UNIVERSITY OF COIMBRA

Faculty Of Science And Technology

Department Of Electrical and Computer Engineering

Using Distributed Accelerometers for Gesture Recognition and Visualization

Pedro Emanuel dos Santos Vale Sousa Trindade

A dissertation presented for the degree of

Master of Science in Electrical and Computer Engineering

September, 2010

UNIVERSITY OF COIMBRA

Faculty Of Science And Technology

Department Of Electrical and Computer Engineering

Using Distributed Accelerometers for Gesture Recognition and Visualization

Supervisor

Prof. Doutor Jorge Nuno de Almeida e Sousa Almada Lobo

Juries

Prof. Doutor Jorge Manuel Moreira de Campos Pereira Batista

Prof. Doutor João Pedro de Almeida Barreto

Dissertation presented to the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra in partial fulfillment of the requirements for the Degree of Master of Science in Electrical and Computer Engineering

September, 2010

Acknowledgements

I reserve this space to have a written acknowledgment to my supervisor Prof. Doutor Jorge Lobo for his guidance, patience and for the oportunity to do this thesis. I became a much more mature scientist while working under his guidance. I thank him for that.

I would also like to express my gratitude to Prof. Doutor Jorge Dias for the oportunity of working at Mobile Robotics Lab - ISR where I found the right environment and people to evolve in my scientific path.

Not less important, I am also thankful to my family, friends and colleagues for their neverending support.

I would like to end by thanking Mr. Pat Metheny whose genius inspired me when I most needed.

Abstract

Acceleration information captured from the hand can provide valuable information of its 3D angular pose. From this we can recognize hand gestures and visualize them.

The applications for this technology range from industrial toutchless human-machine interface to hearing impairment where one uses gestures to communicate with another person. In this case it will be possible to have a translation of the gesture to another recognizable form such as a written language.

The use of accelerometer sensors in the hand implies the user to have them somehow worn. The development of silicon chip manufacture allowed these sensors to fit in the top of a nail or implanted in the skin and still wirelessly communicate to a processing unit.

This work demonstrates that it is possible to have gesture recognition from a clutter-free system by wearing very small devices and have them translated by a nearby processing unit.

This work will focus on the processing of the acceleration information. It will be shown the use of methods to estimate hand pose, finger joints' position, and from that recognize gestures.

In this thesis it is also presented a visualization of the angular pose of the hand. This visualization can show a single render of the pose of a recognized gesture or it can provide a simple real-time (low-latency) rendering of the hand pose.

The processing of the acceleration information will use the gravity acceleration as a vertical reference. This will be the core of this thesis work for the recognition and visualization of human gestures.

Key Words: Accelerometers; Hand; Gesture Recognition; Gesture Visualization.

Declaration

The work in this thesis project is based on research carried out at the Mobile Robots Laboratory of ISR (Institute of Systems and Robotics) in Coimbra, Portugal. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter an overview of the document is given and the context of research is specified, giving some outlines for the usage of accelerometers as means of gesture recognition and visualization.

1.1 Context and motivation

Nowadays the skills of communication are vital to the society. Whether in industrial environments or plain social human activities there is a constant need of good communication skills.

In a work environment there may be a need to communicate to a machine in a remote, secure, pratical or non-intrusive manner. Such requirements are achieved when it is possible to represent the normal human activity by a virtual representation. In the particular case of human gestures, it is possible to have a virtual perception of the human gesture and be able to communicate in a noisy, possibly at a long distante or with low visibility environment, as in the case of Figure 1.1 where the airport apron operator needs to signal the aircraft pilots.

When related to human-to-human activities, communication can become complicated when one of the interlocutors does not know the other's language. This is the case when two persons try to communicate and one is hearing impaired and knows sign language, while the other is not and does not know such language. Being possible to intermediate the communication between this two persons allows a great social achievement with an

Figure 1.1: Ilustration of an easily compromised environment where direct human communication can be difficult.

immeasurable value to those who carry that limitation.

Science and Industry are constantly evolving and today it is possible to create a system capable of facilitating communications, as mentioned above, and yet be portable, reliable, eventually self-powered and very simple to use.

The development of silicon chip manufacture enabled the development of low-cost single chip inertial sensors. These sensors can fit in a person's thumbnail or even implanted on the skin. Having them connected to some terminal available to the user, it is possible to have a system available to the user's mobile devices like a telefone, smartphone or an iPad.

1.2 Gestures

People are naturally highly skilled in communication with gestures. It is common to see an oral communication being complemented with gestures to create some sort of visualization of the spoken idea. Or it can be used without oral communication just by signaling something at a distance, like signaling the number of apples to a seller in a noisy supermarket.

Despite our natural use of the gestures in normal social life, it is not often we see applications allowing gestures for controlling objects in a real environment [Baudel, 1993].

1.3 Related work

Several types of gesture capture systems are possible. It can be an optical capture system using its vision for recognizing the configuration of a hand and it can be a hand sensor-based recognition. Based on the extended analysis of [Wang et al., 2009], an overview of these systems is given next.

Color markers. Color markers systems for gesture recognition use color patterns to estimate the pose of the hand. This estimation is obtained with inverse kinematics. It has been demonstrated that it is possible to have a low-cost and effective recognition with this approach yet it obliges the user to have that pattern in the hand. That is not as clutter-free and unobtrusive as the approach proposed in this work where the use of a glove is, ultimatelly unnecessary.

Bare-hand tracking. Bare-hand tracking are an active area of research. These systems rely on edge detection and silhouettes typically and are generally robust to lightning conditions. Reasoning from them involves inference algorithms to search the high-dimensional pose space of the hand. That is computationionally expensive and goes far from realtime and becomes unsuitable for Human-Machine interfaces.

Marker-based motion-capture. Marker-bsed systems involve the use of retro-reflective markers or LED and expensive many-camera setups. Such systems are highly accurate but rely in a an expensive cost for the system. The system proposed in this thesis uses an easy to deploy very low-cost sensor setup.

Data-driven pose estimation. The Data-driven pose estimation makes use of the values from sensors attached to the hand to define the pose of the hand. Such approach allows simple computation processing to estimate the hand pose. This type of system can easily become very intrusive since it needs the user to somehow wear the sensors. In this thesis is shown that only a few sensors are enough, meaning that it will be ultimatelly a clutter-free system.

1.4 The approach followed in this dissertation

This work mainly addresses the problem of recognizing static gestures from the information provided by inertial sensors. By using a triaxial accelerometer in each finger and one on the palm, we can measure the acceleration of gravity in relation to each axis of the sensor. Based on this information we are able to estimate the pose of all the hand joints. This enables us to have a clear representation of the hand.

From the representation we developed an algorithm to recognize the gesture against a pre-defined library of gestures. From this algorithm it will be shown that it is possible to have only one reference gesture in the library and still achieve useful results. This avoids a cluster-based approach for recognition and simplifies the process without compromising the results.

Key contributions:

- A study of the information extracted from hand distributed accelerometers.
- An algorithm to estimate the 3D angular pose of each finger, despite of the unobservance of rotation in the gravities axis.
- A visual represetation in real-time (low latency) and offline of the hand using Python and Blender 3D software package.

1.5 Outline of the document

The first chapter introduces the context and motivation, important advantages of a data-driven approach and some important aspects of the work done throughout the project.

Chapters 2 reviews the most relevant concepts used in this work, namely the human hand and its representation as well as the static and dynamic gestures in sign language. An overview of the Portuguese Sign Language is also presented at this point.

In Chapter 3 an overview of inertial sensors is presented.

Chapter 4 details with the gesture recognition method. This is one of the main goals of this work.

In chapter 5 is where the visualization phase is addressed. It provides information on how the visual representation of the data is built.

The chapter 6 goes specific on the details of the implementation. It provides information regarding the hardware used and the results of the implementation.

The final chapter sums up the work, provides final conclusions and prospects interesting directions for this research.

Chapter 2

Overview of Hand Gestures

Some preliminary concepts on hand gestures are established in this chapter to set the framework to the problems being discussed at a later stage. Firstly, it is addressed the hand representation and its physical properties relevant to this work. In this chapter an analysis of the gestures in the context of sign language is also presented.

2.1 The human hand

The human hand structure can be considered in terms of its bones' associated joints, muscles and skin. This work is focused in bones and its associated joints. It is important to know their properties to understand the behaviour and limitations as described next.

2.1.1 Representation

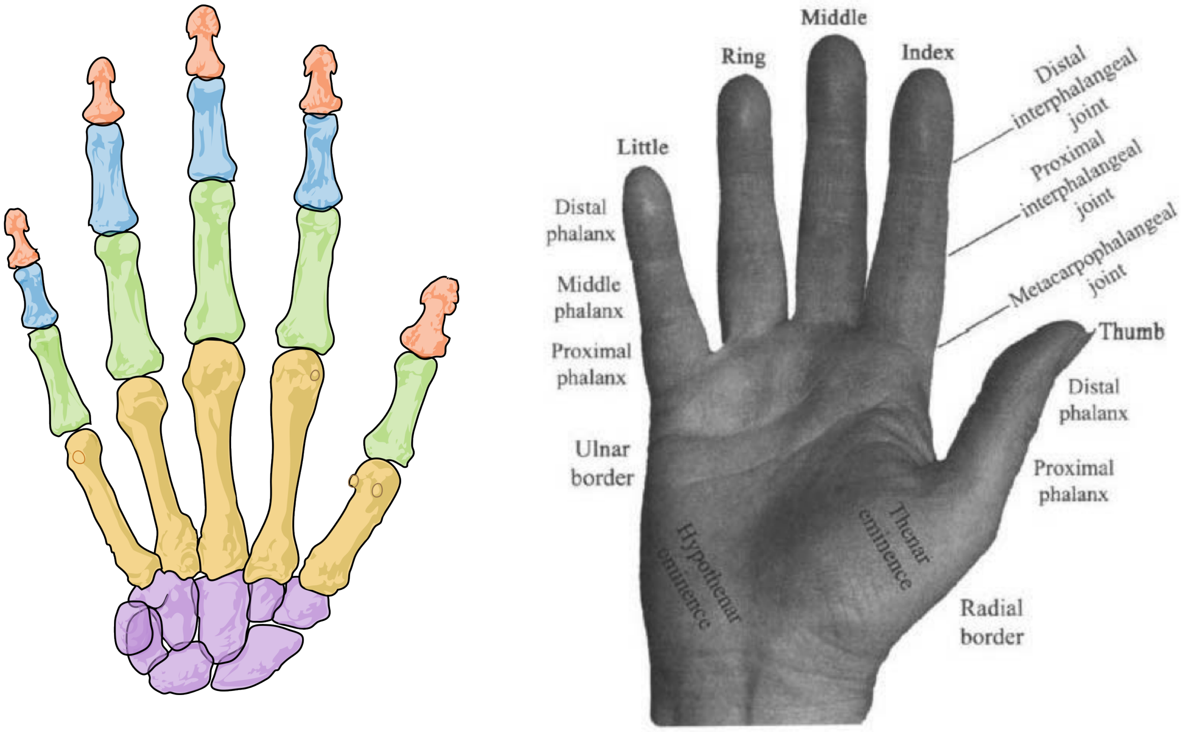


Figure 2.1: Bones and volar aspect of the hand and standard anatomical terminology.

