiagent System" [47] . Billon, Nédélec and Tisseau put on a short play featuring a real actor and a virtual actor, who will communicate through movements and choreography with mutual synchronization. They worked on a system that can recognize key-gestures made by a real actor using a method for real-time recognition based on reducing movements from any motion capture system to a single artificial signature. They use properties from Principal Component Analysis (PCA) to generate it.

Dagostar and Sarrafzadeh in their work "***Gesture recognition through angle space*** " [48] , present a novel approach for moving hand gesture modeling using neural networks. The method is independent of the input device and can be applied as a general back-end processor for gesture recognition systems.

3D range sensor based works:

"**Gesture recognition using a range camera**" [49]. Holte and Moeslund use of range information acquired by a CSEM SwissRanger SR-2 camera for one and two arms gesture recognition. The range data enables motion detection and 3D representation of gestures. Motion is detected by double difference range images and filtered by a hysteresis bandpass filter. Gestures are represented by shape contexts in the form of spherical histograms. This representation allows a simple matching by binary comparison of the histogram bins.

" ***Hand gesture recognition with a novel IR time-of-flight range camera–a pilot study***" [50]. Breuer, Eckes and Muller present a gesture recognition system for recognizing hand movements in near real-time. The system uses a infra-red time-of-flight range camera with up to 30 Hz frame rate to measure 3d-surface points captured from the hand of the user. The measured data is transformed into a cloud of 3d-points after depth keying and suppression of camera noise by median filtering. Principle component analysis (PCA) is used to obtain a first crude estimate on the location and orientation of the hand. An articulated hand model is fitted to the data to refine the first estimate. The system is able to estimate the first 7 Degrees of Freedom of the hand within 200 ms. The reconstructed hand is visualized in AVANGO/Performer and can be utilized to implement a natural man-machine interface.

3.4 Continuous sign language recognition

Continuous gesture recognition brings harder challenges and problems to be solved. A primary goal of continuous gesture recognition research is to create a system which can identify and use human gestures to convey information.
We had access to some literature where several studies and solutions are presented and between them we can highlight a few:

A) Hidden Markov Models

"***A Hidden Markov Model-based continuous gesture recognition system for hand motion trajectory***" [51]. Elmezain et al. propose an automatic system that recognizes both isolated and continuous gestures for Arabic numbers (0-9) in real-time based on hidden Markov model (HMM). To handle isolated gestures, HMM using ergodic, left-right (LR) and left-right banded (LRB) topologies with different number of states ranging from 3 to 10 is applied. Orientation moving features are obtained from spatio-temporal trajectories and then quantized to generate its codewords. The continuous gestures are recognized by their novel idea of zero-codeword detection with static velocity motion.

Eickeler et al. work entitled "***Hidden Markov model based continuous online gesture recognition***" [52] presents the extension of an existing vision-based gesture recognition system using hidden Markov models(HMMs) with improvements to increase the capabilities and the functionality of the system. These improvements include position independent recognition, rejection of unknown gestures, and continuous online recognition of spontaneous gestures.

B) **Bayesian Classifier**

"**Continuous Gesture Recognition using a Sparse Bayesian Classifier**" [53]. Wong and Cipolla proposed an approach to recognize and segment 9 elementary gestures from a video input applied to continuous sign recognition. An isolated gesture is recognized by first converting a portion of video into a motion gradient orientation image and then

classifying it into one of the 9 gestures by a sparse Bayesian classifier. The portion of video used is decided by using a sampling technique based on condensation. By doing so, gestures can be segmented from the video in a probabilistic manner.

C) **Neural Networks**

Bailador et al. present "**Real time gesture recognition using continuous time recurrent neural networks**" [54]. A new approach to the problem of gesture recognition in real time using inexpensive accelerometers. This approach is based on the idea of creating specialized signal predictors for each gesture class. These signal predictors forecast future acceleration values from current ones. The errors between the measured acceleration of a given gesture and the predictors are used for classification. This approach is modular and allows for seamless inclusion of new gesture classes. These predictors are implemented using Continuous Time Recurrent Neural Networks (CTRNN).

D) **Latent- Dynamic Discriminative Models**

"**Latent-Dynamic Discriminative Models for Continuous Gesture Recognition**" [55]. Morency et al. developed a discriminative framework for simultaneous sequence segmentation and labeling which can capture both intrinsic and extrinsic class dynamics. Their approach incorporates hidden state variables which model the sub-structure of a class sequence and learn dynamics between class labels. Each class label has a disjoint set of associated hidden states, which enables efficient training and inference in our model.

E) **Invariant Features**

Campbell et al. on their work "**Invariant features for 3-D gesture recognition**" [56], tested ten different feature vectors in a gesture recognition task which utilizes 3D data gathered in real-time from stereo video cameras, and HMMs for learning and recognition of gestures. Their results indicate that velocity features are superior to positional features, and partial rotational invariance is sufficient for good performance.

3.5 Conclusions

This chapter was an overview of the "state-of-the-art" in Gesture and Sign Language Recognition. We highlighted contributions divided in three classes according to the organization and architecture of our own thesis work. First (section 3.1), we enumerated a set of works that used a range of different devices. Then (section 3.2), some contributions that covered "*static hand posture recognition and finger-spelling*" were referred. In section 3.3, we highlighted some works on "*Isolated Moving hand gesture and sign language recognition*" and finally, in section 3.4, we addressed *Continuous sign language recognition*", which is out of the scope of this work, but still important to mention.

I have not failed. I've just found 10,000 ways that won't work.

Thomas A. Edison

4. System Architecture and Software Design

As mentioned in previous chapters, this thesis has focused on a first approach to the problem of Automatic Gesture Recognition, considering two types of gestures: *Isolated Static Gestures* and *Isolated Moving Gestures*. To tackle these two problems, we propose a two level system architecture, presented in this chapter.

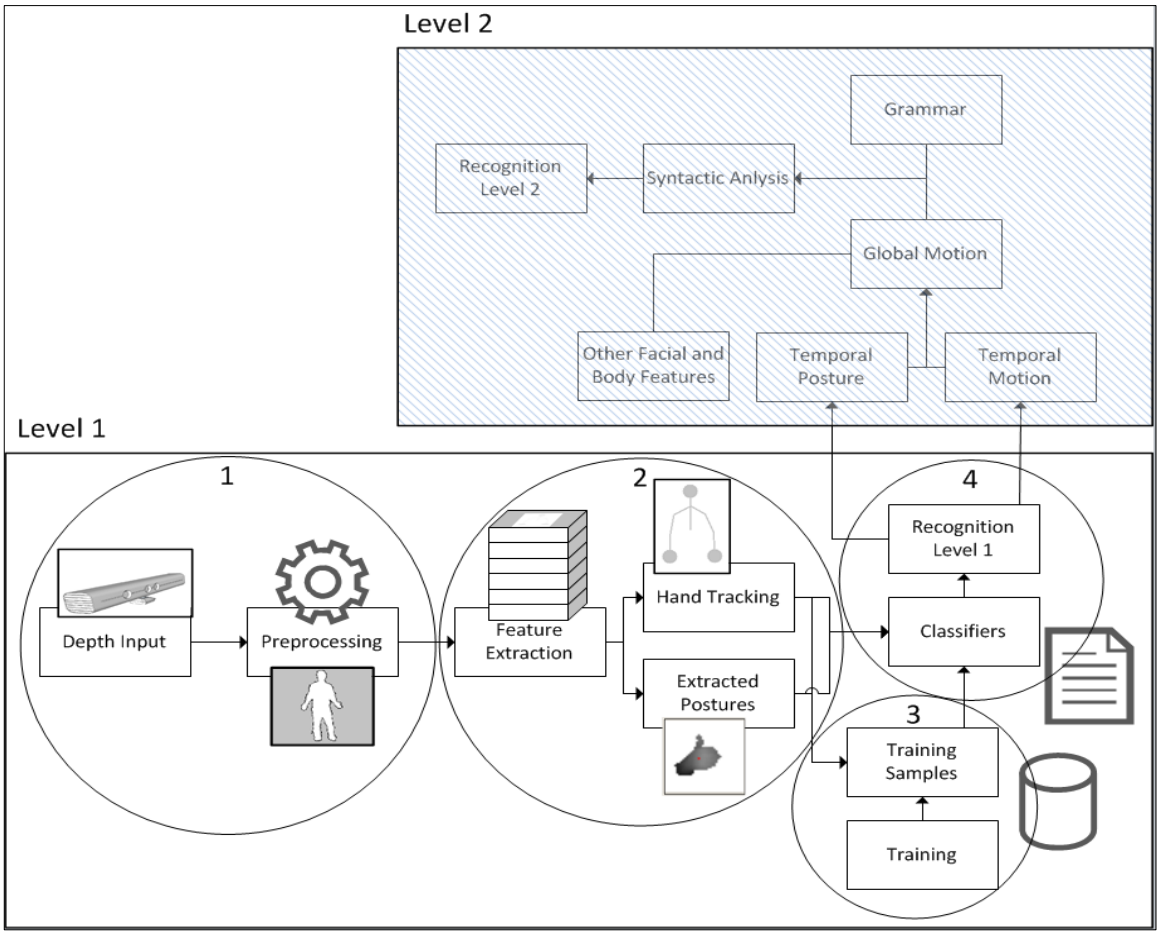

Figure 5 – Two-level architecture for Gesture Recognition

4.1 System Architecture

The proposed architecture is depicted in *Figure 5*. The first level covers *Isolated Gesture Recognition* and the second level covers *Continuous Gesture Recognition*. For the first level, both *Static* and *Moving Gesture Recognition* have four main modules. The first module performs signal acquisition and preprocessing from sensor data. The second module handles data processing, feature extraction and logging. The third module produces and stores training sample data. Finally, the fourth module creates classifiers and gesture recognition.This first module prepares organizes and provides all the information needed for the "*Setup*", "*Training*" and "*Recognition*" processes.

4.2 Acquisition and Preprocessing from Sensor Data

Depth Data and RGB Data are collected and processed in this module, as well as runtime variables, constants and data stacks of head, hands and elapsed time. Display positions of joints are calculated. Information is appended and cleared whenever it is necessary. Another function of this module is to allow the setup of camera parameters and methods such as "set camera optimal angle position", "start camera" and "stop camera".

This module provides functions to calculate and get real-time *global speed* of movements as well as *global speed* of *right* and *left hand movements.*

Some auxiliary functions were also implemented in this module, such as "get joint position in display" or "get polylines of body segments from a set of points", "distance of 3D vectors", "distance from a 3D vector to skeleton head" and "normalized direction of hand joints". The module comprises also a set of posture acquiring functions to get and convert posture dimensions and also get postures from the respective joints, 3D positions and Depth Data.

4.3 Data Processing, Feature Extraction and Logging

The second module adds the functionality needed for the feature extraction and processing. This feature extraction will provide information for real-time automatic gesture recognition and also provide the training samples set, that will be stored in a database, which is a critical data source for the gesture recognition process.

We have named **postures** (*Figure 9*) to static gesture representations and **movements** (*Figure 18*) to moving gesture representations. Both of these representations require a data structure and a set of methods to manipulate and provide the tools for the feature extraction process (*Figure 10*,Figure 19).

Some methods were created, as mentioned in above sections, to perform adjustments, such as resizing, several types of translations on different axes as well as scaling, and to enable *dexel* data normalization (*Figure 13*).

The module implements also the creation of bitmap images from **postures** and **movements** and data exporters to aid on later analysis for automatic gesture recognition performance, reliability and accuracy.

4.4 Classifiers and Recognition

The Runtime Recognition process has two main windows for **posture** and **movement** respectively (*Figure 6*,*Figure 7*). After the learning process a notification pops up and the system is ready for the Recognition.

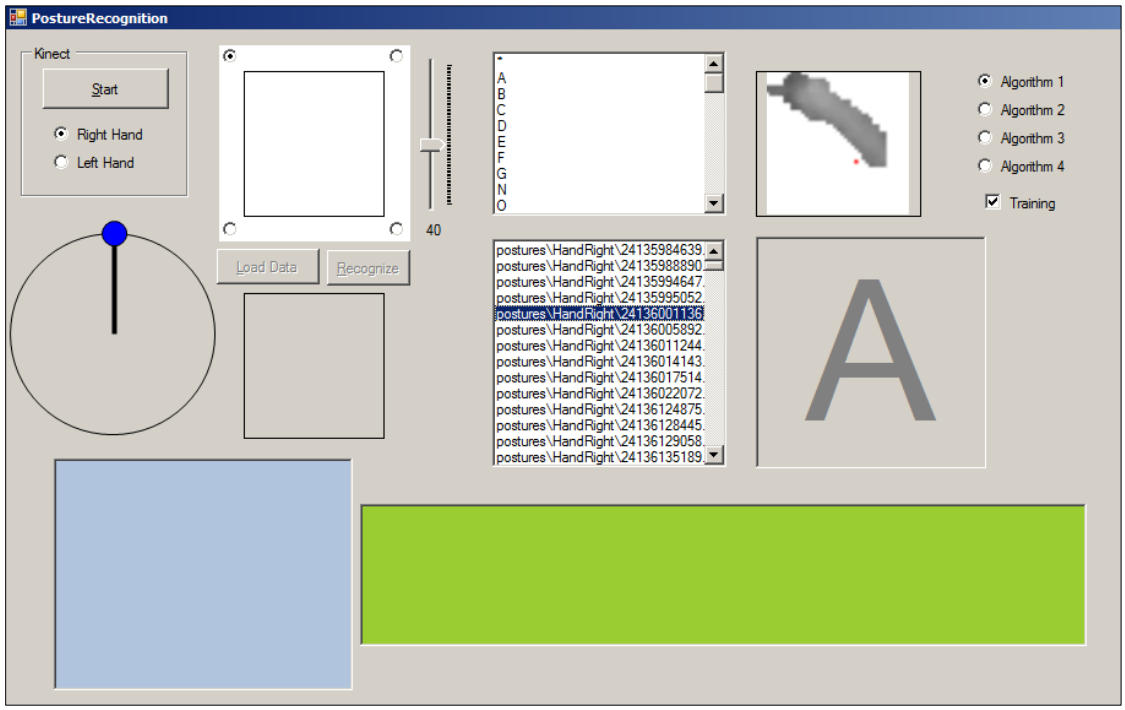

Figure 6 - Posture Recognition screen

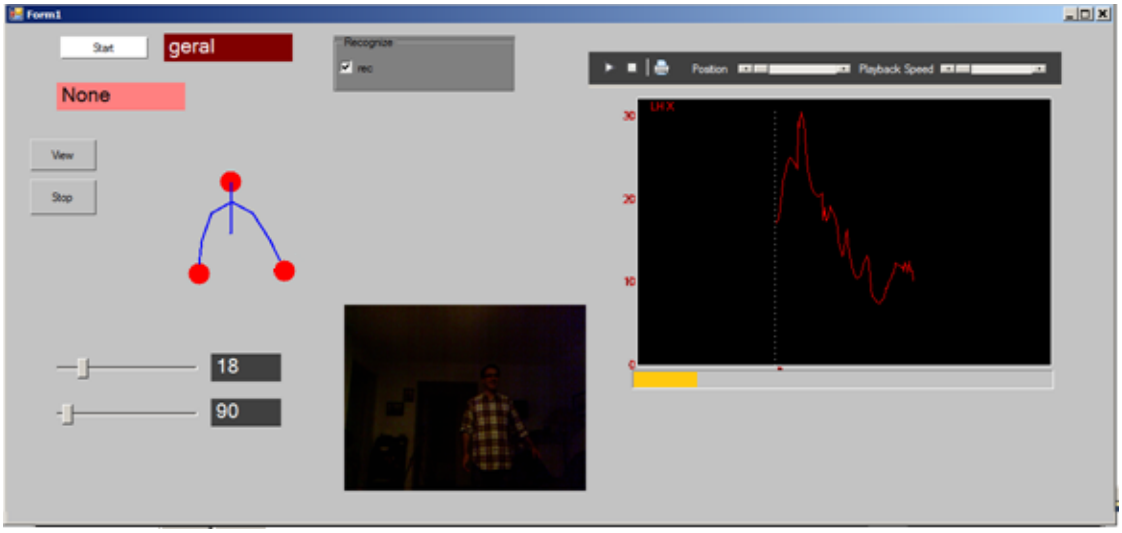

Figure 7 - Movement recognition screen

4.5 Development Environment: Technologies, Platforms and SDKs

IDE: Microsoft Visual Studio 2010 [63]

Programing Languages: C#

Frameworks: .Net Framework 4.0 ; Accord.NET Framework-2.3.0 - Module Statistics

SDKs: Microsoft Research Kinect for Windows SDK

Visual Studio 2010 Code Metrics:

Project: PSLRCVDS

Description: This component has the main function of integrating the other components in a user-friendly interface. It provides the navigation and operation between the collection, storage and feature processing interfaces and the recognition and analysis functions.