The hand is made of 27 bones, with 8 carpal bones constituting the wrist. Referring to Figure 2.1 there are 5 metacarpal bones in the palm, 2 phalangeal bones in the thumb and 3 phalangeal bones in the other 4 fingers. The proximal row comprises the scaphoid, lunate, triquetrum, and pisiform, and the distal row, which articulates with the metacarpals, comprises the trapezium, trapezoid, capitate, and hamate. The metacarpal bones in the palm of the hand articulate closely with the adjacent carpal bones in the distal row, and these carpometacarpal joints are capable of flexion/extension movements and radial and ulnar deviation [Jones et al., 2006].

With the exception of the first metacarpal of the thumb, independent motion of these joints is very limited but the range of movement increases from the second to the fifth metacarpal [Kapandji, 1970, Taylor et al., 1970]. The five meta-carpophalangeal joints(biaxial or condyloid type) are universal or saddle joints capable of both flexion/extension and abduction / adduction movements), whereas the nine interphalangeal joints in the digits are hinge joints capable of only flexion and extension. The three bones in the fingers are known as the proximal, middle, and distal phalanges, and each finger has three joints, the metacarpophalangeal(MP), the proximal interphalangeal (PIP), and the distal interphalangeal (DIP) joints.

2.1.2 Relationship between joints

According to [Jones et al., 2006] the total active range of motion of a typical finger is 260°, in which is the sum of active flexion at the MP (85°), PIP (110°), and DIP (65°) joints. The range of active extension at the MP joint varies between people but can reach 30-40° [Kapandji, 1970]. Passive and active flexion of the MP joint increases linearly from the index to the little finger, and the total active range of motion of the fingers also increases from the index to the little finger. Although there is no passive extension beyond 0° at the PIP joint, at 30° it is appreciable at the DIP joint, a uniquely human feature. With the exception of the thumb, the index finger has the greatest range of abduction/adduction movements at 30°. These movements become difficult, if not impossible, when the MP joint is flexed due to tautness in the collateral ligaments of the joint. In contrast to the other digits, the thumb does not have a second phalanx and so has only two phalangeal bones and much greater mobility in the carpometacarpal joint. The carpometacarpal joint of the thumb is described by many authors as a saddle joint with two degrees of freedom. Although there is a considerable axial rotation in addition to flexion/extension and abduction/adduction movements, this is constrained and so it is not considered a true third degree of freedom. In total, the human hand, including the wrist has 21 degrees of freedom of movement [Jones et al., 2006, Kang Li et al., 2010].

A common reduction of the number of degrees of freedom is to consider that for the

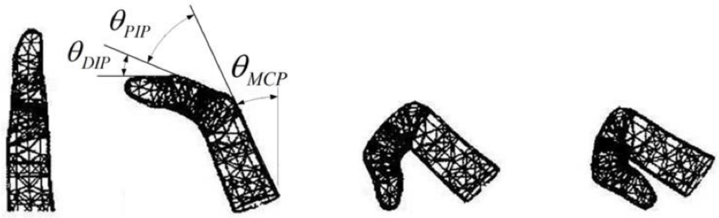


Figure 2.2: Contrains between PIP and DIP joints.

index, middle, ring, and little fingers, in order to bend the DIP joints the correspondent PIP joints must also bend, as seen in Figure 2.2.

According to [Kang Li et al., 2010] the relationship can be approximately presented as:

$$\theta_{DIP} = \frac{2}{3} \times \theta_{PIP} \tag{2.1.1}$$

2.2 Sign Language

A gesture is a unit of meaning. It can provide the same information of the "word" in a vocal language. From the dynamics of a gesture, like speed or intensity, one can express, different meanings. From the broad need of representing those meanings a sign language is used [Khan et al., 2009]. In the next sections the difference between static and dynamic gestures is presented and an overview of the portuguese sign language is given.

2.2.1 Static gestures

Static gestures allows the representation of simple sign language expressions of actions. They can be understood as the static positioning of the fingers in relation to the hand's wrist. It may also include some rotation of the wrist.

2.2.2 Dynamic gestures

Dynamic gestures allows a richer expression of the sign language. A dynamic gesture means that all the fingers and wrist can vary its positioning and rotation in relation to time. Usually there is a start and stop moment only defined by the observer's recognition of gesture. An analogy to a vocal communication would be one person speaking a sentence too fast and yet a listener understanding every word since he knows *à priori* the words of that sentence.

2.2.3 Portuguese Sign Language

Like many sign languages in many countries, there has not been a unified sign language in Portugal. By the descriptive nature of this kind of language it was common to have

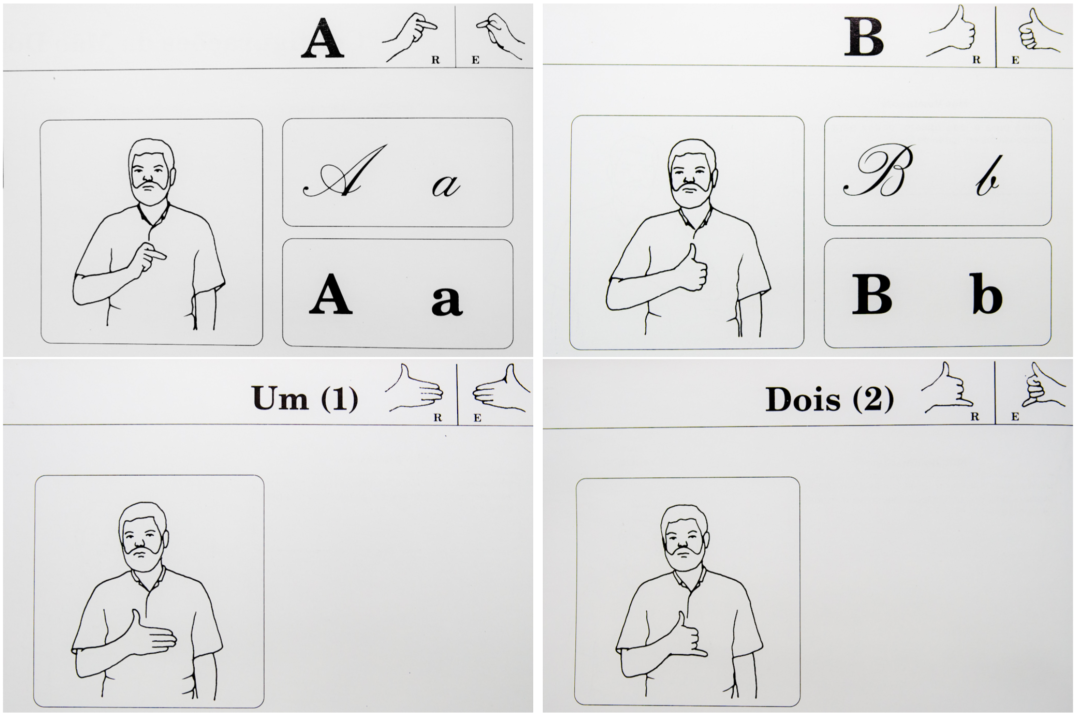


Figure 2.3: Some examples of static gestures from the alphabet of the Portuguese Sign Language.

localized languages in different places in the same country. To deal with this reality, the portuguese institutions with the support of the government worked to provide a common framework for the identification of the words [Gestuário-LGP, 1997]. This framework, called *Gestuário*, details what is now called *Língua Gestual Portuguesa*.

In this sign language the most elementary words such as the alphabet and the numbers like in Figure 2.3 are represented by static gestures. More meaningful words make use of more complex gestures and therefore it is much more common the use of dynamic gestures to express those words.

2.3 Summary

This chapter presented the human hand and the relevant concepts that will be used in the following chapters. In this dissertations the recognition of gestures will be based on the Portuguese Sign Language. Specifically, the alphabet and the first ten numbers which are represented by static gestures will be used.

Chapter 3

Inertial Sensors

In this chapter a brief introduction on inertial sensors and in some detail the low-cost sensors that enable the measurement of gravity's acceleration will be presented.

3.1 Definition

Accelerometers and gyroscopes can be understood as inertial sensors since they relate to the properties of inertia, i.e., refer to the resistence to a change in momentum. These sensors measure the changes in motion. In the case of accelerometers they measure the linear motion and in the case of gyroscopes they measure changes in the angular motion. Also simpler innertial sensors are inclinometers who measure the orientation of the acceleration vector.

In this thesis a distribuition of accelerometers to capture the measurements of gravity's acceleration in each finger and palm is used.

3.2 Distributed accelerometers

A triaxial (or 3-axial) accelerometer is a device capable of measuring acceleration in three orthogonal axis. An example of this sensor is the MM7260QT from Freescale Semiconductor [Freescale MMA72]. This device is a low-cost capacitive micromachined accelerometer and can measure gravity's acceleration up to 6G.

The acceleration in each axis is measured by the movement of a movable central mass,

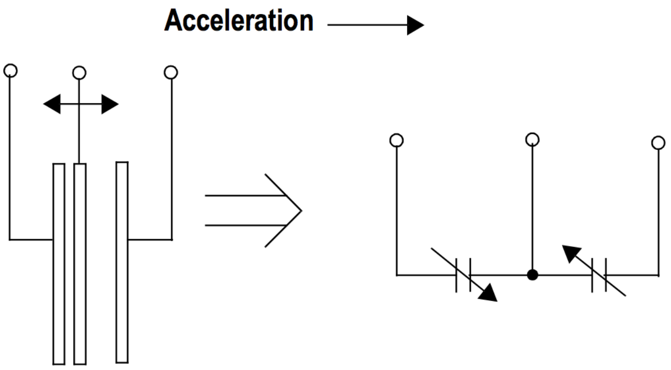


Figure 3.1: Simplified transducer physical model of a capacitive accelerometer.

as shown in Figure 3.1 and 3.4. It uses switch capacitor techniques to measure the capacitors. From the difference between the capacitors the sensor extracts the acceleration and outputs a voltage proportional to acceleration [MMA72 Technical Data].

In Figure 3.2 it is shown how a distribuited accelerometer approach, with a sensor in each finger and palm, would have its frames of reference disposed.

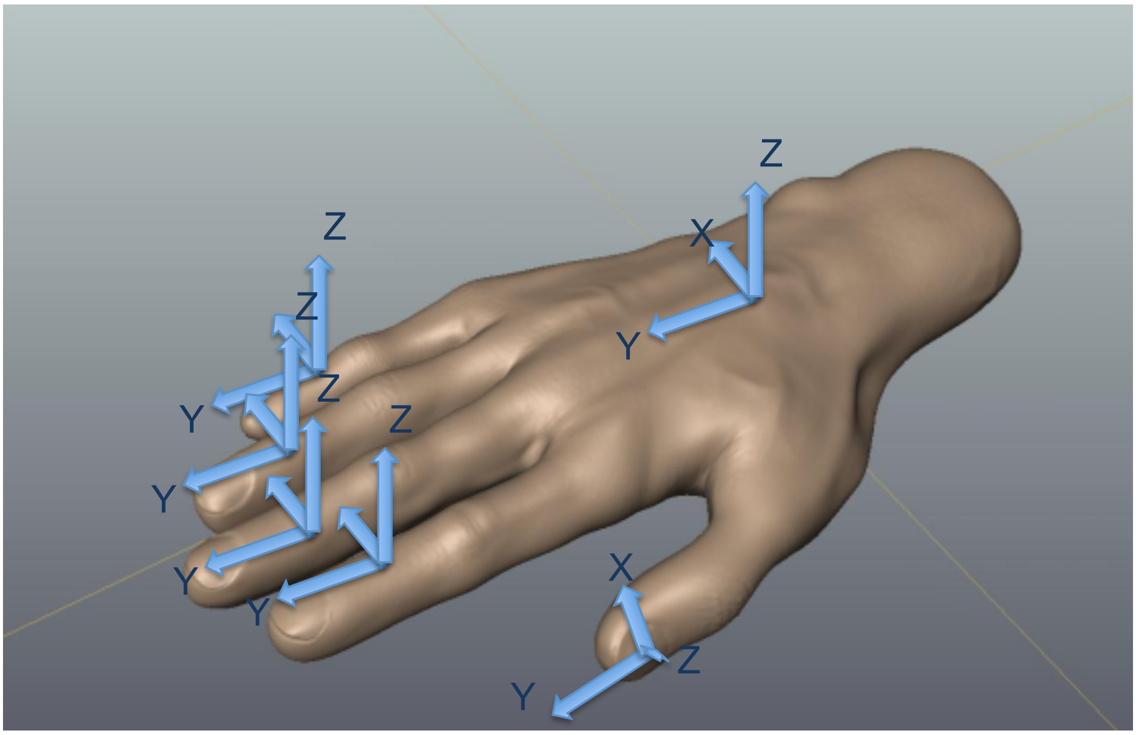


Figure 3.2: The different frames of reference for a distributed accelerometer system with a sensor in each finger and the palm.

3.3 Other movement sensors

Other sensors such as gyroscopes measure angular velocity. They have a very large use in military and some comercial industries. In [Barbour, 2003] a much greater detail is presented in concern to the military impact of gyroscopes technology.

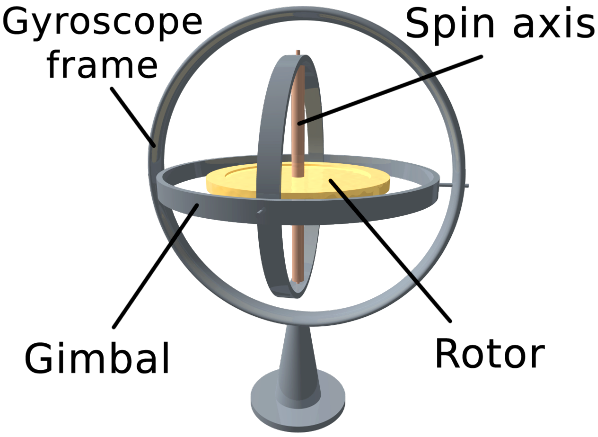


Figure 3.3: General overview of a gyroscope.

In Figure 3.3 a model of a gyscope can be visualized. Its working principal for measurement relies on the deviation of its constant rotation. In more detail, [Fraden, 2006] says it can be explained by Newton's law of motion for rotation: The time rate of change of angular momentum about any given axis is equal to the torque applied about the given axis. That is to say, when a torque T is applied about the input axis, and the speed ω of the wheel is held constant, the angular momentum of the rotor may be changed only by rotating the projection of the spin axis with respect to the input axis; that is, the rate of rotation of the spin axis about the output axis is proportional to the applied torque $T = I\omega\Omega$ where Ω is the angular velocity about the output axis and I is the inertia of a gyro wheel about the spin axis.

Although it could be a precise tool for velocity measurement in the first instances of measurement, over time it degrades and deviates from the real values. This drift is relative to its angular position (obtained by integrating the measured velocity). In [Weinberg et al., 2006] it is described that this error can vary between 1 to 10º/h in tactical-grade perfomance while in comercial-grade it can vary thousand º/h. This makes the use of this technology too unprecise or too expensive.

MEMS gyroscopes have become increasingly important as they have been implemented in a variety of consumer products, because of their gradually decreasing cost.

The basic principle of MEMS Vibrating Structure Gyroscopes (VSG) is producing radial linear motion and measuring the Coriolis effect induced by rotation. If a sensing element is made to vibrate in a certain direction, say along the x-axis, rotating the sensor around the z-axis will produce vibration in the y direction with the same frequency. The amplitude of this vibration is determined by the rotation rate. The geometry used takes into account, amongst other factors, the cancelling out of unwanted accelerations.

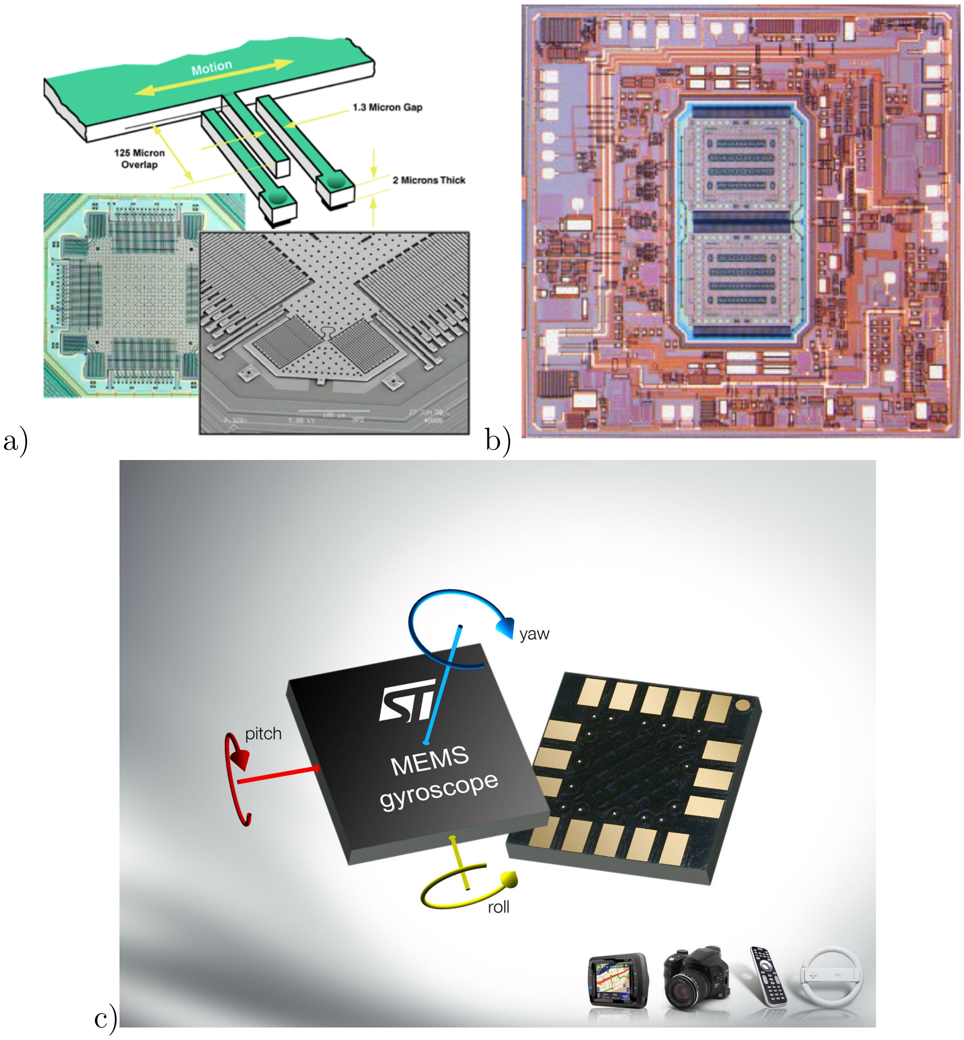


Figure 3.4: A look at the current MEMS accelerometers and gyroscopes.
a) Analog Devices ADL202 dual axis accelerometer; b) Analog Devices ADXRS150 vibrating structure angular rate sensor; c) STMicroelectronics 2009 gyroscopes' family

3.4 Summary

In the above sections there is an overview of the technologies for inertial measurements. It was refered the accelerometers, gyroscopes and even inclinometers as sensors capable of movement measurement. For this work accelerometers were used, details are given in chapter 6 on the specific hardware, namely the Anthrotronix Acceleglove system.

Chapter 4

Gesture Recognition

This chapter describes one of the main goals in this thesis. As mentioned in chapter 2 it is recognizing gestures from the Portuguese Sign Language, in specific, the numbers and letters of the latin alphabet (0, ..., 9; A, ..., Z). In the next sections a description of the steps involved in the recognition, are described.

4.1 Overview

When there is no acceleration, the gravity vector provides a vertical reference so that the measured acceleration can provide information about the pose of the accelerometer (triaxial) sensor. However this provides only two inclination angles, but no azimuth since rotations about the vertical are not sensed. To overcome this limitation a method, based on [Lobo, 2007] is used. Each triad of accelerometers can be seen as an observer of the gravity vertical reference, when the sensor is static the measuments provide a vertical reference vector in the sensor local frame of reference. By using two set of vectors, one given by the accelerometer on a finger and the other set of vectors given by the accelerometer in the palm it is possible to obtain the full 3D pose of the finger's accelerometer in relation to the palm accelerometer.