It includes several application screens such as "Posture management", "Posture Recognition", "Movement management" and also "Movement recognition" screens that make use of Kinect.dll.

Assembly: PSLRCVDS.exe

Maintainability Index: 61

Cyclomatic Complexity: 427

Depth of Inheritance: 7

Class Coupling: 129
Lines of Code: 2.495

Project: Kinect
Description: This library provides all of the necessary functions to make data collection, storage and feature extraction. It has a set of methods to make use of the Recognition algorithms in real-time. It makes the integration of the information provided by the Kinect SDK with the specific Gesture Recognition information using Depth Sensors.
Assembly: Kinect.dll
Maintainability Index: 72
Cyclomatic Complexity: 1.246
Depth of Inheritance: 7
Class Coupling: 158
Lines of Code: 3.339

Project: KHMM
Description: This library provides Hidden Markov Model (HMM) specific functions. It presents all of the necessary methods to make use of the HMM algorithms in real-time. It makes the integration of the information provided by the Kinect.dll with the Hidden Markov Model Algorithms and sequence classifiers provided by the *Accord.Net* framework [62].
Assembly: KHMM.dll
Maintainability Index: 69
Cyclomatic Complexity: 33
Depth of Inheritance: 1
Class Coupling: 29
Lines of Code: 117

4.6 Summary

In this chapter, we have presented a short summary of the system architecture developed to support this thesis, which we believe is adequate to deal with the problems of static and moving isolated hand gesture recognition that have an application to simple problems in

Automatic Sign Language Recognition. We have also described the visual interfaces, regarding the modules of "*Acquisition and Preprocessing from Sensor Data*", "*Data Processing, Feature Extraction and Logging*", "*Producing and Storing Training Sample Data*" and "*Classifiers and Recognition*". Finally, we have presented thedevelopment environment, including technologies, platforms and SDKs.

5. Development

The desire of knowledge, like the thirst for riches, increases ever with the acquisition of it.

Laurence Sterne

In this chapter we introduce and explain in detail, our approach to solve the thesis problem and the adopted methodologies towards the respective solution. The topic of the first subsection 5.1 is our work on depth sensor signal pre-processing. Then we explain our approach to deal with Isolated Static Gestures, including the respective algorithm formalization. In the third subsection, our approach on Isolated Moving Gestures, particularly Hand Movements, is presented. Our algorithms, "3D Path Analysis" and an adaptation of Hidden Markov Models (HMM), are then formalized and explained.

5.1 Depth Sensor Signal Pre-Processing

In this subsection we will make some considerations about theoretical aspects of depth sensor signal processing. In practice both the Kinect Sensor Hardware and the Microsoft Research Kinect SDK, which we are using for this work, implement these functions, but it is anyway important to give a theoretical overview on this topic. We will provide an initial approach to background subtraction, using only depth information.

5.1.1 Simple Background Subtraction with Depth information

Based on the generalization of a 2D approach we propose a very simple method adapted to use depth sensor information, to perform background subtraction.

Consider $I_1, I_2, \ldots, I_n$ as a sequence of 640 x 480 of depth (11-bit) pixel values for the time instants 1, 2, ... *n,* respectively.

- For the instant 1:

- IA = I_1 (5.1.17)

- For the instants *m* such as *m* < 100:

  - We consider I_m as being the matrix/depth map for instant *m*: *m* < 100.
  - Consider the depth values *p*(*i, j*) from I_m and IA represented by m_i and ia_i respectively.
  - If $m_i \leq ia_i$ then $m_i \rightarrow ia_i$ (5.1.18)
  - That is, IA will now contain the lowest depth values observed in the first 99 frames, assuming that the sensor only "sees" the scenario (that, is the user should be absent in this first set-up phase).

- Then for the instants *n* such as *n* > 99:

  - Let λ =10 be a sensor noise tolerance factor.
  - We calculate the matrix MN (8-bit) – That is, the mask of the values l_i from I_n less or equal to ($ia_i - \lambda$). That is $mn_i = 255$ if $l_i \leq (ia_i - \lambda)$ and $mn_i = 0$ for the remaining cases.

We can see the observed results in *Figure 8*:

Figure 8 – With background subtraction, the body silhouette becomes the only visible object as it moves around the sensor field of view.

We can then use this mask to obtain RGB pixels of the mask or any other task.

5.1.2 Summary

In this subsection 5.1 we have made some considerations about theoretical aspects of *Depth Sensor Signal Processing*. We have provided an approach for "simple background subtraction with depth information", applied to the adopted *Kinect Sensor*.

5.2 Isolated Static Gestures

In this section, we will introduce our approach to perform hand posture representation and recognition. We will describe the representation model, the algorithms and the technique to find the distance used to match posture features with models from our training corpus database.

5.2.1 Posture Representation

The sensor depth data stream provides frames in which the 11 bits of each pixel gives the distance, in millimeters, to the nearest object at that particular *x* and *y* coordinate in the depth sensor's camera reference frame. We are using the frame size of 320×240 pixels from the available depth data streams.

The NUI Skeleton API [59] provides information about the location of a set of points, called skeleton positions. This skeleton represents a user's current position and orientation, or pose. We have used skeleton data from the tracking API, to get both hands joint positions. These positions *x*, *y*, and *z* coordinates are defined in the skeleton reference frame. Unlike the units of depth image space, these point coordinates are expressed in meters. The API provides a method to return the depth space coordinates of a given pixel belonging to the skeleton image.

This way we have all the information to represent elements that we call "**postures**" (*Figure 9*). *Postures* are sets of depth information captured in a cubic space of points in skeleton space, $(x_0, x_1, x_2)\ with\ -\mathrm{n}/2 < x_i < \mathrm{n}/2$. We register this information converted to depth space in a scaled $n \times n$ depth information matrix calibrated around the centroid point.

Figure 9 - Left hand graphical posture representation

The information is stored in XML files and represented by a class in the following structure (*Figure 10*):

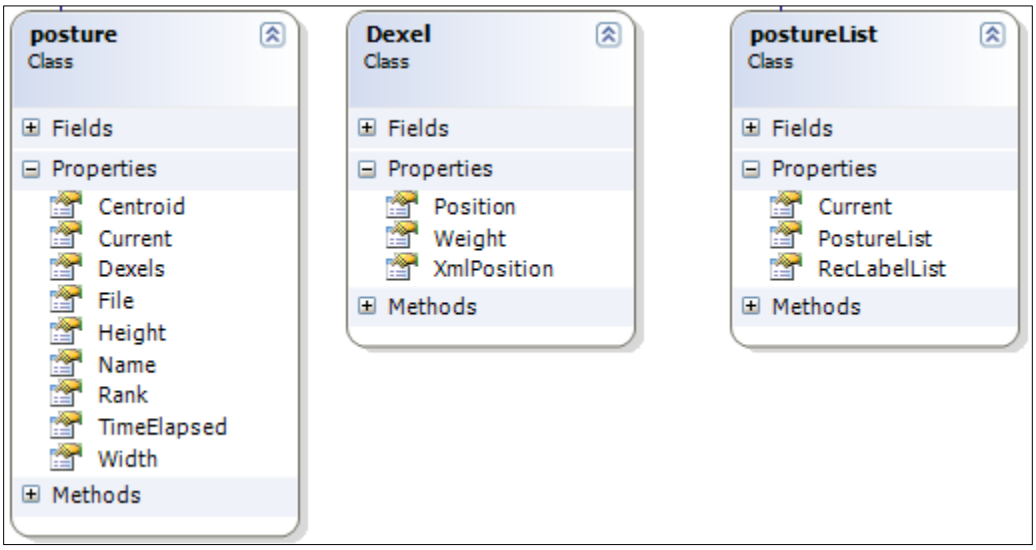

Figure 10 - Posture class structure

Postures are collections of what we refer to as *dexels* (an abbreviation for ***de****pth picture* ***el****ements*) and can be grouped in lists of homogeneous *postures* called *postureLists*. For example, a *postureList* can contain a group of *postures* representing the same static symbol or letter in sign language.

5.2.2 Posture Matching Algorithms

To perform static gestures recognition we have developed two different techniques, both based on Depth Sensor data. The first one uses our novel **3D Template Matching** and the second one is based in **2D Euclidean Distance.** These methods are based in linear interpolation, which computes average shapes from depth point sets. Both techniques use a base training corpus of postures and a test corpus obtained at run-time, to make real time recognition of two hands gestures. We provide a comparison between the results of both algorithms and we believe we have created the basis for a future hybrid approach using both techniques, as they use complementary features.

5.2.3 Assumptions and preparation steps of the two techniques

A. Transforming the original "posture", to a posture translated to a given corner of the cubic space.

To accelerate the Template Matching correlation, reducing the effects of noisy data and also reducing the effects of "trembling" skeleton joint positions from NUI Skeleton API, we have created a method that we refer to as **Corner Matching** explained bellow.

Corner matching can process different hands and different positions using the appropriate or *best matching corners*, moving the *dexels* to those corners, generating target postures (postures translated to the appropriate corner of the cubic space).

The Posture Matching Algorithms (sections 5.2 and 5.3), analyze postures after being adjusted to the 2D (*x*, *y*) corners using this method and give back the results.

To generate a target posture from the original posture, we use the following procedure (*Figure 11*, right hand top corner):

Figure 11 - Left Hand adjusted to top right corner

Let's consider the source posture **P** and

$\mathrm{D} = \{d = (x, y, z): d \text{ is a } dexel \text{ from } \mathbf{P}\}$

$(MIny, MAxx) = (y_d, x_d): y_d < y_i \wedge x_d > x_i, (x_i, y_i, z_i) \in D$

For the top right corner:

$(s, t) = (w - MAxx - 1, MIny)$

$Target\ posture\ \boldsymbol{T} = \{(x + s, y - t, z): (x, y, z) \in D\}$

Making analogy for the other 3 corners we have:

For the lower right corner:

$(s, t) = (w - MAxx - 1, h - MAyy - 1)$

Where

$w = width(P)\ (width\ of\ the\ posture);$

$h = height(P)\ (height\ of\ the\ posture); Target\ posture\ \boldsymbol{T} = \{(x+s, y+t, z): (x, y, z) \in D\}$

For the top left corner:

$(s, t) = (MIxx, MIny)\ \ Target\ posture\ \boldsymbol{T} = \{(x-s, y-t, z): (x, y, z) \in D\}$

For the lower left corner:

$(s, t) = (MIxx, height\ -\ MAyy\ -\ 1)\ \ Target\ posture\ \boldsymbol{T} = \{(x-s, y+t, z): (x, y, z) \in D\}$

B. Another problem we faced was **Scaling Invariance**.

Scaling of postures usually cause poor matches in Template Matching [60]. We have used a NUI Skeleton API [59] method, to convert skeleton data scale into depth image scale. This way we can solve this problem and we can get similar posture sizes independently from the distance from the user to the sensor. The main purpose was to get a corpus database of postures of the same size and also to get real time recognition of a posture of that same size. Despite these corrections, near or far from the range limits of the field of view of the sensor, we have observed poor results mainly because of lack of *dexel* information after the scaling transformation (*Figure 12*).

As for future work, we hope to address the problem of hand rotation invariance. A way forward could be to make use of information of other joints linked to the arms, as the wrist, elbow and shoulder. This may be a way to make a more effective sample screening and achieve a better matching.

Figure 12 - Lack of dexel information when scaling or when the hand is far away from sensor field of view.

C. Last but not the least, we have solved **misalignments in size of posture dimension and depth information**, caused by the trembling observed in skeleton joint positions and differences in distances to the sensor (*Figure 13*).

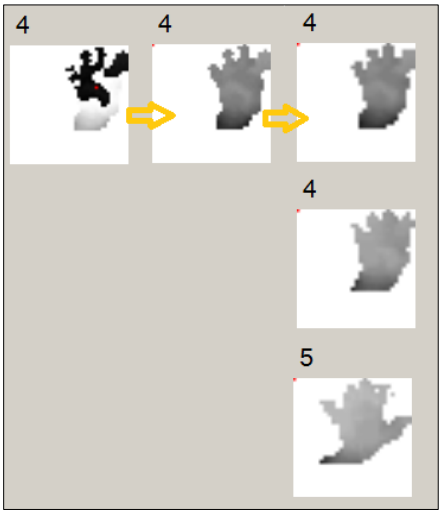

Figure 13 - Depth alignment and image scaling. In the first row, we depict 3 stages of posture named "4".In second row, the final stage of posture named "4". In third row, the final stage of posture named "5"

To perform the depth alignment, our first approach was to compute the centroid point and make the translation of *dexels* in the *z* axis direction in order to standardize all postures. Then we have changed this approach and, instead of considering the centroid, we have started to use the joint position. As the sensor API provides that information, we have realized that there was no need to use the centroid, which takes into account parts of the arm that are not relevant to the process. Hand size according to morphology of different age users is another subject on which we expect to dedicate some study in the future.

5.2.4 Posture Matching Algorithm 1 – Method of 2D Distances

In this algorithm we have used features of orthogonal linear contours (*Figure 15*), by maximization and minimization of values from depth data information (*Figure 14*).

The orthogonal contours of the posture are extracted and used to check for the best match among training data on *PostureLists*.

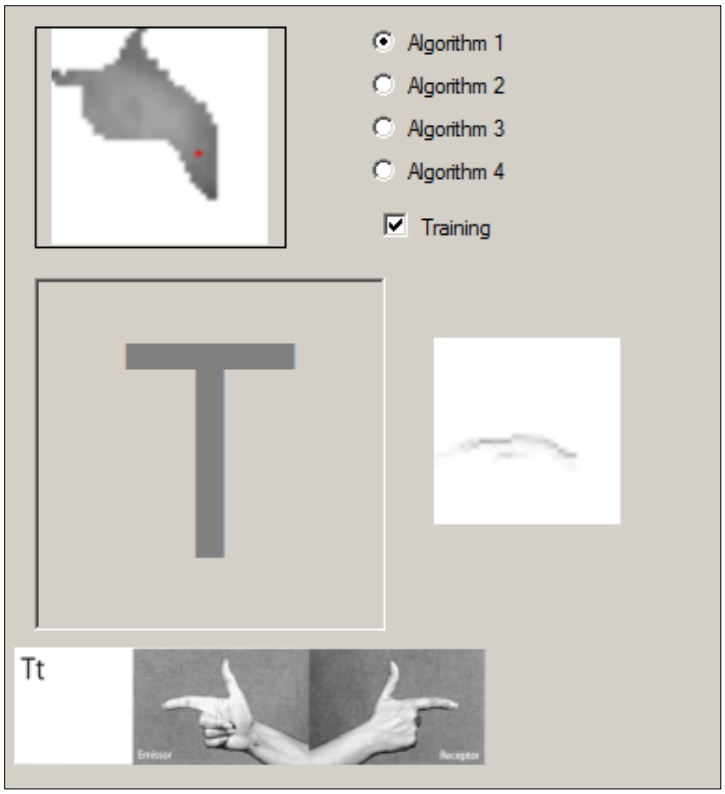

Figure 14 - Depth data representation of PSL letter T

Let $\boldsymbol{L}$ be a set of *PostureLists* and $\boldsymbol{P_j}$, a posture performed in the application in real time, having j=1,2,…,t time ticks. All postures have been pre-processed with a transformation that translates each of the postures to a given corner of the cubic space and also, for scaling invariance. We can understand the *PostureLists* also as a pre-computed training set of postures, with each list representing a finite set of sample postures of one static gesture.