In Figure 4.1 is presented a general overview for the process of recognizing gestures. The starting point is with values of the acceleration of each axis of each sensor. From that a filtering of some systematic errors is made. Then it is necessary to identify the subsets of time where there is no motion (static) or where the motion is smooth or sudden. This

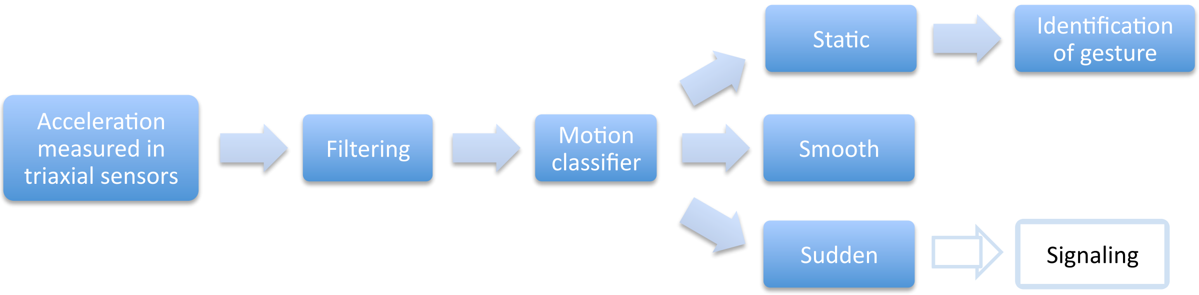
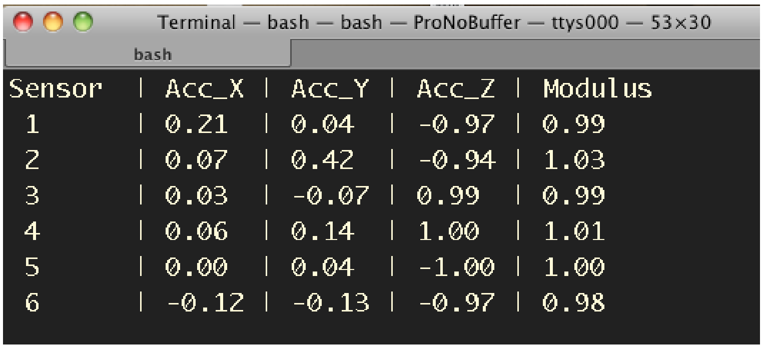

Acceleration measured in triaxial sensors
Filtering
Motion classifier
Static
Smooth
Sudden
Identification of gesture
Signaling

Figure 4.1: Overview of the phases for a gesture recognition.



```
Sensor  | Acc_X | Acc_Y | Acc_Z | Modulus
 1      | 0.21  | 0.04  | -0.97 | 0.99
 2      | 0.07  | 0.42  | -0.94 | 1.03
 3      | 0.03  | -0.07 | 0.99  | 0.99
 4      | 0.06  | 0.14  | 1.00  | 1.01
 5      | 0.00  | 0.04  | -1.00 | 1.00
 6      | -0.12 | -0.13 | -0.97 | 0.98
```

Figure 4.2: Example values of readings, in G-foce units, from 6 triaxial accelerometers.

is important because we are recognizing static gestures. When there is no motion, i.e., static, the process enters the final step of the classifier, where the gesture is recognized. Also represented in Figure 4.1 is the signaling obtained from sudden motion detection. This has the use of allowing to signal the start and end of a gesture.

4.2 Acceleration

When the hand is static, gravity is evaluated by its G-force value in each axis of the 3-axial sensor. Since we are interested in recognizing static gestures we use gravity's acceleration for the recognition. In the case of the Figure 4.2 it is shown the readings of the 6 triaxial accelerometers (Freescale MMA720QT) of a commercial glove (Anthronix Acceleglove).

4.3 Filtering

Like in all transducers, a measurement from an accelerometer comes with some error due to noise. In the case of accelerometers this error can be explained by the conversion from the real physical quantity to the electric signal; it can also be easily understood that even a static hand pose from a living person will always produce some movement.

The filter applied was a mean filter. A mobile window of a pre-defined size was used to calculate the mean for each sample.

So, for every sample processed s_i, after aplying the filter we would obtain the filtered sample f_i given by:

$$f_i = \frac{1}{N}\sum_{k=-N/2}^{N/2} s_{(i+k)} \qquad , \qquad N = \begin{cases} W-1 & , if\, W\, is\, odd \\ W & , if\, W\, is\, even \end{cases} \tag{4.3.1}$$

where W is the size of the mobile window. This value is determined experimentally.

4.4 Motion classifier

The motion classifier defines for each sample a level of movement. This step is of major importance because it allows to procure the static gestures. Moreover it can detect some signalling from sudden movements.

Each sample of a sensor reading is defined by a vector, since it's a triaxial sensor. With that vector it was possible to use its modulus. If there is no appreciable movement with the sensor then its acceleration modulus should remain static. Based on this information every sample was compared against its neighbouring samples. Three thresholds levels $\{M_1, M_2, M_3\}$ are defined to allow the motion classification in relation to the deviations of any given samples against their closest neighbours.

Let W be the depth of neighbouring search, so that for a given a sample s_i its neighbours are defined between $\left[s_{(i-W)}, \ldots, s_{(i+W)}\right]$ and let L_i be the level of motion of sample s_i, then for each sample s_i.

$$\begin{cases} L_i = 1 & \left(\max\left\{s_{(i-W)}, \ldots, s_{(i+W)}\right\} - \min\left\{s_{(i-W)}, \ldots, s_{(i+W)}\right\}\right) < M_1 \\ L_i = 2 & M_1 \leq \left(\max\left\{s_{(i-W)}, \ldots, s_{(i+W)}\right\} - \min\left\{s_{(i-W)}, \ldots, s_{(i+W)}\right\}\right) < M_2 \\ L_i = 3 & M_2 \leq \left(\max\left\{s_{(i-W)}, \ldots, s_{(i+W)}\right\} - \min\left\{s_{(i-W)}, \ldots, s_{(i+W)}\right\}\right) < M_3 \end{cases} \tag{4.4.1}$$

4.5 Identification of gesture

Gesture identification happens by comparing the fingers' angles of an unknown gesture against a library of known gestures. By using an accelerometer in each finger and palm, a plethora of different gestures is recognizable.

On previous section each sample was classified with a level ranging from 1 to 3. Level 1 represented the most static motion of the three possible levels. From the list of all samples, each contiguous subset of level-1 samples were grouped. From each group of level-1 samples the mean value of that subset was extracted and labeled as a *frame*. Figure 4.3 provides a visual representation of those relations.

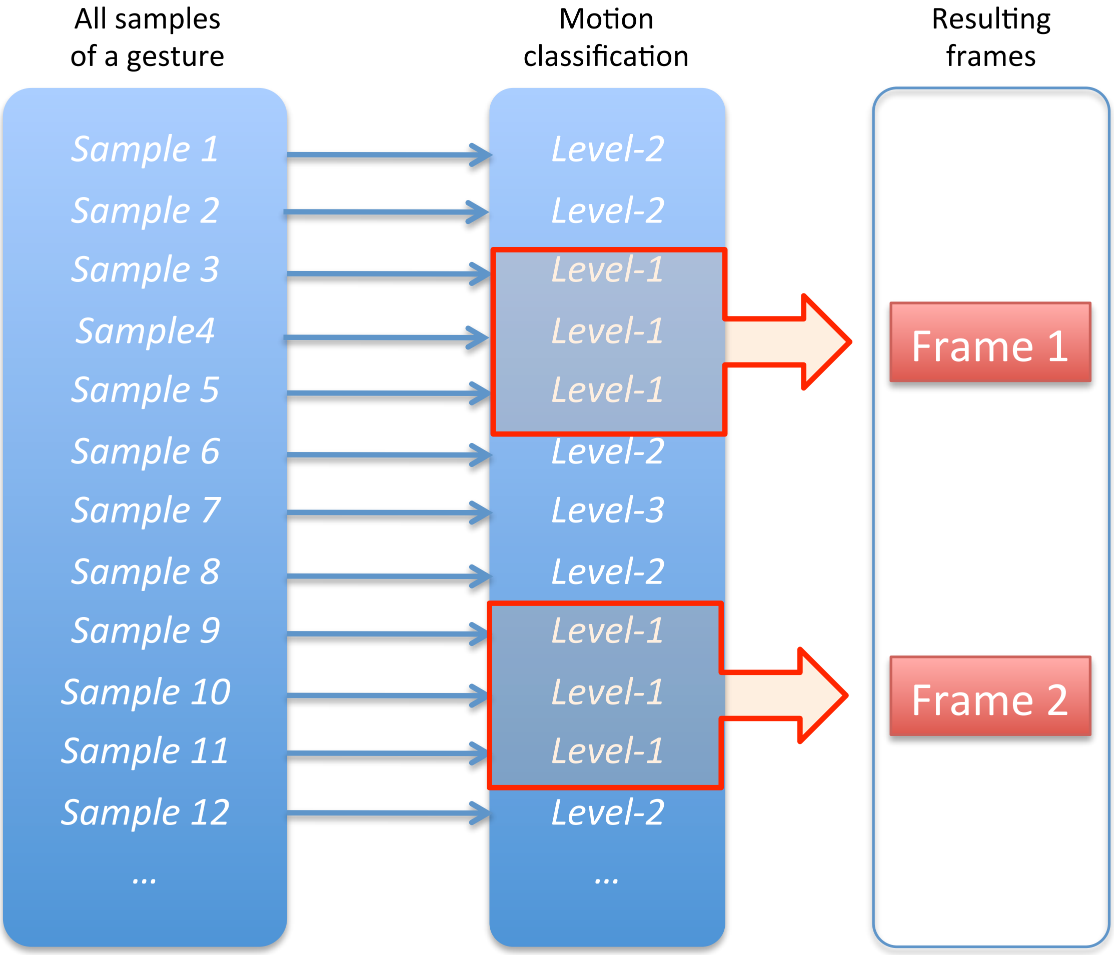


Figure 4.3: Diagram relating the samples of a gesture to a set of frames of that gesture.

With a collection of frames for a single gesture it is possible to estimate the full 3D angular pose of each finger in relation to the palm.

In [Lobo, 2007] a method to determine the rotation between two frames of references is proposed by using Horn's closed-form solution for absolute orientation using unit quaternions. This is described next.