1 – Compute Principal Orthogonal Linear Contours - *POLC*.

We want to find the *PostureList* that minimizes *POLC function* defined bellow*:*

For D_j being the set of *dexels* in any posture P_j and $\boldsymbol{L}$ to be a set of *PostureLists*. Let's compute $\boldsymbol{POLC}(\boldsymbol{P_j}, \boldsymbol{L})$:

$$dA_j = \{(y_j, x_k)\} : (x_j, y_j, z_j) \in D_j \ \wedge y_j < y_k \ \ \forall\, (x_k, y_k, z_k) \ \in D_j$$

$\forall\ (y_j, x_k)\ \in dA_j : FA(y_j, x_k) = y_j$, i.e. *FA* returns the lowest *y* values, for all *x* of a set of *dexels* D_j.

$dC_j = \{(y_j, x_k)\} : (x_j, y_j, z_j) \in D_j\ \wedge y_j > y_k\ \ \forall\ (x_k, y_k, z_k)\ \in D_j$

$\forall\ (y_j, x_k)\ \in dC_j : FC(y_j, x_k) = y_j$, i.e. *FC* returns the highest *y* values, for all *x* of a set of *dexels* D_j.

$dB_j = \{(x_i, y_k)\} : (x_i, y_i, z_i) \in D_j\ \wedge x_i > x_k\ \ \forall\ (x_k, y_k, z_k)\ \in D_j$

$\forall\ (x_i, y_k)\ \in dA_j : FB(x_i, y_k) = x_i$, i.e. *FB* returns the highest *x* values, for all *y* of a set of *dexels* D_j.

$dD_j = \{(x_i, y_k)\} : (x_i, y_i, z_i) \in D_j\ \wedge x_i < x_k\ \ \forall\ (x_k, y_k, z_k)\ \in D_j$

$\forall\ (x_i, y_k) \in dD_j : FD(x_i, y_k) = x_i$, i.e. *FD* returns the lowest *x* values, for all *y* of a set of *dexels* D_j.

$S = |dA_j| = |dC_j| = |dB_j| = |dD_j| = width\ P_j = Heigth\ P_j = 40$ (see example in *Figure 15*)

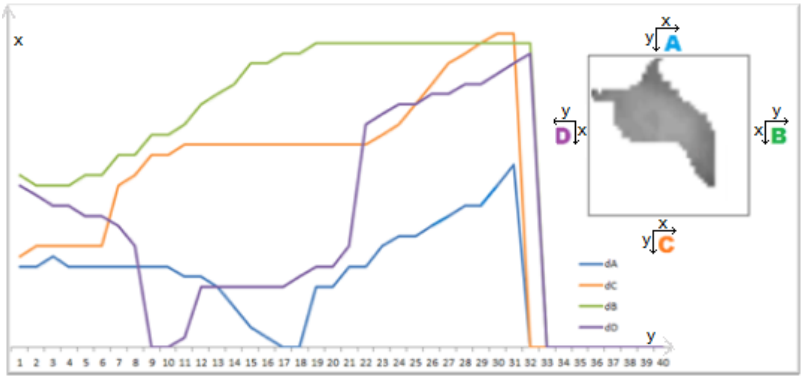

Figure 15 - Linear contours of PSL letter T.

Let $P_l\ \in L$

$$DA\ (P_j\ , P_l) = \sum_{k=1}^{S} \sqrt{\left(FA(y_j, x_k) - FA(y_l, x_k)\right)^2} : (y_j, x_k) \in dA_j, (y_l, x_k) \in dA_l$$

$$DC\ (P_j\ , P_l) = \sum_{k=1}^{S} \sqrt{\left(FC(y_j, x_k) - FC(y_l, x_k)\right)^2} : (y_j, x_k) \in dC_j, (y_l, x_k)\ \in dC_l$$

$$DB\ (P_j\ , P_l) = \sum_{k=1}^{S} \sqrt{\left(FB(x_j, y_k) - FB(x_l, y_k)\right)^2} : (x_j, y_k) \in dB_j, (x_l, y_k)\ \in dB_l$$

$$DD\ (P_j\ , P_l) = \sum_{k=1}^{S} \sqrt{\left(FD(x_j, y_k) - FD(x_l, y_k)\right)^2} : (x_j, y_k) \in dD_j, (x_l, y_k)\ \in dD_l$$

$$D\ (P_j\ , P_l) = DA\ (P_j\ , P_l) + DC\ (P_j\ , P_l) + DB\ (P_j\ , P_l) + DD\ (P_j\ , P_l)$$

$$\boldsymbol{POLC}\left(\boldsymbol{P_j}, \boldsymbol{L}\right) = D\ (P_j\ , P_l) : D\ (P_j\ , P_l) < D\ (P_j\ , P_n)\ \forall\ P_n\ \in L \qquad (5.2.0.1)$$

The *PostureList* that minimizes *POLC* represents the most likely gesture for $\boldsymbol{P_j}$

1 – Extend the Algorithm to consider depth.

As we have depth information from sensor (*Figure 16* and *Figure 17*), we can extend the previous algorithm to handle depth. Therefore, we can have:

$$dE_j = \{(z_j, x_k)\} : (x_j, y_j, z_j) \in D_j\ \wedge z_j < z_k\ \ \forall\ (x_k, y_k, z_k)\ \in D_j$$

$$\forall\ (z_j, x_k)\ \in dE_j : FE(z_j, x_k) = z_j$$

$$dF_j = \{(z_i, x_k)\} : (x_i, y_i, z_i) \in D_j\ \wedge z_i > z_k\ \ \forall\ (x_k, y_k, z_k)\ \in D_j$$

$$\forall\ (z_j, x_k)\ \in dF_j : FF(z_j, x_k) = z_j$$

$$dG_j = \{(z_i, y_k)\} : (x_i, y_i, z_i) \in D_j\ \wedge z_i < z_k\ \ \forall\ (x_k, y_k, z_k)\ \in D_j$$

$$\forall\ (z_i, y_k)\ \in dG_j : FG(z_i, y_k) = z_i$$

$$dH_j = \{(z_i, y_k)\} : (x_i, y_i, z_i) \in D_j\ \wedge z_i > z_k\ \ \forall\ (x_k, y_k, z_k)\ \in D_j$$

$$\forall\ (z_i, y_k) \in dH_j : FH(z_i, y_k) = z_i$$

$S = |dE_j| = |dF_j| = |dG_j| = |dH_j| = width\ P_j = Heigth\ P_j$=40 (see example in *Figure 16*).

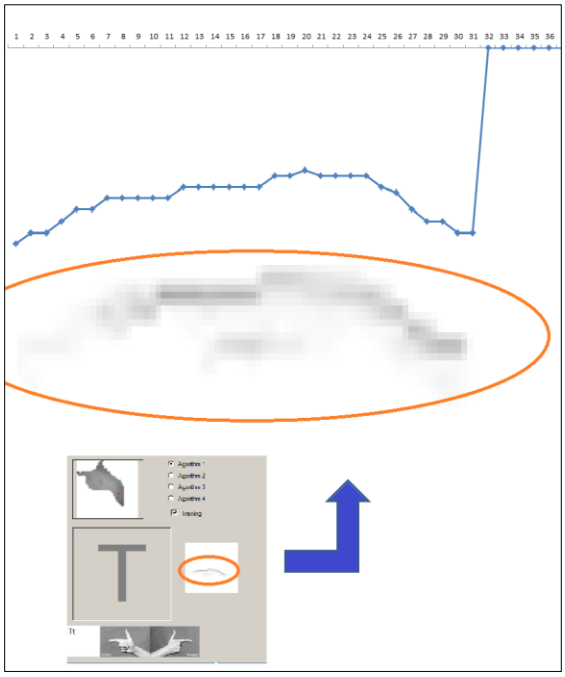

Figure 16 -Detail of depth representation and depth contour

$$DE\ (P_j\ ,P_l) = \sum_{k=1}^{S} \sqrt{\left(FE(z_j,x_k) - FE(z_l,x_k)\right)^2} : (z_j,x_k) \in dE_j, (z_l,x_k) \in dE_l$$

$$DF\ (P_j\ ,P_l) = \sum_{k=1}^{S} \sqrt{\left(FF(z_j,x_k) - FF(z_l,x_k)\right)^2} : (z_j,x_k) \in dF_j, (z_l,x_k) \in dF_l$$

$$DG\ (P_j\ ,P_l) = \sum_{k=1}^{S} \sqrt{\left(FG(z_j,y_k) - FG(z_l,y_k)\right)^2} : (z_j,y_k) \in dG_j, (z_l,y_k) \in dG_l$$

$$DH\ (P_j\ ,P_l) = \sum_{k=1}^{S} \sqrt{\left(FH(z_j,y_k) - FH(z_l,y_k)\right)^2} : (z_j,y_k) \in dH_j, (z_l,y_k) \in dH_l$$

$$D2\ (P_j\ ,P_l) = DA\ (P_j\ ,P_l) + DC\ (P_j\ ,P_l) + DB\ (P_j\ ,P_l) + DD\ (P_j\ ,P_l) + DE\ (P_j\ ,P_l) + DF\ (P_j\ ,P_l) + DG\ (P_j\ ,P_l) + DH\ (P_j\ ,P_l)$$

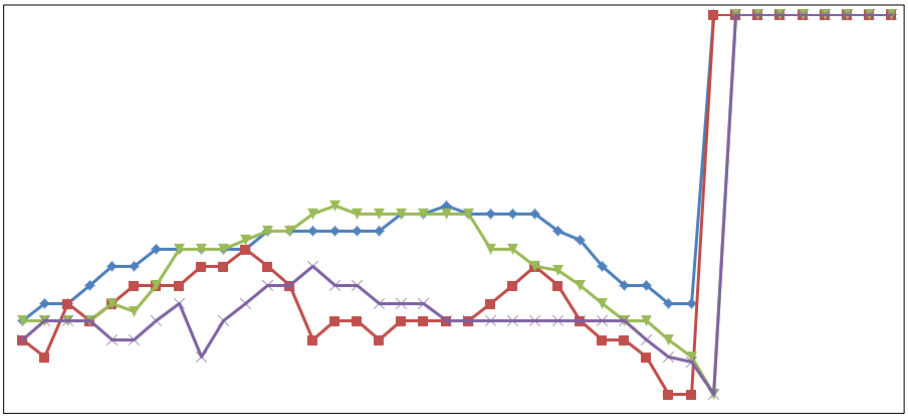

Figure 17 - Graphical representation of 4 depth contours.

$$\boldsymbol{POLC2(P_j, L)} = D2\ (P_j\ ,P_l) : D2\ (P_j\ ,P_l) < D2\ (P_j\ ,P_n)\ \forall\ P_n\ \in L$$

5.2.5 Posture Matching Algorithm 2 – Skeletal-based Template Matching Adaptation

This algorithm is based on the sum of absolute differences from current posture to each posture from the *PostureList* (from the training database).

Consider $\boldsymbol{L}$ to be a set of *PostureLists* and $\boldsymbol{P}$ a posture performed in application in real time. We first define three auxiliary functions, namely, **CompareAvgDistance, CountXYInOut** and **CountXYOutIn:**

1 – CompareAvgDistance, $\boldsymbol{CAD}$

This method is based on Euclidean Distances.

For the current performed posture - $\boldsymbol{P}$ and for each PostureList - $\boldsymbol{PL} \in \boldsymbol{L}$, we can compute $\mathbf{CAD}\ (\boldsymbol{P}, \boldsymbol{PL})$ **:**

Let $nPL = number\ os\ postures\ in\ \boldsymbol{PL}$

$$\boldsymbol{P} = \{(x_P, y_P, z_P)\} \wedge \boldsymbol{PL} = \{P(i) = (x_{PL}(i), y_{PL}(i), z_{PL}(i)) ;\ i = 1, \dots, nPL\}$$

$$\boldsymbol{CAD}(\boldsymbol{P}, \boldsymbol{PL}) = \frac{\sum_{i=1}^{nPL} \frac{\sum_A \sqrt{(z_{PL}(i) - z_P)^2}}{|A|}}{nPL}$$

$$where\ A = \{(x, y, z) : x = x_P = x_{PL}, y = y_P = y_{PL}\}$$

A is the set of *dexels* in ***P*** not having corresponding *dexels* in ***PL.***

2 – CountXYInOut - magnitude of posture dissimilarity

Let ***L*** be a set of *PostureLists.*

For the currently performed posture - ***P*** and for each postureList - $\boldsymbol{PL} \in \boldsymbol{L}$, we compute $\boldsymbol{CXIO}(\boldsymbol{P}, \boldsymbol{PL})$, that is, we want to count the number *dexels* in ***P*** not having corresponding *dexels* in ***PL*:**

$$\boldsymbol{CXIO}(\boldsymbol{P}, \boldsymbol{PL}) = |B|\ where,$$

$$B = \{(x, y, z) : x = x_P \wedge \nexists x : x = x_{PL} = x_P, y = y_P \wedge \nexists y : y = y_{PL} = y_P\}$$

3 – CountXYOutIn - magnitude of posture similarity

Let ***L*** be a set of *PostureLists.*

For the currently performed posture - ***P*** and for each postureList - $\boldsymbol{PL} \in \boldsymbol{L}$, we compute $\boldsymbol{CXOI}(\boldsymbol{P}, \boldsymbol{PL})$**,** that is, we want to count the number *dexels* in ***PL*** not having corresponding *dexels* in ***P:***

$$\boldsymbol{CXOI}(\boldsymbol{P}, \boldsymbol{PL}) = |C|\ where,$$

$$C = \{(x, y, z) : x = x_{PL} \wedge \nexists x : x = x_P = x_{PL}, y = y_{PL} \wedge \nexists y : y = y_P = y_{PL}\}$$

4 - Final Step: Get Best Matching *PostureList*

Let ***L*** be the *PostureLists* training set and ***P***, a real time performed posture. We want to find the best of ***P*** in ***L***. Our solution is to minimize the following formula:

$$For\ each\ \boldsymbol{PL}_j\ \in \boldsymbol{L}: i = 'best\ match'\ if$$

$$CAD(P, PL_i) + CAD(P, PL_i) * (\ (CXIO(P, PL_i) + CXOI(P, PL_i))$$

$$<= CAD\big(P, PL_j\big) + CAD\big(P, PL_j\big) *(\ (CXIO\big(P, PL_j\big) + CXOI\big(P, PL_j\big))$$

In both members of this inequality we have our proposal for a "posture matching equation". The equation has two members. The first tries to "favor" postures with the smallest distance between them and the second member, tries to "penalize" postures with different *dexels* between them. It uses a penalty factor equal to the average distance between the postures.

5.2.6 Summary

In this subsection we have addressed the first of the two classes of gestures covered by this thesis: Static Isolated Gestures and Hand Postures. We have started by explaining how to describe and represent hand postures. Initial assumptions and preparation steps of the introduced techniques were presented, and then the formal description of the two novel proposed algorithms, "*Method of 2D Distances"* and *"Skeletal-based Template Matching adaptation"*, were mathematically formulated.

5.3. Isolated Moving Gestures and Hand Kinematics

In this section, we present our approach to recognize Isolated Moving Gestures and the corresponding hand kinematics mathematical representation. We describe the representation model, the overall algorithms and the methods used to compare the features that capture the kinematics of moving gestures, within a test set, with feature models in a training set corpus.

5.3.1 Movement Representation

As we have referred earlier in section 3, the NUI Skeleton API [59] provides the location of a set of human skeleton joint positions, in a given skeleton reference frame. This skeleton represents a user's current position and orientation (pose). We have used skeleton data (*x*, *y*, *z* coordinates in meters), taken from skeleton tracking to get both

hands joint positions and also head position, since all considered hand movements, are relative to the head position.

This way we have all the data to represent elements that we have named "***movements***". *Movements* are time sequences of *feature vectors* representing the hands and head laws of motion over time (kinematics), defined in the skeleton reference frame which is centered in the head. We have recorded the head position and, for this first approach, we have considered continuous kinematic laws of just two 3D points obtained directly from the NUI Skeleton API [59]: one for the left hand and another for the right hand. This gives us a 6-dimensional feature vector (*Figure 18*), representing *x, y, z* coordinates of both hands in the mentioned skeleton reference frame.

At instant T we have $p(T) = \left(x_l(T), y_l(T), z_l(T), x_r(T), y_r(T), z_r(T)\right) \in \mathbb{R}^6$.

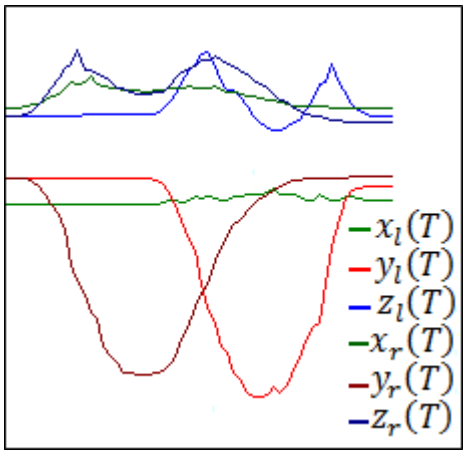

Figure 18 - Continuous signal from a movement made of x, y, z coordinates of both hands (left and right), relative to the head position.

This data is stored in XML files, represented by a class with the following structure (Figure 19):

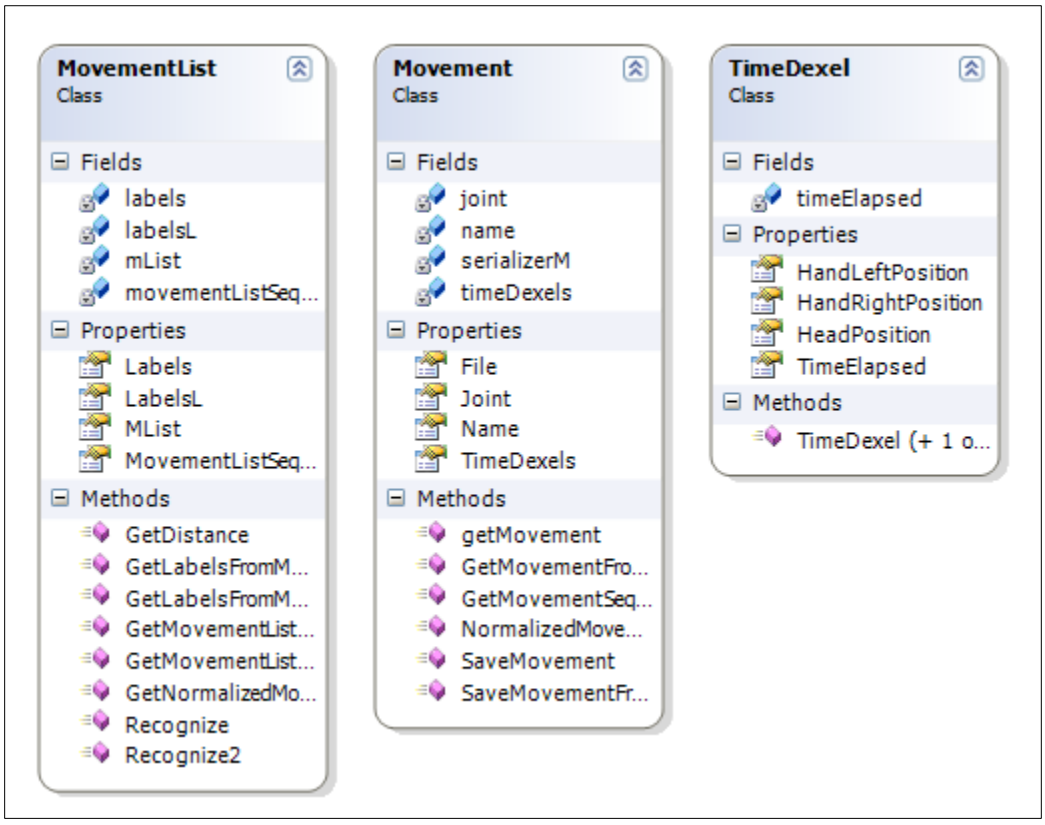

Figure 19 - Structure of Classes from Moving Gestures

Movements are collections of *TimeDexels* and can be grouped in lists of homogeneous *movements* called *MovementLists*. For example, a *MovementList* can contain a group of *movements* representing the same moving gesture in a sign language.

5.3.2 Isolated Moving Gesture Recognition

To perform moving gestures recognition we have developed algorithms based on **Generalized Euclidean Distances in 6-dimensional (6D) space** techniques, using as a training set, a base corpus of movements created by the author of the thesis and aiming at making real time recognition of both hands moving gestures. As the sensor does not provide continuous (in time) joint position information, we have used a discrete set of the same provided by the mentioned API.

5.3.3 Assumptions and preparation steps

A. The base information for movement analysis and registration comes from the NUI Skeleton API. Every time a skeleton data frame is ready, we have four *real time*

application stacks were we sequentially store, respectively, *time elapsed*, *right hand*, *left hand* and *head* information. *Time elapsed* is the number of milliseconds elapsed since 1-Jan-2011. This way the application can monitor the stacks with any desired criteria and use, collect and register only the selected segments of information.

B. For this approach we didn't use any complex co-articulation[9] solver (in fact, co-articulation is not handled in this thesis and will be subject for further research) and the *movement* segmentation method is based on the *global speed of movement* parameter, observed in real time. We have assumed a threshold for this parameter, above which the application determines that a *movement* corresponding to a gesture is being made.

Let's consider a threshold h and a delay d and four data sets T, L, R, H corresponding to time, left hand, right hand and head *application stacks,* respectively, and let's consider n the number of elements on each of the four *stacks* (every stack has the same number of elements n because all of them are appended at the same time *tick*).

We define the *global speed* of the *movement* when $n > d$ as:

$N = \{1,2,3, \dots, n\}, |L| = n\,, t_i \in T$:

$e(i) = t \;\forall\, i \in N \wedge t \in T$

$u(i) = \left(x_l(i), y_{l,}(i), z_l(i)\right) \in L \;\forall\; i \in N$

$v(i) = \left(x_r(i), y_r(i), z_r(i)\right) \in R \;\forall\; i \in N$

$M(i, i+d) = \left(e(i), u(i), e(i)\right)$ for $i, i+1, \dots, i+d \in N$

Let t be the last element from T $stack$

$$rs(d) = \frac{\sum_{j=n-d+1}^{n} \|v(j) - v(j-1)\|}{e(n) - e(n-d+1)} =$$

$$\frac{\sum_{j=n-d+1}^{n} \sqrt{(x_r(j) - x_r(j-1))^2 + (y_r(j) - y_r(j-1))^2 + (z_r(j) - z_r(j-1))^2}}{e(n) - e(n-d+1)}$$

[9] See 2.2 Hand Gesture Taxonomy (2-Continuous Gesture)

$$ls(d) = \frac{\sum_{j=n-d+1}^{n} \|u(j) - u(j-1)\|}{e(n) - e(n-d+1)} =$$

$$\frac{\sum_{j=n-d+1}^{n} \sqrt{(x_l(j) - x_l(j-1))^2 + (y_l(j) - y_l(j-1))^2 + (z_l(j) - z_l(j-1))^2}}{e(n) - e(n-d+1)}$$

$$fs(d) = rs(d) + ls(d) \qquad (5.3.0.1)$$

If $\boldsymbol{fs(d)} > h$ then $\boldsymbol{M(i, i+d)}$ is considered to be a *movement*

C. We assume for every *movement* corresponding to a moving gesture, that when the user starts the gesture, he/she is in the standing position, as depicted in *Figure 20*.

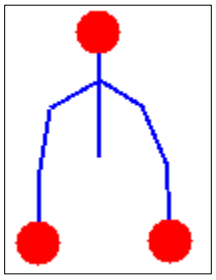

Figure 20 - Resting standing position from starting pose of moving gestures.

D. As a pre-processing step, we have applied a *time scaling* method to transform the *movement* sequence into an equivalent sequence with a different time interval, maintaining the proportions over time. With this scheme, every *MovementList* can be scaled to have all the movements with the same time duration (*Figure 21*).

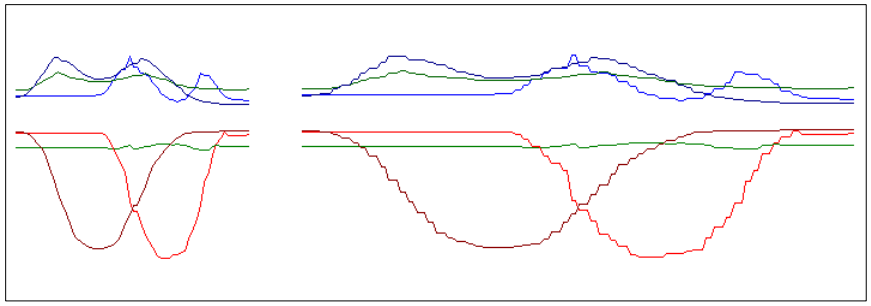

Figure 21 - Continuous 6-dimensional signal of moving gesture movement, before (left) and after (right) being time-scaled.

E. With the Isolated Moving Gesture Recognition technique, we intend to test moving gestures that take part in Sign Languages production, in particular PSL. We assume that *postures* (static hand gestures) and *facial expressions* are not taken into account in the recognition part of this technique. Therefore, in this section, our modeling effort and the validation experiments will only focus on the recognition of the kinematic aspects of the gesture. Our goal is to model and test the ability of the system to distinguish between different *movements*.

5.3.4 Isolated Moving Gesture Recognition Algorithm 1 – 3D Path Analysis

Consider L_i , i=1,…, d , to be a set of *movements*.

Consider $C = \{L_i\}$: L_i is a set of $movements$ from the same class (same gesture) and $M \in C$: $|\mathrm{M}| = d$

$$T = \{t(1), t(2), t(3), \dots, t(n))\}: |T| = n$$

T is the set of time values for the sensor processed frames 1,2,…, n

Let $t \in T$ and consider a feature vector in 6D:

$$u \in M : u = u(t) = (ux1(t), uy1(t), uz1(t), ux2(t), uy2(t), uz2(t))$$

Let V be a real time performed *movement* in 6D:

$$V = \{v(t)\} = \{\big(kx1(t), ky1(t), kz1(t), kx2(t), ky2(t)k, z2(t)\big)\}: |V| = d$$

(see *Figure 18*)

The distance between the real time performed gesture and the set M of *movements* from the same class (same gesture), becomes:

$$\mathrm{d}(V,M) = \frac{\int_{t(k-d)}^{t(k)} \|v(t) - u(t)\|\, dt}{d} : v \in V \wedge u \in M \qquad (5.3.0.2)$$

Note: We have a limited number *d* of observations from the sensor, so (5.3.0.3)

We replace $\int_{t(k-d)}^{t(k)} \|v(t) - u(t)\|\, dt$ by $\sum_{m-d}^{m} \|v(t) - u(t)\| : m > d \wedge m < n$ (see *Figure 22*)

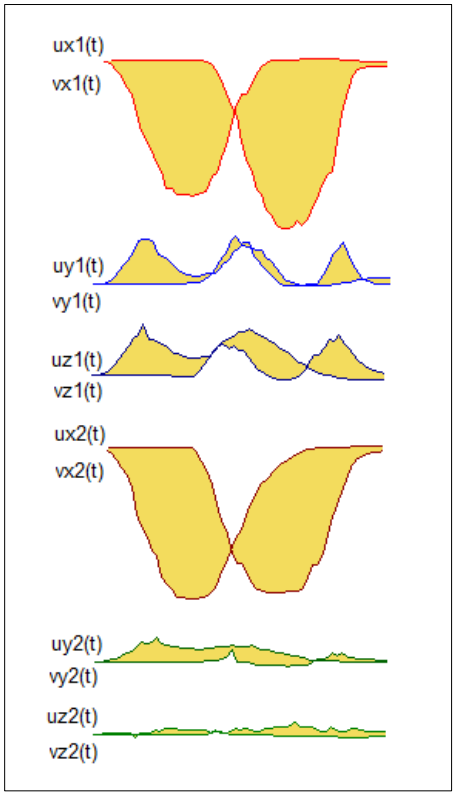

Figure 22 - Area of distance from 6 components of 2 movements.

The following condition represents the distance minimization and when it is fulfilled the performed gesture most likely belongs to the same class of M.

$$\text{For } M \in C \text{ , } \boldsymbol{M} \textbf{ is best match } \Longrightarrow \forall H_i \in C : d(V,M) \leq d(V,H_i)$$

5.3.5 Isolated Moving Gesture Recognition Algorithm 2 - HMM-based Moving Gesture Recognition

For this approach we have chosen the *Ergodic* (fully-connected) Topology for Hidden Markov Models[10], having two states. *Ergodic* models are commonly used to represent models in which a single (large) sequence of observations is used for training (such as when a training sequence does not have well defined starting and ending points and can potentially be infinitely long). Models starting with an *Ergodic* transition-state topology typically have only a small number of states, such that any state can be reached from any other state.

Left-Right (LR) and *Left- Right Banded* (LRB) topologies with different number of states for Hidden Markov Models could be considered in future approaches. Such HMM topologies are commonly used to initialize models in which training sequences can be organized in samples, such as in the recognition of spoken words. In spoken word recognition, several examples of a single word can be used to train a single Whole Word Model model, to achieve the most general model able to generalize over a great number of word samples.

a) In our problem of Gesture Recognition, the observations are 6-dimensional (6D) continuous sequences, since we are monitoring arbitrary kinematic laws of two 3D points in a given reference frame.

   Consider a set of those continuous signals

   $O = \{v(t)\} = \{(kx1(t), ky1(t), kz1(t), kx2(t), ky2(t)k, z2(t))\}$ (see *Figure 18*)

   Consider a *MovementList* L to be a set of movements M

   $O_l = \{L\}$: L is a set of $movements$ from the same class(same gesture).

   C is the set of all diferent classes of gestures (*Figure 23*)

   $O \in L : |O| = d$

[10] See Appendix A. Hidden Markov Models

In continuous time, we have a finite set of time observations from sensor processing: $T = \{t(1), t(2), t(3), \dots, t(n))\}$: $|T| = n$

For each $o_i \in 0$, we apply the Baum-Welch learning process in order to get the HMM's.