Let ${}^{\mathcal{F}}\mathbf{v}_i$ be a measurement of the vertical by the inertial sensors on the finger, and ${}^{\mathcal{P}}\mathbf{v}_i$ the corresponding measurement made by the sensor in the palm. We want to determine the unit quaternion $\mathring{\mathbf{q}}$ that rotates inertial measurements in the finger frame of reference $\{\mathcal{F}\}$ to the palm frame of reference $\{\mathcal{P}\}$. We want to find the unit quaternion $\mathring{\mathbf{q}}$ that maximises

$$\sum_{i=1}^{n} (\mathring{\mathbf{q}}\, {}^{\mathcal{F}}\boldsymbol{v}_i\, \mathring{\mathbf{q}}^*) \cdot {}^{\mathcal{P}}\boldsymbol{v}_i \tag{4.5.1}$$

which can be rewritten as

$$\sum_{i=1}^{n} (\mathring{\mathbf{q}}\, {}^{\mathcal{F}}\boldsymbol{v}_i) \cdot ({}^{\mathcal{P}}\boldsymbol{v}_i\, \mathring{\mathbf{q}}) \tag{4.5.2}$$

The quaternion product can be expressed as a matrix. Using ${}^{\mathcal{F}}\boldsymbol{v}_i = ({}^{\mathcal{F}}x_i,^{\mathcal{F}} y_i,^{\mathcal{F}} z_i)^T$ and ${}^{\mathcal{P}}\boldsymbol{v}_i = ({}^{\mathcal{P}}x_i,^{\mathcal{P}} y_i,^{\mathcal{P}} z_i)^T$ we define

$$\mathring{\mathbf{q}}\, {}^{\mathcal{F}}\boldsymbol{v}_i = \begin{bmatrix} 0 & -{}^{\mathcal{F}}x_i & -{}^{\mathcal{F}}y_i & -{}^{\mathcal{F}}z_i \\ {}^{\mathcal{F}}x_i & 0 & {}^{\mathcal{F}}z_i & -{}^{\mathcal{F}}y_i \\ {}^{\mathcal{F}}y_i & -{}^{\mathcal{F}}z_i & 0 & {}^{\mathcal{F}}x_i \\ {}^{\mathcal{F}}z_i & {}^{\mathcal{F}}y_i & -{}^{\mathcal{F}}x_i & 0 \end{bmatrix} \mathring{\mathbf{q}} =^{\mathcal{F}} \mathbf{V}_i \mathring{\mathbf{q}} \tag{4.5.3}$$

and

$${}^{\mathcal{P}}\boldsymbol{v}_i\, \mathring{\mathbf{q}} = \begin{bmatrix} 0 & -{}^{\mathcal{P}}x_i & -{}^{\mathcal{P}}y_i & -{}^{\mathcal{P}}z_i \\ {}^{\mathcal{P}}x_i & 0 & -{}^{\mathcal{P}}z_i & {}^{\mathcal{P}}y_i \\ {}^{\mathcal{P}}y_i & {}^{\mathcal{P}}z_i & 0 & -{}^{\mathcal{P}}x_i \\ {}^{\mathcal{P}}z_i & -{}^{\mathcal{P}}y_i & {}^{\mathcal{P}}x_i & 0 \end{bmatrix} \mathring{\mathbf{q}} =^{\mathcal{P}} \mathbf{V}_i \mathring{\mathbf{q}} \tag{4.5.4}$$

Substituting in (4.5.2)

$$\sum_{i=1}^{n} ({}^{\mathcal{F}}\mathbf{V}_i \mathring{\mathbf{q}}) \cdot ({}^{\mathcal{P}}\mathbf{V}_i \mathring{\mathbf{q}}) \tag{4.5.5}$$

or

$$\sum_{i=1}^{n} \mathring{\mathbf{q}}^T\, {}^{\mathcal{F}}\mathbf{V}_i^T\, {}^{\mathcal{P}}\mathbf{V}_i \mathring{\mathbf{q}} \tag{4.5.6}$$

factoring out $\mathring{\mathbf{q}}$ we get

$$\mathring{\mathbf{q}}^T \left(\sum_{i=1}^{n} {}^{\mathcal{F}}\mathbf{V}_i^T \, {}^{\mathcal{P}}\mathbf{V}_i \right) \mathring{\mathbf{q}} \tag{4.5.7}$$

So we want to find $\mathring{\mathbf{q}}$ such that

$$\max \mathring{\mathbf{q}}^T \, \boldsymbol{N} \, \mathring{\mathbf{q}} \tag{4.5.8}$$

where

$$\boldsymbol{N} = \sum_{i=1}^{n} {}^{\mathcal{F}}\mathbf{V}_i^T \, {}^{\mathcal{P}}\mathbf{V}_i \; . \tag{4.5.9}$$

Having

$$S_{xx} = \sum_{i=1}^{n} {}^{\mathcal{F}}x_i \, {}^{\mathcal{P}}x_i \; , \; S_{xy} = \sum_{i=1}^{n} {}^{\mathcal{F}}x_i \, {}^{\mathcal{P}}y_i \tag{4.5.10}$$

and analogously for all 9 pairings of the components of the two vectors, matrix $\boldsymbol{N}$ can be expressed using these sums as in (4.5.11). The sums contain all the information that is required to find the solution.

$$\boldsymbol{N} = \begin{bmatrix} (S_{xx} + S_{yy} + S_{zz}) & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & (S_{xx} - S_{yy} - S_{zz}) & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & (-S_{xx} + S_{yy} - S_{zz}) & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & (-S_{xx} - S_{yy} + S_{zz}) \end{bmatrix} \tag{4.5.11}$$

Since $\boldsymbol{N}$ is a symmetric matrix, the solution to this problem is the four-vector $\boldsymbol{q}_{max}$ corresponding to the largest eigenvalue λ_{max} of $\boldsymbol{N}$. [Horn, 1987]

Having the quaternion of the rotation between frames (obtained from a sequence of distinct static poses of the same gestures) we can also imediately convert it to get the Roll, Pitch and Yaw of the 5 relative pose rotation quaternions and 2 angles (Roll and Pitch) directly from the palm triaxial accelerometer.

For the accelerometer in the palm of the hand we use the approach proposed by [Takayuki et al., 2007] where

$$s = \begin{bmatrix} gsin\beta \\ -gcos\beta sin\alpha \\ -gcos\beta cos\alpha \end{bmatrix} \tag{4.5.12}$$

where $s = [s_x, s_y, s_z]$ is the measured accelation and in our case g is the G-force of gravity, which is equal 1. The angles Roll, α, and Pitch, β are obtained by solving 4.5.12.

Having the pose of every accelerometer estabilished it is then possible to define some metrics to compare to the gestures' library. Every gesture stored in the library has the information of the pose for each of the 6 sensors. Using this information, a deviation error is calculated between the gesture being evaluated and every gesture in the library.

More explicity, each component (Roll, Pitch and Yaw) of every finger and palm is compared to the ones in the gesture library. The error is calculated by measuring the angular distance between components. Then it is made a weighted aritmetic mean calculation of all the components-errors. This final value is normalized against pi since it's the maximum value of the error.

From this a table emerges providing the gestures in the library and the estimated probability of being the correct gesture. The minimum deviation error is used to recognize the correct gesture being performed.

To enhance the probability of a correct recognition it is added weight to the values being used to calculate the deviation error. Namely, the roll movement of the human finger, in relation to its palm has the most influence in the calculation when compared to the influence of the yaw rotation. In the case of the fingers there is no relevant pitch rotation, as stated before.

4.6 Summary

This chapter defined the methods and approaches used for the gesture cognition. The implementation and results are presented in later chapters.

Chapter 5

Visualization

Visualization of a gesture is the other main goal of this thesis. In this chapter it is overviewed how this can be achieved and the solutions proposed in this thesis. The developed visualization tools were also developed to accommodate the datasets from the Handle[1] project that had multiple sensors registering human manipulation of objects.

5.1 Graphics software

3D graphics is the grandchild of Euclids Elements, a geometric construction of the Universe as a mesh of points connected by measurable lines [Oliver, 2008]. OpenGL is a natural choice for drawing those primitives. But to allow more control and fast implementation, a higher level software layer is needed. Blender is a free open source 3D content creation suite that allows several different approaches for 3D representation. Moreover, it allows scripting control of the software.

Also, with aim to escalate the use of this visualization beyond the scope of this thesis, choosing Blender allowed the use of this modeling in the European project HANDLE: Developmental pathway towards autonomy and dexterity in robot in-hand manipulation, a Large Scale IP project coordinated by the university Pierre and Marie Curie of Paris that includes a consortium formed by nine partners from six EU countries: France, UK,

[1] HANDLE is a collaborative project is funded by the European Commission within the Seventh Framework Programme FP7, as part of theme 2: Cognitive Systems, Interaction, Robotics, under grant agreement 231640.

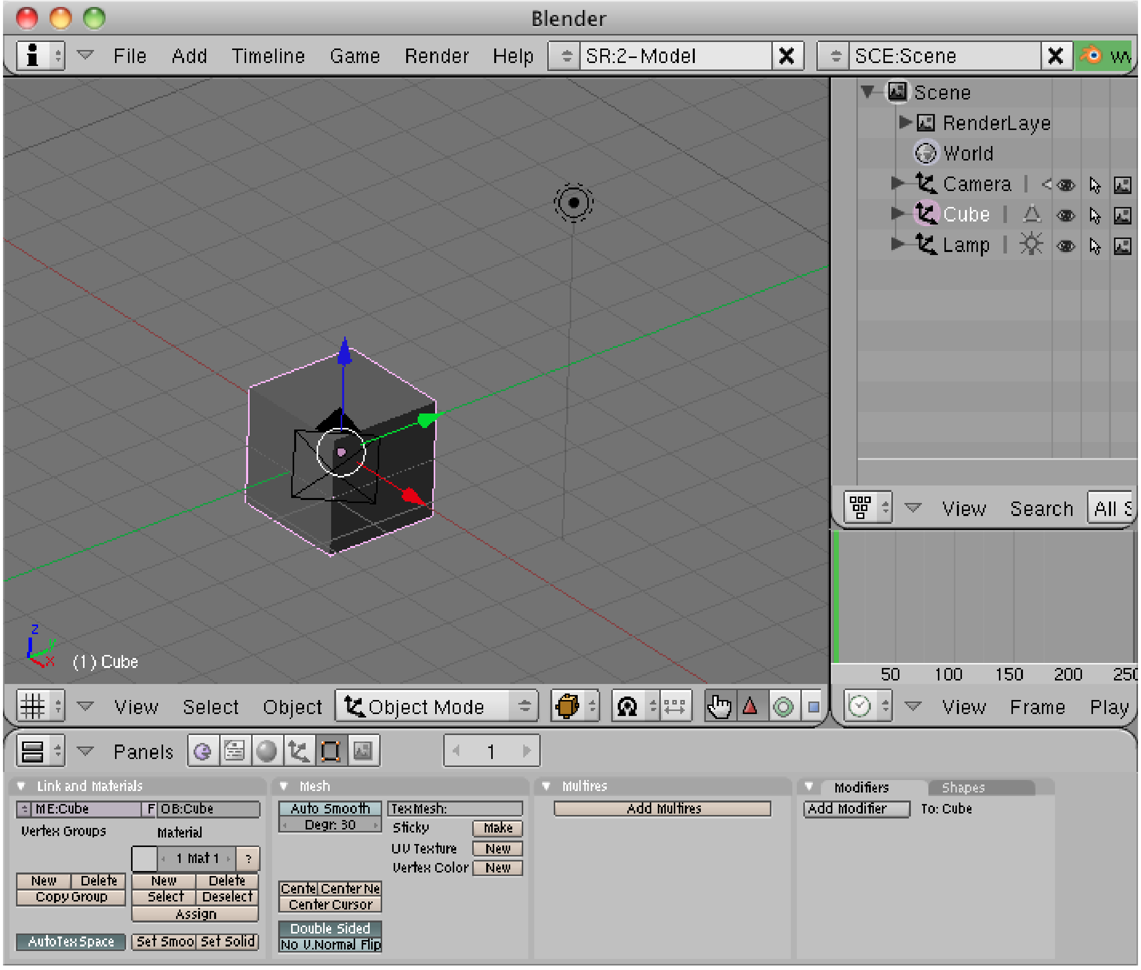


Figure 5.1: The typical Blender environment.