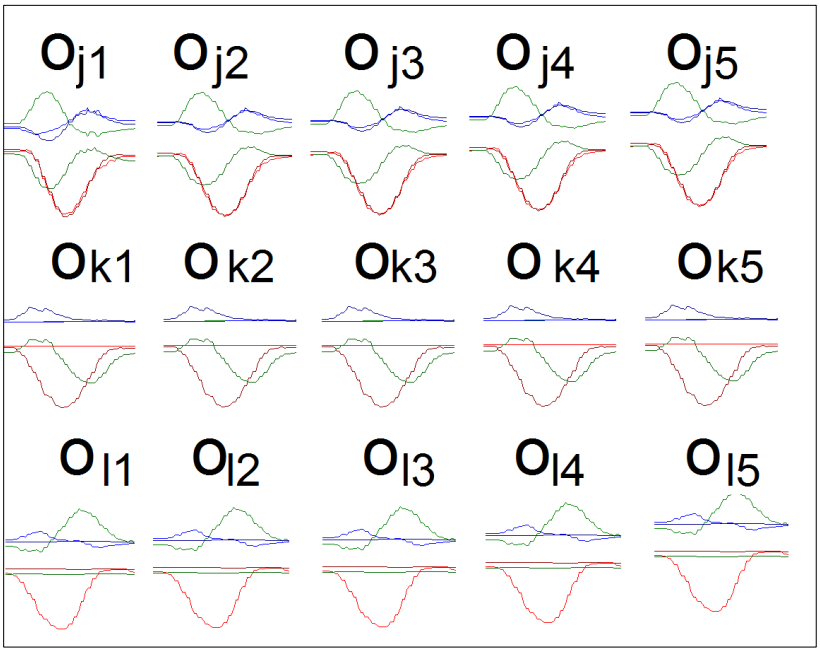

Figure 23 - Sequences of movements from 3 different classes.

We use for each movement a continuous probability density function. A multivariate Gaussian distribution [62] ℵ with zero mean vector ($\mu_{jm} = 0$) and identity covariance matrix ($\Sigma_{jm} = \mathrm{I}$) . In this case we specify the parameters of the probability density function. The probability density is approximated by a weighted sum of *M* Gaussian distributions, from 5.3.5,

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm}\, \aleph(\mu_{jm}, \Sigma_{jm}, o_t)$$

This way we get a sequence ($\lambda_1, \lambda_2, \dots, \lambda_{|C|}$) of HMM's $\lambda = (A, c_{jm}, \mu_{jm}, \Sigma_{jm}, \pi)$ that is the *Sequence Classifier* for our problem.

b) Let's consider a movement

$O = \{v(t)\} = \{(kx1(t), ky1(t), kz1(t), kx2(t), ky2(t)k, z2(t))\}$ (*Figure 18*)

Then the *forward algorithm* is used to calculate the best probability of the model λ_i that has most likely generated the sequence O.

5.3.6 Summary

In this subsection, we have explained the way we have addressed the problem of Isolated Moving Gestures and hand kinematics. First we have defined and explained what *movements* mean in the context of this work. Some assumptions and preparations steps were set and a detailed theoretical formulation and explanation of the algorithms to solve our problem of Isolated Moving Gestures Recognition.

5.4 Conclusions

In this chapter we have introduced and explained in detail, our approach and methodologies to solve the stated thesis problems. We have started by describing our work on Depth Sensor Signal Processing and Feature Extraction. Then we have detailed our novel approach to Isolated Static Gestures and presented the respective algorithm. Finally, we made a description of the algorithms that provide automatic Recognition of Isolated Moving Gestures.

Rules are just helpful guidelines for stupid people who can't make up their own minds.

Seth Hoffman

6. Results and Evaluation

In this chapter we explain the training data collecting and storing process of the algorithms for Isolated Static Gestures – "2D Distance from Axis Analysis" and "Skeletal-based Template Matching" and the algorithms for Isolated Moving Gestures – "3D Path Analysis" and "Hidden Markov Model". Then we explain the details of the performed tests on each algorithm, and from the presentation and analysis of the collected data, we draw a discussion about the obtained results of each of the algorithms.

6.1 Producing and Storing Sample Data for Training Sets

The "PostureDatabase" (*Figure 24*) and "MovementSetup" (*Figure 25*) components have the role of collecting and managing the gesture database and sample data (known as the training set) for the learning and classification process. It has a set of functions to capture, process, rename, delete and manually classify gestures issued by a subject in an off-line mode.

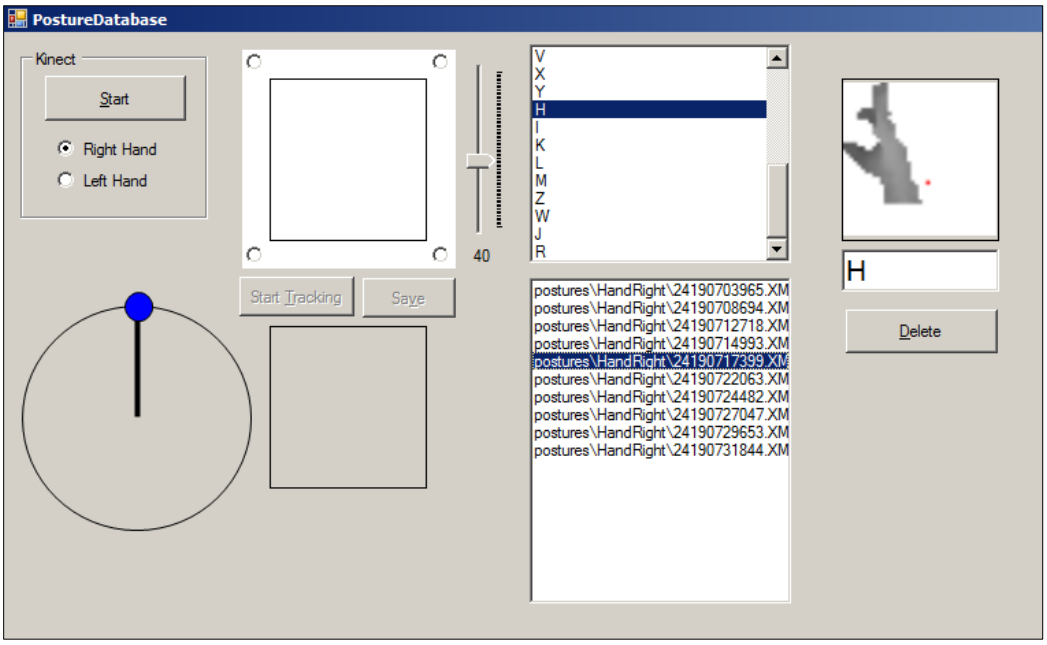

Figure 24 - Posture management screen

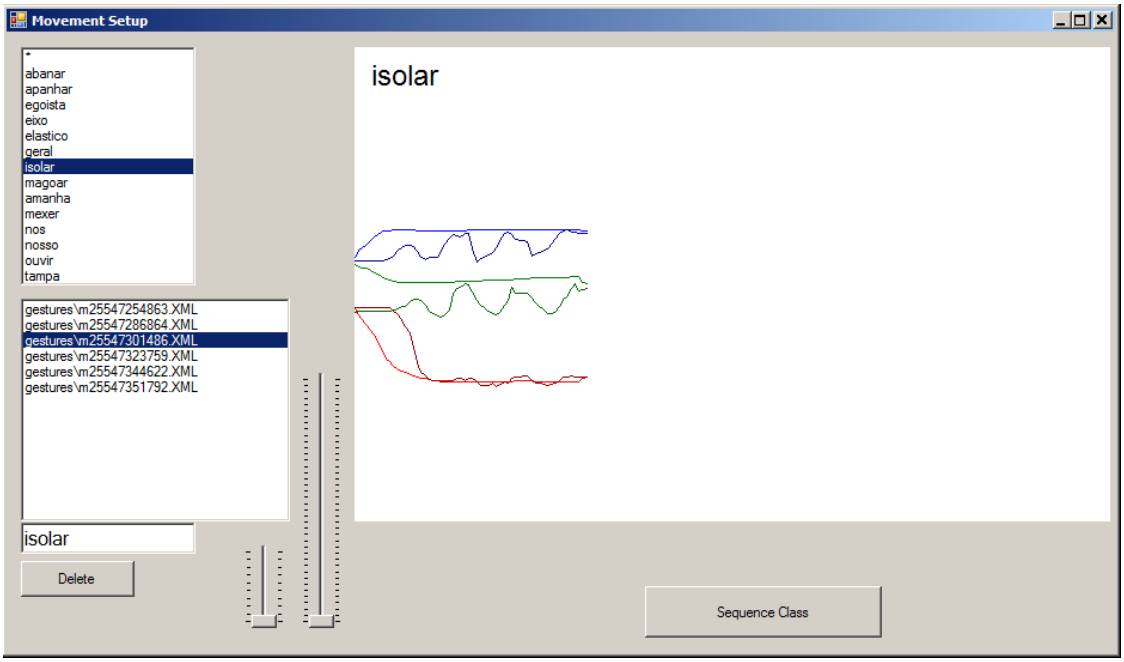

Figure 25 - Movement management screen

6.2 Isolated Static Gesture Results

We have made comparable performance and accuracy tests, for both Isolated Static Gesture algorithms (**Method of 2D Distances** and **Skeletal-based Template Matching Adaptation**). For that purpose we have created a training set, where the author of the

thesis performed in off-line mode, each of the 26 letters of the alphabet spelled in the Portuguese Sign Language (*Figure 26*), for about 31 seconds.

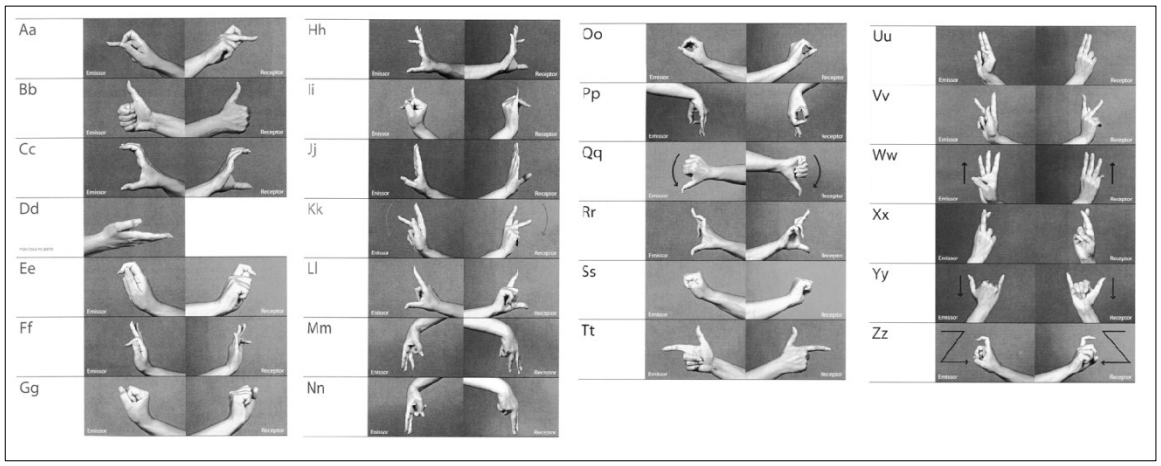

Figure 26 - Static gestures of 26 finger-spelled letters using the PSL alphabet (Images from "Dicionário da Língua Gestual Portuguesa" **[10]**.

In the experiment, the subject maintained the same pose for 31 seconds trying to make the algorithm recognize the pose for as much time as possible. The counting started after 10 seconds of preparation for each gesture. The subject was standing approximately 1.5 meters away in front of the sensor and the sensor is positioned approximately 1 meter above the ground. The produced training set had exactly 260 samples, 10 samples for each symbol, collected with no restrictions in the range of sensor field-of-view.

In the run-time gesture recognition process, with the same conditions as when the training set was created (user standing approximately 1.5 meters away in front of the sensor and the sensor positioned approximately 1 meter above the ground), the system was continually filling a stack of results and the final result presented as valid was the most common result in the last 7 recognition outputs[11] from the stack. For each symbol, we have recorded the number of recognitions per second and the corresponding recognized letter.

[11] We could try a different value, but 7 seemed empirically to be a good one.

6.2.1 Results for Algorithm 1 – 2D Distance from Axis Analysis (not extended to depth)[12]

In *Table 1* we can see "*AvGRecs"* representing the average number of executions of the algorithm per second for the corresponding letter, during the 31 seconds experiment. "*Letter Recognition Rate – LRR"* is the rate of positive recognitions of PSL letters, "*Fails"* represents the number of seconds[13] failed out of 31 seconds and *"FalsePositives"* represents the number of false positives, in which the algorithm returned as a result one alphabet letter different from the expected. No other experiment was done with a training set and/or a test set created by other user.

[12] See 5.2.4 $POLC$

[13] Not necessarily continuous.

PSL Letter	AvGRecs	LRR	Fails	FalsePositives
A	20,65	1,00	0	0
B	26,848	1,00	0	0
C	27,584	1,00	0	0
D	28,774	1,00	0	0
E	24,13	0,95	0	38
F	28,35	1,00	0	0
G	32,77	0,97	0	30
H	27,06	0,88	0	102
I	20,678	1,00	0	0
J	28,26	0,96	1	31
K	28,58	0,84	0	154
L	22,29	1,00	0	0
M	24,10	0,70	1	245
N	24,16	0,96	0	27
O	33,10	0,93	0	76
P	30,26	1,00	0	0
Q	17,16	1,00	0	0
R	18,94	0,64	2	211
S	24,93	0,69	1	223
T	22,26	1,00	0	0
U	20,23	0,85	0	111
V	26,00	0,94	0	47
W	23,16	0,68	1	235
X	39,03	0,70	0	371
Y	29,67	0,67	1	302
Z	24,71	0,87	0	109

Table 1- Results from testing Static Isolated Gestures on2D Distance from Axis Analysis (not extended to depth).

The achieved average of calculations per second was 25,91 and the global average LRR rate obtained is 89%. We had 7 fails, where the system was unable to recognize for 1 entire second and we had 2312 false positives out of 20884 recognitions. The set {A,B,C,D,F,I,L,P,Q,T} contains the symbols rated with best results, having an overall LRR=100%, and ‘ R’ was the symbol with the worst results with an LRR = 64% and 2 fails.

6.2.2 Discussion

The results compare well with prior work [38], where static hand pose recognition algorithms, based in different Computer Vision techniques, were tested for all 36 letters and numbers of the alphabet, spelled using PSL. In this prior work, each hand pose was tested ten times against a user trainnig set library (that is, the user performing the gestures was the same user whose hand poses have been stored in the library), and 7 different techniques were tested. Template Matching, DCT and PGH were, significantly, the best methods studied for robust shape recognition (see table below). However our current technique obtains an average LRR of 89%. across 26 Portuguese letters against, 53,6%, 50,6% and 58,1%, respectively, for Template Matching, DCT and PGH. Our technique thus brings a substantial improvement in the SOA for spelled alphabet letters recognition using the PSL language.

Algorithms	LRR (%) User Test Set
Template Matching	53.6
DCT – Discrete Cosine Transform	50.6
Hu Moments	25.8
PGH -Pair-wise Geometrical Histogram	58.1
SSD - Simple Shape Descriptors	5.8
PGH-SSD Hybrid	21.9

Table 2 - Testing Static Hand Poses with different algorithms for all 36 alphabet symbols spelled in PSL, as described in [38]

6.2.3 Results for Algorithm 2 – Skeletal-based Template Matching adaptation

Table 3 represents the results for this second algorithm, with the same training set of the first algorithm and the same experimental conditions. The results are depicted in the table below.

PSL Letter	AvGRecs	LRR	Fails	FalsePositives
A	22,68	1,00	0	0
B	34,84	1,00	0	0
C	31,48	1,00	0	0
D	31,81	1,00	0	0
E	29,39	1,00	0	0
F	24,19	1,00	0	0
G	22,87	1,00	0	0
H	31,48	1,00	0	0
I	33,35	1,00	0	0
J	26,00	1,00	0	0
K	32,77	1,00	0	0
L	35,10	1,00	0	0
M	31,84	1,00	0	0
N	24,10	1,00	0	0
O	25,65	1,00	0	0
P	26,55	1,00	0	0
Q	28,35	1,00	0	0
R	23,16	1,00	0	0
S	26,39	1,00	0	0
T	26,97	1,00	0	0
U	30,42	1,00	0	0
V	26,32	1,00	0	0
W	31,55	1,00	0	0
X	22,90	1,00	0	0
Y	37,84	1,00	0	0
Z	34,26	1,00	0	0

Table 3 - Results from testing Static Isolated Gestures on Skeletal-based Template Matching algorithm.

The average of calculations per second is 28,93 and the global average recognition rate achieved is 100%. We had no fails, and the system was able to perform successful recognitions for every second and we had no false positives out of 23320 recognitions.

Making a comparison between the two algorithms we can easily verify that the "Skeletal-based Template Matching", achieves better results , in terms of Letter *Recognition Rate* with 100% against the 89% LRR of the "*Distance from Axis Analysis*" and also at the level of performance with 28,93 average of calculations per second against 25,91. As

with the other presented technique, no other experiment was done with a training set and/or a test set created by other user.

6.2.4 Discussion

Analyzing traditional approaches using Computer Vision in the *State of the Art,* for PSL and other Sign Languages, we can conclude those methods present recognition rates from 50% to 90%. With an obtained figure of 100% in overall recognition rate, we can say that our approach using Depth Sensors is a promising contribution. It´s worth referring, that for both techniques our obtained LRR are quite high and, as mentioned, compare very well with the literature. However, more experiments need to be done with more variety on both training and test sets, to further conclude on the repeatability of these high figures of LRR.

6.3 Isolated Moving Gestures Results

To test the Moving Gesture Recognition algorithms we have prepared a training set and a test set, for both. The training set was composed of 14 different random words selected from the dictionary of Portuguese Sign Language (*Figure 27*). For each word the author recorded 6 samples, resulting in a final training set database of 84 samples. The test set consisted in repeating the same gesture 10 times. During the experiments, we have recorded the response of the system (see *Table 4*). *Recs* is the number of recognitions of PSL gestured words, "*Fails"* represents the number of false positives, in which the algorithm returned as a result a PSL word different from the expected.

Figure 27 - Gestures used to test movement recognition algorithms

6.3.1 Results for Algorithm 1 – 3D Path Analysis

For the first algorithm, in all cases, the correct 2 hands movement has always been recognized for the entire sequence of 10 repetitions, even for the case of similar gestures mentioned above (Table 4).

Movement	Recs	Fails
abanar	10	0
apanhar	10	0
egoista	10	0
eixo	10	0
elástico	10	0
nosso	10	0
ouvir	10	0
geral	10	0
isolar	10	0
magoar	10	0
amanhã	10	0
mexer	10	0
nós	10	0
tampa	10	0

Table 4 - Results from algorithm 1 - 3D Path Analysis

6.3.2 Discussion

From *Figure 27* we can easily observe that the 6 lines from *x*, *y*, and *z* coordinates from hand position show substantially different shapes from each other. In the case of this algorithm that analytically examines and compares the distances between the lines, it must justify the fact that for this limited set of movements, the differences between the lines will be found and easily distinguish the movements between them. Even in the case of the words “nós", and “nosso” where the movements are very similar, with very few differences between them, the algorithm was able to distinguish the small differences. Larger training and test sets should be considered in order to obtain results about robustness of the technique in terms of performance and outcome.

6.3.3 Results for Algorithm 2 - Hidden Markov Model in Moving Gestures Recognition

For the second algorithm, we have noticed a wider range of results. There are cases where the movement was identified in all the 10 repetitions. There are also cases in which the movement was not recognized (2,5,7 and 11) in any one of the 10 repetitions. There are also gestures from which the movement recognized as a movement from another gesture

(2,5,6,7,10 and 13) and even cases (7,10, 11 and 13) in which no movement was recognized .

*Table **5*** shows the overall summary of the results. ***Recs*** indicate the number of successful recognitions and ***Fails*** is the number of times, the recognition failed out of 10 repetitions.

Movement	Recs	Fails
1-abanar	10	0
2-apanhar	0	10
3-egoista	10	0
4-eixo	10	0
5-elástico	0	10
6-nosso	8	2
7-ouvir	0	10
8-geral	10	0
9-isolar	10	0
10-magoar	4	5
11-amanhã	0	10
12-mexer	10	0
13-nós	5	5
14-tampa	3	7

Table 5 - Overall results from algorithm 2 - HMM in Moving Gesture Recognition

Movement	1	2	3	4	5	6	7	8	9	10	11	12	13	14	None
1	10														
2		0				2					3	2	3		
3			10												
4				10											
5				10	0										
6						8							2		
7							0			8					2
8								10							
9									10						
10				1					2	4					2
11											0	6			4
12												10			
13						4							5		1
14									7					3	

Table 6 - Cross results between target movement (lines) and recognized movement (columns)

Table 6 shows the intersection between target movement and recognized movement. The first column on the left represents the real-time performed movement and first row on top represents the same 14 movements. In the column below each movement, we depict the number of times that each one of the movements on the left was recognized as the movement of that corresponding column. The sum of numbers in each row always totals 10, which was the number of times that each movement was repeated.

6.3.4 Discussion

Differently from the previous algorithm that uses an analytical method, Hidden Markov models (HMM) are stochastic processes where probabilities are considered to get final recognition likelihoods. We have used the *Ergodic* topology in 2 states. We can see from Table 6 that gestures having similarities are not clearly distinguished among them, like in the case of "nós" and "nosso". Unlike the analytical process where the different individual distances are considered, in this case the 6-dimensional signal is considered as a whole in the unobservable states. This fact could also explain the not so good distinction between movements, contrary to the first algorithm (3D Path Analysis).

The following table shows the results of the the related work titeled, "A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition" [46] - An automatic system that recognizes isolated meaningful gesture from hand motion for Arabic numbers from 0 to 9 in real-time based on Hidden Markov Models (HMM). In order to handle isolated gesture, HMM using *Ergodic*, *Left-Right* (LR) and *Left-Right Banded* (LRB) topologies is applied over the discrete vector feature that is extracted from stereo. Unlike our work, this study does not handle 3D arbitrary movements of sign language gestures, but only isolated or continuous 2D planar paths, based on the centroid points of the hands. However, it is important to analyze these results, since different topologies have been studied and the results indicate that the *Ergodic* topology we have tested in our work had the worst results. This analysis brings therefore good guidelines for our future approaches.

Experimental results show that the proposed system can successfully recognize isolated and meaningful gesture and achieve average rate recognition 98.6% and 94.29% respectively.

Number of states	Data		Topologies Recognition (%)		
	Train	Test	Ergodic	LR	LRB
3	200	98	70,40	100,00	100,00
4	200	98	58,16	100,00	100,00
5	200	98	65,31	98,98	100,00
6	200	98	59,18	96,94	96,94
7	200	98	45,92	96,94	97,96
8	200	98	46,94	94,90	96,94
9	200	98	53,06	94,90	97,96
10	200	98	44,90	96,94	98,98
Average	200	98	55,48	97,45	98,60

Table 7- Results of the related work , "A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition" [46]

Left-Right Banded (LRB) and Left-Right (LR) show better results when compared with the Ergodic topology. We can also observe better results in models having a smaller number of states and as the number of states increases the worse the recognition rate is.
A *Forward Left-Right* (LR) and a *Left-Right Banded* (LRB) topology [46] having 2 or more states should be then considered in future approaches. Different and wider training and test sets, covering a larger audience of subjects, should be used for future tests.

6.4 Conclusions

For the "Method of 2D Distances" technique, we have measured the average number of executions of the algorithm per second for the corresponding letter, during a 31 seconds experiment and we have observed a value of 25,91. We have also measured the rate of positive recognitions of PSL letters (LRR) and the global average LRR rate obtained was 89%. Finally we have recorded the number of "Fails" that represents the number of seconds failed, out of 31 seconds and "FalsePositives", that represents the number of false positives, in which the algorithm returned as a result one alphabet letter different from the expected. We had 7 fails, where the system was unable to recognize for 1 entire second and we had 2312 false positives out of 20884 recognitions. The set {A,B,C,D,F,I,L,P,Q,T} contains the symbols rated with best results, having an overall LRR=100%, and ' R' was the symbol with the worst results with an LRR = 64% and 2 fails.

For the "Skeletal-based Template Matching", we have observed an average of calculations per second of 28,93 and the global average recognition rate achieved 100%. We had no fails, and the system was able to perform successful recognitions for every second with no false positives out of 23320 recognitions.

By comparing the two algorithms, we have verified that the "Skeletal-based Template Matching", achieves better results , with 100% against the 89% LRR of the "Distance from Axis Analysis" and also at the level of performance, with 28,93 average of calculations per second against 25,91. Analyzing traditional approaches using Computer Vision in the State of the Art, for PSL and other Sign Languages, we can conclude those methods present recognition rates from 50% to 90%. With an obtained figure of 100% in overall recognition rate, we can say that our approach using Depth Sensors is a promising contribution. For both techniques our obtained LRR are quite high and compare very well with the literature. However, more experiments need to be done with more subject variety on both training and test sets, to further conclude on the repeatability of these high figures of LRR.

Our technique on Isolated Moving Gestures, particularly Hand Movements was also presented in detail. Our proposed innovative technique "3D Path Analysis" and an adaptation of Hidden Markov Models (HMM) were formalized and explained and the corresponding experimental results were given. To analyze these Moving Gesture Recognition algorithms, we have prepared a training set and a test set, for both. The training set was composed of 14 different random words selected from the dictionary of Portuguese Sign Language (*Figure 27*). In the case of the words "nós", and "nosso" the movements are very similar with very few differences between them. For each word the author recorded 6 samples, resulting in a final training set database of 84 samples. The test set consisted in repeating the same gesture 10 times. During the experiments, we have recorded the response of the system (see Table 4).

For the "3D Path Analysis" algorithm, in all cases, the correct 2 hands movement has always been recognized for the entire sequence of 10 repetitions, even for the case of similar gestures mentioned above (see Table 4). For the "Hidden Markov Model in Moving Gestures" algorithm, we have noticed a wider range of results. There are cases where the movement was identified in all the 10 repetitions. There are also cases in which

the movement was not recognized in any one of the 10 repetitions. There were also gestures from which the movement was recognized as a movement from another gesture, and even cases in which no movement was recognized. Table 5 shows the overall summary of the results and the Table 6 shows the intersection between target movement and recognized movement. We have concluded that "3D Path Analysis", presents more promising results than Hidden Markov Model in Moving Gestures" using an Ergodic topology.

7. Conclusions

It is hard enough to remember my opinions, without also remembering my reasons for them!

Friedrich Nietzsche

7.1 Thesis hypothesis and goals

According to the three stated hypothesis of Portuguese Sign Language Adoption, Gesture Modeling and Recognition in 3D and Depth Sensor Application, as well as the thesis objectives aiming at demonstrating the validity of the mentioned hypothesis, as described in chapter 1, we can enumerate the following thesis achievements and contributions:

Portuguese Sign Language Adoption:

1. We have proposed a proper Gesture Taxonomy adapted to "Portuguese Sign Language Recognition".
2. We have proposed an architecture and demonstrated that it is able to solve some simple and limited of problems of "Portuguese Sign Language Recognition". We believe that such architecture is also applicable to Sign Languages other than PSL, as well as for generic 3D gesture recognition, other than the recognition of signs from Sign Languages. Additionally, we believe that the architecture has the potential to be extended, with further research, to other gesture problems, not yet tackled in this thesis, such as co-articulation of Signs in PSL and other sign languages, facial gestures and expressions, emotions and general moving body gestures.

Gesture Modeling and Recognition in 3D:

1. From a whole range of techniques commonly used in both Speech Recognition and Vision-based Gesture Recognition, we have selected "Hidden Markov Models" and "Template Matching" as technical foundations and have adapted those, to create our own novel algorithms for PSL Recognition.
2. We have created a framework and respective classes for 3D gesture representation, feature extraction, learning, training set database management and real-time automatic gesture recognition.

3. We have developed and tested some techniques to support "*Isolated Static Gestures and Hand Postures*" and "*Isolated Moving Gestures and Hand Movements*", which are two gesture recognition modalities of our architecture. For the first modality, innovative two algorithms were created, namely, *"2D Distance from Axis Analysis"* and "Skeletal-based Template Matching", whereas for the second, *""3D Path Analysis"* is novel and we also developed an application of the *Hidden Markov Model* technique.
4. We have created and tested a real-time speed quantifier based on the quantity of the motion, to classify isolated from moving gestures.

Depth Sensor Application:

1. We have adopted some techniques used in Computer Vision, such as 3D Calibration and Background Subtraction to be used with depth sensors, and presented the theoretical and practical details of our implementation, adapted to simple cases of PSL Recognition
2. We have used exclusively Depth Information from our sensor, to solve our problems and the results have demonstrated that this sensed information is enough to automatically recognize (in real-time) both static postures and moving gestures.

7.2 Results

For both classes of gestures (isolated static gesture and isolated moving gesture), we have specified, developed, tested and evaluated four algorithms:

Algorithms 1 and 2 for "**Isolated Static Gestures**"[14]:

- Normalization:

We have developed the methods and classes to represent and store elements that we refer to as "*postures*" (see Figure 5). Postures are sets of depth information captured in a cubic space of points in skeleton space, (x_0,x_1,x_2) with -n /2< x_i< n/2 . We register this information converted to depth space in a scaled n × n depth information matrix around

[14] See 5.2 Isolated Static Gestures

the center. *Postures* are collections of what we refer to as *dexels* (an abbreviation for depth picture elements) and can be grouped in lists of homogeneous postures called *postureLists*. For example, a *postureList* can contain a group of postures representing the same static symbol or letter in sign language.
To accelerate the Template Matching correlation, reducing the effects of noisy data and also reducing the effects of "trembling" skeleton joint positions from NUI Skeleton API, we have created a method that we refer to as Corner Matching. It allows processing different hands and different positions using the appropriate or best matching corners, moving the *dexels* to those corners, generating target postures (postures translated to the appropriate corner of the cubic space). This method adjusts the *posture* to the appropriate 2D (x, y) corners.

We have solved the problem of Scaling Invariance by using a method, to convert skeleton data scale into depth image scale. This way we can get similar posture sizes independently from the distance from the user to the sensor.

We have solved misalignments in size of posture dimension and depth information, caused by the trembling observed in skeleton joint positions and differences in distances to the sensor. To make this depth alignment, we used the joint position to make the translation of *dexels* in the z axis direction in order to standardize all of the postures.

- *"2D Distance from Axis Analysis"*

We have created this novel algorithm using features of orthogonal linear contours. We have proposed a new mathematical technique, referred to as the "*Principal Orthogonal Linear Contours – POLC"*. This technique uses maximization and minimization of values from depth data information (Figure 10). The orthogonal contours of the posture are extracted and used to check for the best match among training data on *PostureLists*. The PostureList that minimizes POLC represents the most likely gesture for recognition process.