Spain, Portugal, Sweden, and Germany [Handle, 2010]. The visualization being developed for this thesis fulfills the requirements for the visualization of the information from a full set of sensors attached to a human hand, as used in the HANDLE Project.

5.2 Blender

Blender is an open-source software and has its development from skillful volunteers. This includes Universities and graduated people working to develop parts of Blender to explore its full potential.

Blender, whose typical environment is shown in Figure 5.1 currently allows the creation and edition of meshes, materials. It also allows the user to define motion of the objects it creates inside the software. Although these characteristics are important for any 3D modeler, the most aspect of Blender is its powerful scripting possibilities combined with its game engine.

Blender allows Python scripting language to control all the environment of the software. This means it is possible to dynamically use one's own equations and algorithms to set up the environment.

In this thesis work this resulted in reading the output of the processing of information from other external sources (such as Matlab or anyother program) and have this information resulting in a user-controlled rendering. This workflow can be seen in Figure 5.2.

From the workflow diagram (Figure 5.2) it becomes easy to understand the communication between the softwares. Matlab makes the offline processing for the recognition, as described in the previous chapters. After its processing, an output information of the Roll, Pitch and Yaw angles for each finger and palm is passed to an intermediate File. Blender is expecting that file and continuously monitors it for changes. This is the secret for the Blender real-time capability. This capability is mostly used when the online python scripting is being used. This script connects directly to the sensors and translates its information to a Roll and Pitch values. These values are passed to the intermediate File and consequently Blender displays in real-time the changes to the angular pose of the hand.

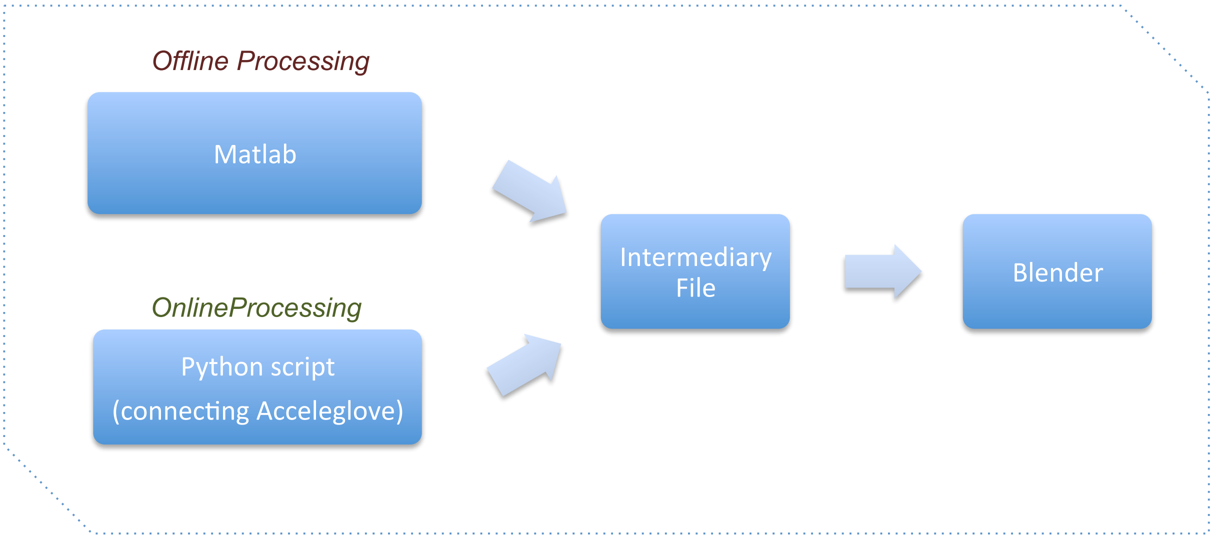
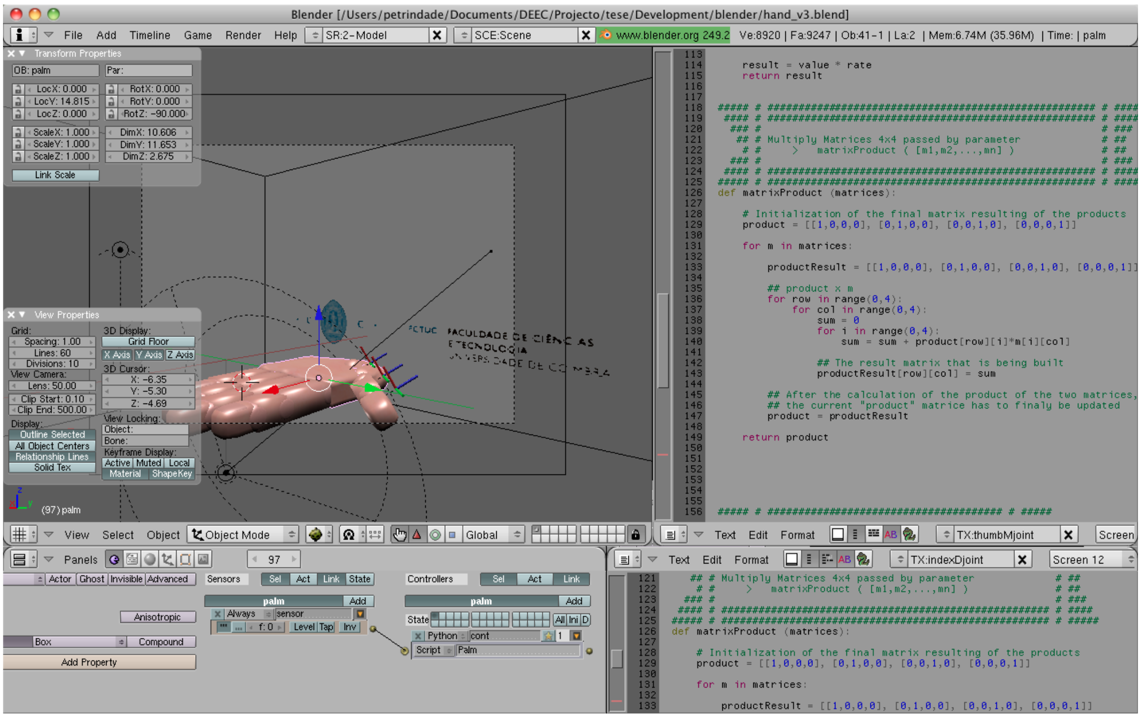


Figure 5.2: Workflow for the visualization of the hand angular pose.

The work of this thesis procured to have a render capability of different poses of the joints in the hand. For this scripting functions were created and connected to the blender environment to allow the change in the pose. This can be seen in Figure 5.3. This programming strategy allowed visualization to be expanded to include HANDLE project requirements. By defining a common platform of communication with external sources of information, it was possible to define any objetct's pose inside blender.


Blender [/Users/petrindade/Documents/DEEC/Projecto/tese/Development/blender/hand_v3.blend]

Ve:8920 | Fa:9247 | Ob:41-1 | La:2 | Mem:6.74M (35.96M) | Time: | palm

Transform Properties
OB: palm
Par:
LocX: 0.000
LocY: 14.815
LocZ: 0.000
RotX: 0.000
RotY: 0.000
RotZ: -90.000
ScaleX: 1.000
ScaleY: 1.000
ScaleZ: 1.000
DimX: 10.606
DimY: 11.653
DimZ: 2.675

View Properties
Grid:
Spacing: 1.00
Lines: 60
Divisions: 10
View Camera:
Lens: 50.00
Clip Start: 0.10
Clip End: 500.00
3D Cursor:
X: -6.35
Y: -5.30
Z: -4.69
View Locking:
Object:
Bone:

(97) palm

Sensors: palm — Always — sensor — f: 0
Controllers: palm — Python — cont — 1 — Script — Palm

```
    result = value * rate
    return result


##### # ################################################ # ####
 #### # ################################################ # ####
  ### #                                                  # ###
   ## # Multiply Matrices 4x4 passed by parameter        # ##
    # #      >   matrixProduct ( [m1,m2,...,mn] )        # ##
  ### #                                                  # ###
 #### # ################################################ # ####
##### # ################################################ # ####
def matrixProduct (matrices):

    # Initialization of the final matrix resulting of the products
    product = [[1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]]

    for m in matrices:

        productResult = [[1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]]

        ## product x m
        for row in range(0,4):
            for col in range(0,4):
                sum = 0
                for i in range(0,4):
                    sum = sum + product[row][i]*m[i][col]

                ## The result matrix that is being built
                productResult[row][col] = sum

        ## After the calculation of the product of the two matrices,
        ## the current "product" matrice has to finaly be updated
        product = productResult

    return product







##### # ########################################## # #####
```

```
   ## # Multiply Matrices 4x4 passed by parameter        # ##
    # #      >   matrixProduct ( [m1,m2,...,mn] )        # ##
  ### #                                                  # ###
 #### # ################################################ # ####
##### # ################################################ # #####
def matrixProduct (matrices):

    # Initialization of the final matrix resulting of the products
    product = [[1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]]

    for m in matrices:

        productResult = [[1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]]
```

Figure 5.3: Blender development environment to allow control of pose from an external source.