- "Skeletal-based Template Matching"

We have created this algorithm based on the sum of absolute differences from current posture to each posture from the *PostureList* (from the training database).
The method is based on Euclidean Distances and uses three steps, namely, *CompareAvgDistance*, *CountXYInOut* and *CountXYOutIn*:
CompareAvgDistance - CAD computes the average distances of dexels having corresponding matches in both postures.
CountXYInOut - CXIO counts the number dexels in source posture not having corresponding dexels in target posture.
CountXYOutIn - CXOI counts the number dexels in target posture not having corresponding dexels in source posture.
One final step gets the "best matching" PostureList in real time by finding the solution that minimizes the algorithm main formula.

Algorithms 3 and 4 for "**Isolated Moving Gestures and Hand Kinematics**"[15]:

- Normalization:

Based on human skeleton joint positions provided by the NUI Skeleton API [59], we have created a method to represent elements that we have named "movements". Movements are time sequences of feature vectors representing the hands and head laws of motion over time (kinematics), defined in the skeleton reference frame which is centered in the head. The method records the head position and uses the continuous kinematic laws of left hand and right hand 3D points obtained directly from the NUI Skeleton API [59]. This gives us a 6-dimensional feature vector (Figure 15), representing x, y, z coordinates.
Movements are collections of *TimeDexels* and can be grouped in lists of homogeneous movements called *MovementLists*. For example, a *MovementList* can contain a group of movements representing the same moving gesture in a sign language.

[15] See 5.3 Isolated Moving Gestures and Hand Kinematics

We have created a "movement segmentation method" based on the global speed of movement parameter, observed in real time. We have assumed a threshold for this parameter, above which the application determines that a movement corresponding to a gesture is being made.

We have developed a pre-processing "time scaling method" to transform the movement sequence into an equivalent sequence with a different time interval, maintaining the proportions over time. With this scheme, every *MovementList* can be scaled to have all the movements with the same time duration (Figure 18).

- *"3D Path Analysis"*

We have created this algorithm in which we consider sets of Movements from the same class (same gesture), and we compute the distances between the real time performed gestures and the movements from that same classes (same gestures). Then we apply a "distance minimization formula" and, when it is fulfilled, the performed gesture most likely belongs to the same class of the corresponding *MovementList*.

- *"Hidden Markov Model variant"*

We have adapted the well-known *Hidden Markov Model - HMM* methods to our *Moving Gesture Recognition*. In our problem solving, we consider 6-dimensional continuous sequences of Movements, and using for each movement a continuous probability density function (Gaussian distribution)[62] , with zero mean vector and identity covariance matrix, we apply the Baum-Welch learning process in order to get the *Sequence Classifier* HMM's for our problem.
Then, for a real time *Movement*, the forward algorithm is used to calculate the best probability of the model that has most likely generated the *Movement* sequence.

For the test cases that we have described in the corresponding chapter[16], at least one algorithm for each class reached 100% the goal of the corresponding test case (see sections section 5.3.2 and 5.3.3), namely "*The Method of 2D Distances* " for posture recognition and "*3D Path Analysis*" for moving gesture, more precisely, for movement recognition.

However, in all experiments both the training sets and the test sets were created by the same author. Therefore, more tests need to be done with more user variety on both training and test sets, to further conclude on the repeatability of these high figures of LRR.

As it can be seen in the experimental results presented in section 5.3.2 and 5.3.3, the algorithms have a good for real-time performance, with little learning effort.

7.3 Future Work

We can identify several topics we would like to see covered in future research for automatic gesture recognition and its application to Sign Language, including PSL:

a) Longer sets of training data must be created, with larger gesture ranges issued by a larger variety of users, including numbers and more signs of "Portuguese Sign Language".
b) Standard test cases must be designed and a set of stable test sets needs to be created., with signs issued by a large variety of users, different from the ones that created the training sets.
c) A rotation invariance feature in gesture pre-processing for training and testing, should be developed.
d) Extend the concept of *posture*[17] to include "both hands", "facial expression", "both hands and face", "full body", etc.
e) At least 14 facial expressions and lip movements take part in PSL [10]. Facial Recognition techniques must be investigated and applied to improve the automatic PSL recognition process.

[16] See 4.2. and 4.3 for static or moving gestures, respectively.
[17] See 2.2 Hand Gesture Taxonomy.

f) This thesis work did not cover the continuous gesture recognition applicable in sign language recognition. However we can formulate some questions that we believe are important to be answered, in order to face and address this complex problem:
   - How is information encoded in continuous gestures[18]? To answer this question, it would be useful to know how scientists coming from different disciplines, like biologists, sociologists, psychologists and others define gesture.
   - What body parts are involved? Hands, arms, facial expressions[19]?
   - Which target user group to consider? Individual or multiple persons or specific groups with impairments, like deaf and dumb citizens?
   - What are the environment conditions and what are the restrictions to consider? Indoor/outdoor scenarios? Moving/static background?
   - After having answers to these questions we think we will be able to create the necessary knowledge to frame the problem. A possible strategy could be to consider a multimodal approach[20] using postures, movements and facial and body features.

g) From c), d), e) and f) together with a solution for "movement epenthesis ", we could address the problem of continuous gesture recognition (*Figure 1*), in particular, for full "Sign Language Recognition", including PSL.

h) Address the problem of modeling, representing and synthetizing PSL gestures, using a 3D avatar.

[18] For the case of Static and Moving Isolated Gestures , see 5.3
[19] Define Gesture (see 2.1)
[20] See 4. System Architecture and Software Design.

Appendix A. Hidden Markov Models

A.1 Definition of HMM

The Hidden Markov Model is a stochastic process with a finite set of *states*, each associated with a probability distribution, multidimensional or not. Transitions among the states are defined by a set of probabilities called *transition probabilities.* In a particular state an outcome or *observation* can be generated, according to the associated probability distribution. The name Hidden Markov Model comes from the fact that only the outcome is visible to an external observer and not the state itself. The states are "hidden" to the outside.

To define an HMM we need the following elements [61]:

- The number of states of the model, *N*.
- T is the length of the observations sequence(number of clock times)
- $\mathrm{Q} = \{q_1, q_2, \dots q_M\}$, the set of states
- A set of state transition probabilities $A = \{a_{ij}\}$ called state transition probability distribution.

  $$a_{ij} = p\{q_{t+1} = j | q_t = i\}, 1 \leq i, j \leq N, \quad (1)$$

  where q_t denotes the current state.
  Transition probabilities should satisfy the normal stochastic constraints,

  $$a_{ij} \geq 0, 1 \leq i, j \leq N$$

  and

  $$\sum_{j=1}^{N} a_{ij} = 1, 1 \leq i \leq N \quad (2)$$

- The number of observation symbols in the alphabet, *M*. For continuous signs observations *M* is infinite.

- V $= \{v_1, v_2, \ldots v_M\}$ a discrete set of possible symbol observations.

- A probability distribution in each of the states, $B = \{b_j(k)\}$, called observation symbol probability distribution in state j.

$$b_j(k) = p\{o_t = v_k | q_t = j\},\ 1 \leq j \leq N,\ 1 \leq k \leq M \quad (3)$$

Where v_k denotes the k^{th} observation symbol in the alphabet, and o_t the current parameter vector.
Following stochastic constraints must be satisfied.

$$b_j(k) \geq 0\ , 1\ \leq j \leq N, 1\ \leq k\ \leq M$$

and

$$\sum_{j=1}^{N} b_j(k) = 1, 1 \leq j \leq N \quad (4)$$

If the observations are continuous then we will have to use a continuous probability density function, instead of a set of discrete probabilities. In this case we specify the parameters of the probability density function. Usually the probability density is approximated by a weighted sum of *M* Gaussian distributions ℵ.

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm}\, \aleph(\mu_{jm}, \Sigma_{jm}, o_t) \quad (5)$$

where,

$c_{jm} =$ weghting coefficients

$\mu_{jm} =$ mean vectors

$\Sigma_{jm} =$ covariance matrices

c_{jm} should satisfy the stochastic constrains,

$c_{jm} \geq 0, 1 \leq j \leq N, 1 \leq m \leq M$

and

$$\sum_{m=1}^{M} c_{jm} = 1, 1 \leq j \leq N \quad (6)$$

1. The initial state distribution, $\pi = \{\pi_i\}$

   where,

   $\pi_i = p\{q_1 = i\}, 1 \leq i \leq N$.

An HMM with discrete probability distributions can be defined by

$\lambda = (A, B, \pi)$ (7)

For the continuous density probability distributions we can denote it by

$\lambda = (A, c_{jm}, \mu_{jm}, \Sigma_{jm}, \pi)$ (8)

A.2 Three Canonical Problems

There are three fundamental problems of interest in the Hidden Markov Model design:

Problem one - Evaluation**(**Recognition)

Given the observation sequence $O = \{o_1, o_2, \ldots, o_T\}$ and the model λ = (π, A, B), what is the probability $P\{O|\lambda\}$ that the observations are generated by the model?

Problem two - Decoding

Given the observation sequence $O = \{o_1, o_2, \dots, o_T\}$ and the model λ = (π, A, B), what is the most likely state sequence $q = \{q_1, q_2, \dots, q_T\}$, in the model that produced the chosen observations?

Problem three – Learning

Given a model, λ and a sequence of observations $O = \{o_1, o_2, \dots, o_T\}$ how should we adjust the model parameters (A, B, π), to maximize $P\{O|\lambda\}$?

A.2.1 Evaluation Problem and Forward Algorithm

Having a model $\lambda = (A, B, \pi)$ and a sequence of observations $O = \{o_1, o_2, \dots, o_T\}$, $P\{O|\lambda\}$ must be found. To calculate this value using simple probabilistic arguments, involves number of operations in the order of N^T, that is a computational heavy problem.

For every fixed state sequence $I = I_1, I_2, \dots, I_N$

we have $P\{O|\lambda\} = b_{I_1}(o_1) b_{I_1}(o_2) \dots b_{I_N}(o_T)$ (9)

This is very large even for a moderate length sequence *T*. So we must find another method for this calculation. Fortunately there is one method considerably less complex that makes use an auxiliary variable, $\alpha_t(i)$ called *forward variable*.

The forward variable is defined as the probability of the partial observation sequence, $o_1, o_2, \dots, o_T$ when it terminates at the time t, state *i*. Mathematically, $\alpha_t(i) = p\{o_1, o_2, \dots, o_T, q_t = i|\lambda\}$

Then it is easy to see that following recursive relationship holds.

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij} \ , 1 \le j \le N, 1 \le t \le T-1 \quad (10)$$

where,

$$\alpha_1(j) = \pi_j b_j(o_1), 1 \le j \le N$$

Using this recursion we can calculate

$\alpha_T(i), 1 \leq i \leq N$

and then the required probability is given by,

$$p\{O|\lambda\} = \sum_{i=1}^{N} \alpha_T(i) \tag{11}$$

This method, known as the *forward algorithm* is proportional to N^2T, which is linear in relation to T has lower complexity when compared to the direct calculation mentioned earlier, that had an exponential complexity.

In a similar way we can define the *backward variable* $\beta_t(i)$ as the probability of the partial observation sequence $o_{t+1}, o_{t+2}, \ldots, o_T$, given that the current state is *i*. Mathematically ,

$\beta_t(i) = p\{o_{t+1}, o_{t+2}, \ldots, o_T, q_{t=i}|\lambda\}$

As in the case of $\alpha_t(i)$ a recursive relationship can be used to calculate $\beta_t(i)$ efficiently.

$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij}\, b_j(o_{t+1}), 1 \leq i \leq N, 1 \leq t \leq T-1$ (12)

where,

$\beta_T(i) = 1, 1 \leq i \leq N$

Further we can see that,

$\alpha_t(i)\beta_t(i) = p\{O, q_t = i|\lambda\}, 1 \leq i \leq N, 1 \leq t \leq T$ (13)

This gives another way to calculate $P\{O|\lambda\}$, by using both forward and backward variables as given in the following equation:

$$P\{O|\lambda\} = \sum_{j=1}^{N} p\{O, q_t = i|\lambda\ \} = \sum_{j=1}^{N} \alpha_t(i)\beta_t(i) \quad (14)$$

A.2.2 The Decoding Problem and the Viterbi Algorithm

In this case we want to find the most likely or optimal state sequence for a given sequence of observations, $O = \{o_1, o_2, \dots, o_T\}$ and a model, $\lambda = (A, B, \pi)$

The solution to this problem depends on the way we define "optimal state sequence". One possible criteria is to find the states q_t which are individually most likely for *t=t* and to concatenate all such q_t's. But sometimes this method does not give a physically meaningful state sequence [61]. There is another method called *Viterbi algorithm* which has no such problems. In this method, the whole state sequence with the maximum likelihood is found. In order to facilitate the computation an auxiliary variable is defined as,

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} p\{q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, \dots, o_{t-1}|\lambda\ \} \quad , \quad (15)$$

which gives the highest probability that partial observation sequence and state sequence up to *t=t* can have, when the current state is *i*.
It induces in the following recursive relation:

$$\delta_{t+1}(j) = b_j(o_{t-1})\left[\max_{1\le i\le N} \delta_t(i)a_{ij}\right], 1 \le i \le N, 1 \le t \le T-1 \quad (16)$$

where,

$$\delta_1(j) = \pi_j b_j(o_1),\ 1 \le j \le N \quad (17)$$

So the procedure to find the optimal state sequence starts from calculation of $\delta_T(j)$, $1 \le j \le N$ using recursion in (16), while keeping the "winning state" in the maximum finding operation. Finally the state j^*, is found where

$$j^* = \arg\max_{1\le i\le N} \delta_T(j) \ , \quad (18)$$

and starting from this state, the sequence of states is back-tracked as "winning state" in each. This gives the required set of states.

This whole algorithm can be visualized as a search in a graph whose nodes are formed by the states of the HMM in each of the time instant t ,$1 \leq t \leq T$.

A.2.3 The Learning Problem Maximum Likelihood (ML) criterion/ Baum-Welch Algorithm

Maximum Likelihood (ML) criterion [61]

The ML tries to maximize the probability of a given sequence of observations O^w , belonging to a given class *w*, given the HMM λ_w of the class *w*, with relation to the parameters of the model λ_w . This probability is the total likelihood of the observations and can be expressed mathematically as

$$L_{tot} = p\{O^w|\lambda_w\}, \quad (19)$$

However there is no known way to analytically solve the model $\lambda = (A, B, \pi)$,which maximizes the quantity L_{tot} . But we can choose model parameters such that it is locally maximized, using an iterative procedure, called Baum-*Welch method* described below.

Baum-Welch Algorithm

This method can be derived counting the arguments in the occurrences or using calculus to maximize the auxiliary quantity $P(\lambda, \bar{\lambda}) = \sum_q p\{q|O, \lambda\} \log[p\{q|O, \bar{\lambda}\}]$ over $\bar{\lambda}$.

The *Baum-Welch* algorithm ensures the convergence of these values.

To describe the *Baum-Welch algorithm* (also known as *Forward-Backward algorithm*), we must define two more auxiliary variables, in addition to the forward and backward variables defined in a previous section. These variables can however be expressed in terms of the forward and backward variables.

a) First one of those variables is defined as the probability of being in state *i* at *t*=*t* and in state *j* at *t*=*t*+1. Formally,

$$\varepsilon_t(\mathrm{i},\mathrm{j}) = p\{q_t = i, q_{t+1} = j|O, \lambda\} \quad (20)$$

This is the same as,

$$\varepsilon_t(\mathrm{i},\mathrm{j}) = \frac{p\{q_t=i,q_{t+1}=j|O,\lambda\}}{p\{O,\lambda\}} \tag{21}$$

Using forward and backward variables this can be expressed as,

$$\varepsilon_t(\mathrm{i},\mathrm{j}) = \frac{\alpha_t(i)a_{ij}\beta_{t+1}(j)b_j(o_{t+1})}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}\beta_{t+1}(j)b_j(o_{t+1})} \tag{22}$$

b) The second variable is the a posteriori probability,

$$\gamma_t(\mathrm{i}) = p\{q_t = i|O,\lambda\} \tag{23}$$

that is the probability of being in state *i* at *t=t*, given the observation sequence and the model. In forward and backward variables this can be expressed by,

$$\gamma_t(\mathrm{i}) = \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}\right] \tag{24}$$

One can see that the relationship between $\gamma_t(\mathrm{i})$ and $\varepsilon_t(\mathrm{i},\mathrm{j})$ is given by,

$$\gamma_t(\mathrm{i}) = \sum_{j=1}\varepsilon_t(\mathrm{i},\mathrm{j}),\ 1 \le i \le N, 1 \le t \le M \tag{25}$$

It is now possible to describe the Baum-Welch learning process, where parameters of the HMM are updated in such a way to maximize the quantity, $p\{O,\lambda\}$. Starting from a model $\lambda = (A,B,\pi)$,we calculate the α's and β's using the recursions 12 and 10, and then ε's and γ 's using 22 and 25 And then update the HMM parameters according to the next 3 equations, known as *re-estimation formulas*.

$\bar{\pi}_i = \gamma_1(\mathrm{i}), 1 \le \mathrm{i} \le N$,

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1}\varepsilon_t(\mathrm{i},\mathrm{j})}{\sum_{t=1}^{T-1}\gamma_t(\mathrm{i})}, 1 \le \mathrm{i} \le N, 1 \le \mathrm{j} \le N \tag{26}$$

$$\bar{b}_j(k) = \frac{\sum^{T}_{t=1,\ O_t=V_k} \gamma_t(\mathrm{j})}{\sum_{t=1}^{T} \gamma_t(\mathrm{j})}, 1 \leq \mathrm{j} \leq N, 1 \leq \mathrm{k} \leq N \quad (27)$$

These re-estimation formulas can easily be modified to deal with the continuous density case too.

REFERENCES

[1] A. Sears and J. . A. Jacko, The Human–Computer Interaction Handbook - Second Edition, New York, New York: Human Factors and Ergonomics, 2008.

[2] Oxford University Press, Oxford English Dictionary, Oxford, United Kingdom.

[3] M. Karam and m. c. schraefel, "A taxonomy of Gestures in Human ComputerInteraction," *ACM Transactions on Computer-Human Interactions,* 2005.

[4] Y. Levy and J. C. Schaeffer, Language Competence Across Populations: Toward a Definition of Specific Language Impairment, Y. Levy and J. C. Schaeffer, Eds., Psychology Press; 1 edition, 2002.

[5] R.-H. Liang, "Continuous Gesture Recognition System for Taiwanese Sign Language," National Taiwan University, 1997.

[6] M. V. Lamar, "Hand Gesture Recognition using T-CombNET - A Neural Network Model dedicated to - Temporal Information Processing," Nagoya Institute of Technology, Nagoya, 2001.

[7] A. . F. Bobick, "Movement, activity and action: the role of knowledge in the perception of motion," *Workshop on knowlegge-based vision in man and machine - Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences,* vol. 352, no. 1358, pp. 1257--1265, 1997.

[8] S. K. Liddell and R. E. Johnson, "American Sign Language: The phonological Base" , Sign Language Studies, vol. 64, 1989.

[9] O. Aran, "Vision Based Sign Language Recognition: Modeling and Recognizing Isolated Signs with Manual and Non-Manual Components," Bo¸gazi»ci University, 2008.

[10] A. B. Baltazar, Dicionário da Língua Gestual Portuguesa, Porto: Porto Editora, 2010.

[11] M. Turk and G. Robertson, "Perceptual user interfaces (introduction)," in *Communications of the ACM*, 2000.

[12] . L. Deusdado, "Apoio 3D ao ensino da língua gestual," in *Global Congress on Engineering and Technology Education. Santos, 2005*, 2005.

[13] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. Badler and M. Palmer, "A machine translation system from English to American Sign Language," *Envisioning Machine Translation in the Information Future,* p. 191–193, 2000.

[14] T. e. a. Ringbeck, "A 3D Time Of Flight Camera for Object Detection," PMDTechnologies GmbH, Siegen, Germany, 2010.

[15] B. Moebius, M. Pfennigbauer and J. P. do Carmo, "Imaging Lidar Technology - Development Of A 3D-LIDAR Eelegant Breadboard for Rendezvous and Docking, Test Results, and Prospect to Future Sensor Application," in *ICSO 2010 International Conference on Space Optics*, Rhodes, Greece, 2010.

[16] A. Kolb, E. Barth, R. Koch and R. Larsen, "Time-of-Flight Cameras in Computer Graphics," *COMPUTER GRAPHICS forum,* vol. Volume 29, p. 141–159, 2010.

[17] Acroname Robotics , "Sharp IR Rangers Information," 09 02 2000. [Online]. Available: http://www.acroname.com/robotics/info/articles/sharp/sharp.html. [Accessed 04 09 2011].

[18] D. . A. Forsyth and J. Ponce, Computer Vision - A modern Aproach, Prentice Hall , 2003.

[19] B. K. P. Horn, Robot Vision, Cambridge, MA: The MIT Press, 1986.

[20] © 2011 PrimeSense, Ltd., *PrimeSense®,* 2011.

[21] R. Hartley and A. Zisserman, Multiple view geometry in computer vision, vol. 2nd Edition, Cambridge University Press, 2004.

[22] P. Breuer, C. Eckes and S. Müller, "Hand Gesture Recognition with a novel IR Time-of-Flight Range Camera – A pilot study," in *MIRAGE'07 Proceedings of the 3rd international conference on Computer vision/computer graphics collaboration techniques*, 2007.

[23] E. Kollorz, J. Penne, J. Hornegger and A. Barke, "Gesture recognition with a Time-Of-Flight camera," *International Journal of Intelligent Systems Technologies and Applications,* vol. 5, no. 3/4, pp. 334 - 343, 2008.

[24] L. Vladutu, "Non-rigid Shape Recognition for Sign Language Understanding," *WSEAS TRANSACTIONS on SYSTEMS,* vol. 8, no. 12, pp. 1263-1272, 2009.

[25] J. M. Lien, G. Kurillo and R. Bajcsy, "Skeleton-based data compression for multi-camera tele-immersion system," *Advances in Visual Computing,* p. 714–723, 2007.

[26] J. Kim, S. Mastnik and E. André, "EMG-based hand gesture recognition for realtime biosignal interfacing," in *Proceedings of the 13th international conference on Intelligent user interfaces*, 2008.

[27] R. King, B. . P. Lo, A. Darzi and G.-Z. Yang, "Hand Gesture Recognition with Body Sensor Networks," in *International Workshop on Wearable and Implantable Body Sensor Networks*, London , UK, 2005.

[28] H. Brashear, T. Starner, P. Lukowicz and H. Junker, "Using multiple sensors for mobile sign language recognition," in *Proceedings. Seventh IEEE International Symposium on*, 2003.

[29] T. Westeyn, H. Brashear, A. Atrash and T. Starner, "Georgia Tech Gesture Toolkit: Supporting experiments in gesture recognition," in *Proceedings of the 5th international conference on Multimodal interfaces* , New York, NY, USA, 2003.

[30] C. Zhu and W. Sheng, "Online hand gesture recognition using neural network based segmentation," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, St. Louis, MO , USA, 2009.

[31] R. M. McGuire, J. Hernandez-Rebollar, T. Starner, V. Henderson, H. Brashear and D. S. Ross, "Towards a one-way American sign language translator," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings*, 2004.

[32] A. Pandit, D. Dand, S. Mehta, S. Sabesan and A. Daftery, "A Simple Wearable Hand Gesture Recognition Device Using iMEMS," *IEEE,* pp. 592-597, 2009.

[33] V. S. Kulkarni and S. D. Lokhande, "Appearance Based Recognition of American Sign Language Using Gesture Segmentation," *International Journal on Computer*

Science and Engineering, vol. 2, no. 03, p. 560–565, 2010.

[34] M. Van den Bergh, E. Koller-Meier, F. Bosche and L. Van Gool, "Haarlet-based hand gesture recognition for 3D interaction," in *2009 Workshop on Applications of Computer Vision (WACV)*, 2009.

[35] E. Stergiopoulou and N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique," *Engineering Applications of Artificial Intelligence,* vol. 22, no. 8, p. 1141–1158, 2009.

[36] T. Kapuscinski and M. Wysocki, "Hand Gesture Recognition for a Man-Machine Interaction," in *Proceedings of the Second International Workshop on Robot Motion and Control*, 2001.

[37] B. Zhang, R. Yun and H. Qiu, "Hand Gesture Recognition in Natural State Based on Rotation Invariance and OpenCV Realization," in *Entertainment for Education. Digital Techniques and Systems*, vol. 6249, S. B. Heidelberg, Ed., Berlin, Heidelberg, 2010, pp. 486-496.

[38] M. S. Dias, P. Nande, N. Barata and A. Correia, "OGRE - open gestures recognition engine," in *17th Brazilian Symposium on Computer Graphics and Image Processing, 2004. Proceedings*, 2004.

[39] J. H. Shin, J. S. Lee, S. K. Kil, D. F. Shen, J. G. Ryu, E. H. Lee, H. K. Min and S. H. Hong, "Hand Region Extraction and Gesture Recognition using entropy analysis," *IJCSNS International Journal of Computer Science and 216 Network Security,* vol. 6, no. 2A, p. 216, 2006.

[40] Y. Liu, Z. Gan and Y. Sun, "Static hand gesture recognition and its application based on support vector machines," *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing,* vol. 0, p. 517–521, 2008.

[41] R. Bastos and M. S. Dias, "Skin Color Profile Capture for Scale and Rotation Invariant Hand Gesture Recognition," in *Gesture-Based Human-Computer Interaction and Simulation: 7th International Gesture Workshop, GW 2007, Lisbon, Portugal, May 23-25, 2007, Revised Selected Papers*, vol. 5085, S. Gibet, M. M. Wanderley and R. Bastos, Eds., Berlin, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 81-92.

[42] e. a. Bastos, "FIRST - Fast Invariant to Rotation and Scale feature Transform," 2008.

[43] E. Ueda, Y. Matsumoto, M. Imai and T. Ogasawara, "Hand pose estimation for vision-based human interface," in *Proceedings. 10th IEEE International Workshop on Robot and Human Interactive Communication*, 2001.

[44] S. E. Ghobadi, O. E. Loepprich, F. Ahmadov, J. Bernshausen, K. Hartmann and O. Loffeld, "Real Time Hand Based Robot Control Using Multimodal Images," *IAENG International Journal of Computer Science,* vol. 35, no. 4, p. 500–505, 2008.

[45] B. Ionescu, D. Coquin, P. Lambert and V. Buzuloiu, "Dynamic Hand Gesture Recognition Using the Skeleton of the Hand," *EURASIP Journal on Advances in Signal Processing,* pp. 2101-2109, 2005.

[46] M. Elmezain, A. Al-Hamadi, G. Krell, S. El-Etriby and B. Michaelis, "Gesture

Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models," *IEEE International Symposium on Signal Processing and Information Technology,* no. 3, pp. 1192-1197, 2007.

[47] R. Billon, A. Nédélec and J. Tisseau, "Gesture recognition in flow based on PCA analysis using multiagent system," in *ACE '08 Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, 2008.

[48] F. Dadgostar and A. Sarrafzadeh, "Gesture recognition through angle space," 2008.

[49] M. .. Holte and T. B. Moeslund, "Gesture recognition using a range camera," *Laboratory of Computer Vision and Media Technology. Aalborg University, Aalborg, Denmark, Tech. Rep,* p. 1601–3646, 2007.

[50] P. Breuer, C. Eckes and S. Müller, "Hand gesture recognition with a novel IR time-of-flight range camera–a pilot study," *Computer Vision/Computer Graphics Collaboration Techniques,* p. 247–260, 2007.

[51] M. Elmezain, A. Al-Hamadi, J. Appenrodt and B. Michaelis, "A hidden markov model-based continuous gesture recognition system for hand motion trajectory," *ICPR 2008. 19th International Conference on Pattern Recognition,* p. 1–4, 2008.

[52] S. Eickeler, A. Kosmala and G. Rigoll, "Hidden Markov model based continuous online gesture recognition," in *Proceedings. Fourteenth International Conference on Pattern Recognition*, 1998.

[53] S. F. Wong and R. Cipolla, "Continuous gesture recognition using a sparse bayesian classifier," in *ICPR 2006. 18th International Conference on Pattern Recognition*, 2006.

[54] G. Bailador, D. Roggen, G. Tröster and G. Triviño, "Real time gesture recognition using Continuous Time Recurrent Neural Networks," in *Proceedings of the ICST 2nd international conference on Body area networks*, 2007.

[55] L. . -P. Morency, A. Quattoni and T. Darrell, "Latent-Dynamic Discriminative Models for Continuous Gesture Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, 2007.

[56] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick and A. Pentland, "Invariant features for 3-D gesture recognition," in *1996., Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 1996.

[57] "Camera Calibration and 3d Reconstruction," Willowgarage opencv v2.1 documentation, 2010. [Online]. Available: http://opencv.willowgarage.com/documentation/python/camera_calibration_and_3d_reconstruction.html. [Accessed 13 02 2011].

[58] R. Bastos, R. Afonso and M. S. Dias, "Automatic Camera Pose Initialization, using Scale, Rotation and Luminance Invariant Natural Feature Tracking," *The Journal of WSCG,* vol. 16, p. 97–104, 2008.

[59] R. Microsoft, "Getting Started with the Kinect for Windows SDK Beta," Microsoft Research, 2011.

[60] H. Kim and S. de Araújo, "Grayscale template-matching invariant to rotation,

scale, translation, brightness and contrast," *Advances in Image and Video Technology,* p. 100–113, 2007.

[61] N. D. Warakagoda, "A Hybrid ANN-HMM ASR system with NN based adaptive preprocessing," 1996.

[62] C. Souza, "Accord . NET Framework," 2011. [Online]. Available: http://code.google.com/p/accord/source/browse/trunk/Sources/Accord.Statistics/Distributions/Multivariate/MultivariateNormalDistribution.cs. [Accessed 04 07 2011].

[63] Microsoft, "Microsoft Visual Studio," Microsoft, 2010. [Online]. Available: http://www.microsoft.com/visualstudio/. [Accessed 14 02 2011].

[64] F. Dadgostar, A. . L. C. Barczak and A. Sarrafzadeh, "A Color Hand Gesture Database for Evaluating and Improving Algorithms on Hand Gesture and Posture Recognition," *Res Lett Inf Math Sci,* 2005.

[65] V. Pashaloudi and K. Margaritis, "Feature Extraction and Sign Recognition for Greek Sign Language," in *Artificial Intelligence and Soft Computing - 2003*, Thessaloniki, Greece.

[66] C. Keskin, A. Erkan and L. Akarun, "Real Time Hand Tracking and 3D Gesture Recognition for Interactive Interfaces using HMM," in *INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS*, 2003.

[67] M.-C. Su, "A fuzzy rule-based approach to spatio-temporal hand gesture," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 30, no. 2, pp. 276-281, 2000.

[68] C. Vogler, H. Sun and D. Metaxas, "A framework for motion recognition with applications to American sign language and gait recognition," in *Human Motion, 2000. Proceedings. Workshop on*, 2000.

[69] P. Garg, N. Aggarwal and S. Sofat, "Vision Based Hand Gesture Recognition," *World Academy of Science, Engineering and Technology,* vol. 49, p. 972–977, 972–977.