5.3 Summary

In this chapter the software to allow the visualization was introduced and described. Blender is the software that allows the visualization. Its main advantages are its capability of allowing the user to control the environment by mean of python scripting. This allows the visual representation of external information such as the pose of objects or joints, which represent the focus of this thesis.

Chapter 6

Implementation and Results

This chapter details the aspects of the implementation of the methods and ideas introduced in the previous chapters, and presents the experimental results.

6.1 Experimental setup

The experimental setup was constituted by the Acceleglove [Acceleglove, 2010], which is a device with USB connectivity and a PC with Matlab software for data processing. The experimental setup also included the Blender software, a C compiler and Python scripting language installed on the PC.

At the Mobile Robotics Laboratory the HANDLE project [Handle, 2010] has its own experimental setup, as ilustrated in Figure 6.1.

Since developments in this thesis were to be used in the HANDLE Project some of the sensors from HANDLE were prepared for use within this thesis. Such is the case of the Polhemus and the Monocular Camera.

The Polhemus, a magnetic tracker system allows to collect the position and pose of each of its trackers. By having this sensors attached to the hand it is possible to have a ground-truth for controlling the development of the software and algorithms.

Another sensor prepared to be used was the Monocular camera. This sensor collects images of the data acquisition providing a visual reference for the pos-acquisition analysis.

To organize this whole information it was proposed a XML file structure where all the sensor's values were timestamped and each was saved in a XML file.

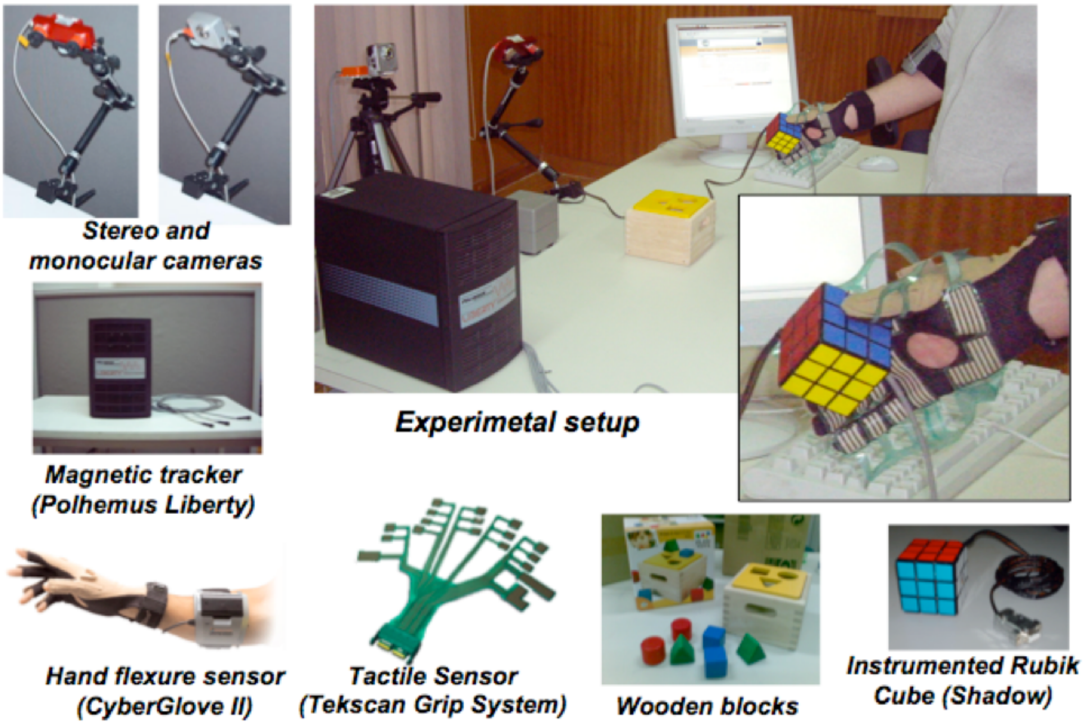

Stereo and monocular cameras
Experimetal setup
Magnetic tracker (Polhemus Liberty)
Hand flexure sensor (CyberGlove II)
Tactile Sensor (Tekscan Grip System)
Wooden blocks
Instrumented Rubik Cube (Shadow)

Figure 6.1: Experimental setup for the HANDLE Project.

6.1.1 Acceleglove

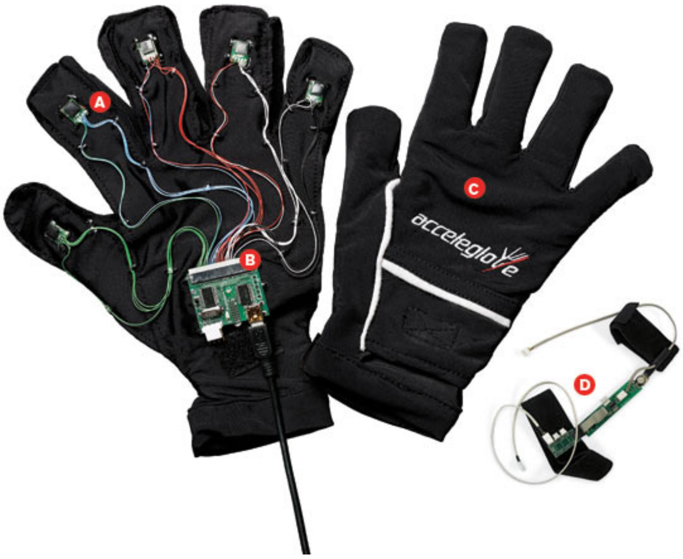


Figure 6.2: A stripped view of AnthroTronix Acceleglove.

The device used for measuring acceleration was the Anthrotronix Acceleglove [Acceleglove, 2010]. This is a licra glove with sensors at the end of the fingers and also a sensor and the controller at the palm (Figure 6.2). These sensors are the Freescale's MMA720QT [Freescale MMA72], similar to the one shown in Figure 6.3.

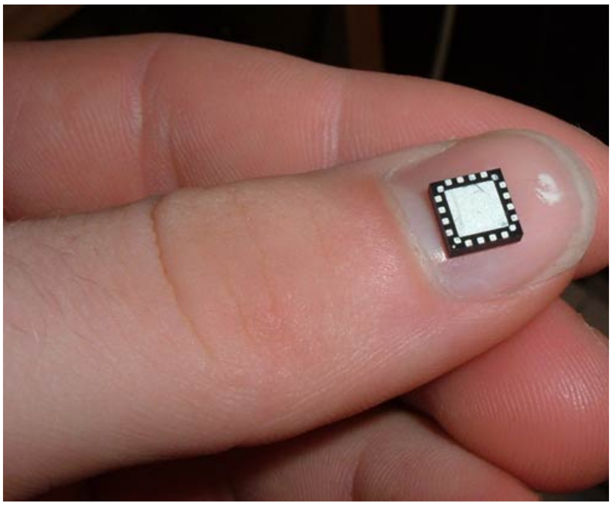


Figure 6.3: Freescale Semiconductor MMA720QT accelerometer.

This device (Acceleglove) connects to the USB port of a computer and its communication is done through a virtualized serial port. That means the controller of the Acceleglove is queried with an ASCII command and it responds back with the values of the sensors. Those values range between 0 and 1023 (10-bit) and represent a linear scale of the acceleration. In the next section a calibration procedure that was needed to correct the values provided by the Acceleglove is described.

6.1.2 Calibration of Acceleglove

Acceleglove provides values in a 10-bit scale. But it does not provide correctly those values in a well described or good calibrated scale. That means it is not possible to know the exact value of acceleration in each axis of each sensor without prior calibration.

This calibration was done recurring to a Python script where the idea was to find the lowest and highest values of acceleration for each axis of each sensor, when only gravity was present. To capture this, the software was recording all the values while a very slow rotation of the sensors in each axis was performed. Figure 6.4 shows the setup used for the calibration.

After this calibration every sensor, whose output was a 10-bit value, now ranged in a

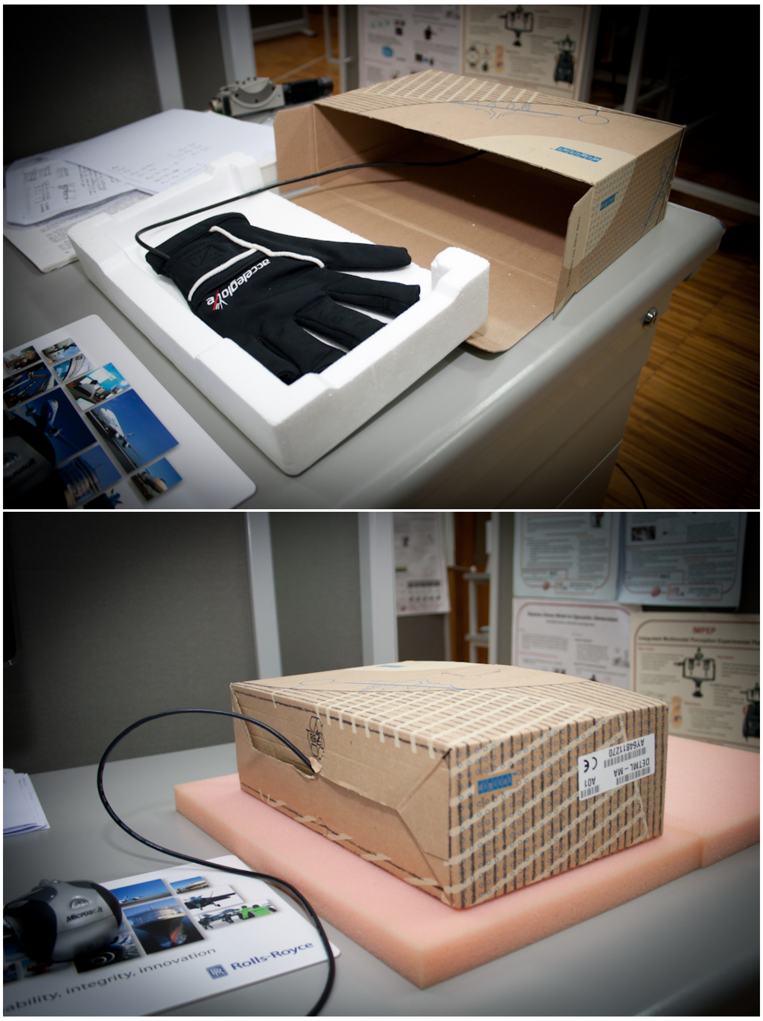


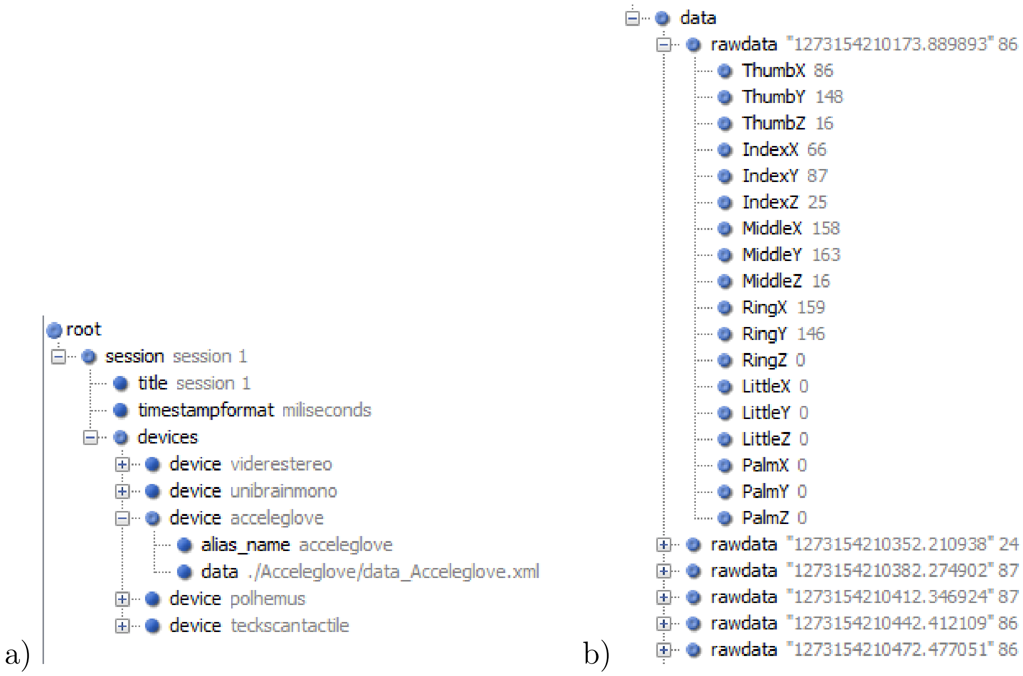
Figure 6.4: Calibration of the Acceleglove sensors.

float scale and between around -1 to around 1, being such values related to the G-force of acceleration. This result, from calibration is shown in Figure 4.2.

6.1.3 XML Structure

In order to organize all the data acquired, it was decided to use a XML file struture. This was used for the data acquisition and is forseen to be capable of dealing with future sensors, such is the case of the HANDLE project.

Each data acquisition created at least two files: root.xml and data_acceleglove.xml. The contents of these xml files are shown in Figure 6.5. The root file is where all the sensors are identified for a given acquisition and the xml_acceleglove is where all the data, with



```
root
  session session 1
    title session 1
    timestampformat miliseconds
    devices
      device viderestereo
      device unibrainmono
      device acceleglove
        alias_name acceleglove
        data ./Acceleglove/data_Acceleglove.xml
      device polhemus
      device teckscantactile
```

```
data
  rawdata "1273154210173.889893" 86
    ThumbX 86
    ThumbY 148
    ThumbZ 16
    IndexX 66
    IndexY 87
    IndexZ 25
    MiddleX 158
    MiddleY 163
    MiddleZ 16
    RingX 159
    RingY 146
    RingZ 0
    LittleX 0
    LittleY 0
    LittleZ 0
    PalmX 0
    PalmY 0
    PalmZ 0
  rawdata "1273154210352.210938" 24
  rawdata "1273154210382.274902" 87
  rawdata "1273154210412.346924" 87
  rawdata "1273154210442.412109" 86
  rawdata "1273154210472.477051" 86
```

Figure 6.5: Structure of XML from the data acquisition.
a) root.xml ; b) data_acceleglove.xml

its timestamp is stored.

6.2 Results of Gesture Recognition

To implement the methods described in chapter 4 the Matlab software was used. This software is well known and allows to more easily implement and test algorithms than any C program would.

Figure 6.6 shows the result of processing a data acquisition and motion classifier.

From Figure 6.6 it is possible to see the motion level classifier working. According to the deviation in the modulus of each sensor a classification of the level of movement is made, meaning Level-1 to be fairly static, while Level-2 refers to a smooth movement and Level-3 refers to a sudden movement that eventually could be used to signal the start and stop of a stream of gestures.

As mentioned in chapter 4, each contiguous subset of Level-1 samples vectors are converted into a single vector, called *frame*.

All the frames are then used to find the quaternion of rotation between each finger and the palm. After calculating that quaternion, as explained in chapter 4, a reprojection

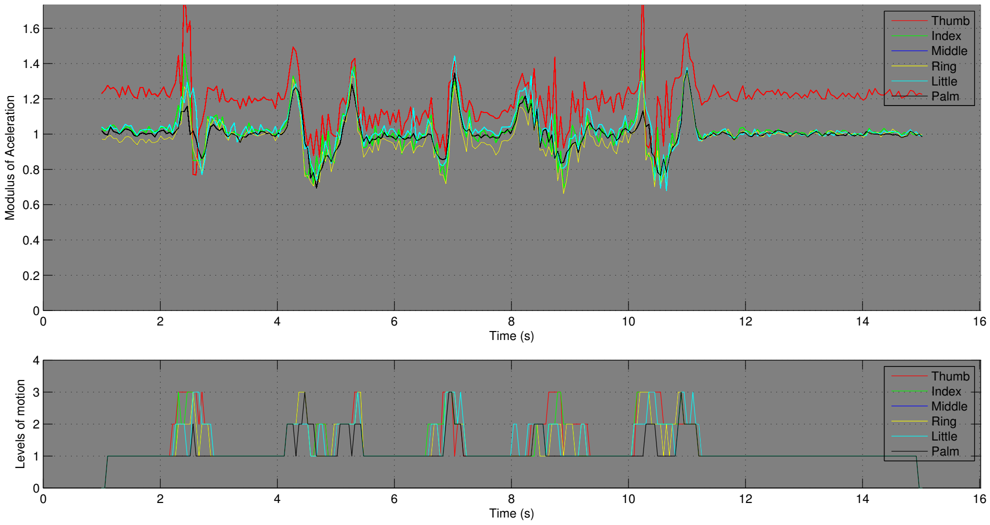


Figure 6.6: Example of the motion classification after a data acquisition.

error is measured for each of the frames used. This reprojection error is shown in Figure 6.7.

The values found on Figure 6.7 allows to say reprojection error is very small, with all the values standing before 5 degrees. This correspond to 3 distinct positions (frames) for which the user had to maintain the same gesture, and a few degrees will not significantly alter the gesture so this result indicates that the estimated relative angular pose values

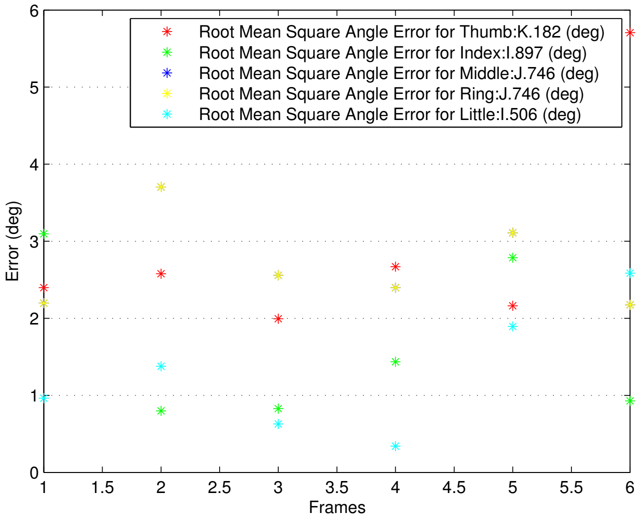
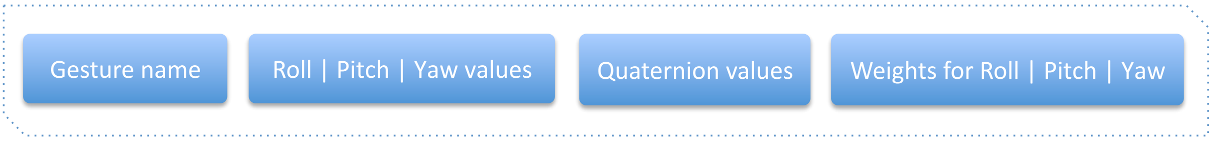


Figure 6.7: Rotation reprojection error after the calculation of the quaternions of rotation.

are suitable for gesture recognition.

Then the frame of the gesture that it is being analysed is compared against a library.

The library structure is shown in Figure 6.8. In this library each entry relates all the information of a single gesture. That includes the name of gesture, the roll, pitch and yaw angles, the quaternion values and weight that define how much impact should roll, pitch or yaw values have in the recognition.



Gesture name

Roll | Pitch | Yaw values

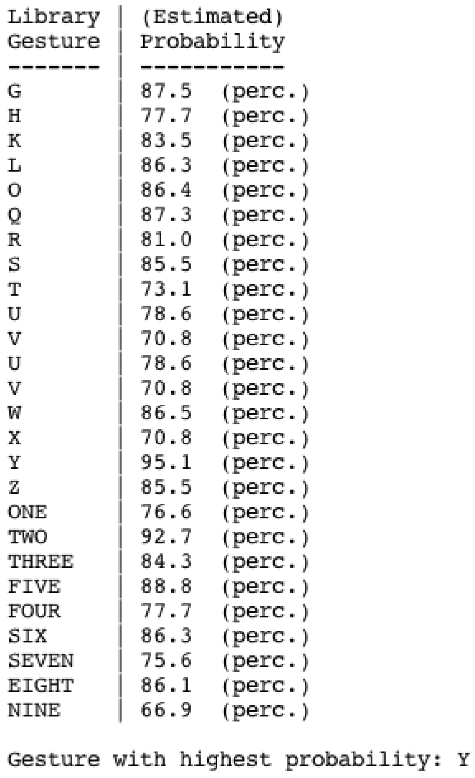
Quaternion values

Weights for Roll | Pitch | Yaw

Figure 6.8: Structure of an entry of the gestures' library.

Comparing a gesture against the library produced the results like the ones shown in Table 6.1. In this table it is possible to see a list of all the gestures included in the library and another column displaying the estimated probability of current gesture being the one of ones in the library list.

The gesture performed that resulted in the comparison shown in Table 6.1, was indeed



```
Library | (Estimated)
Gesture | Probability
------- | -----------
G       | 87.5  (perc.)
H       | 77.7  (perc.)
K       | 83.5  (perc.)
L       | 86.3  (perc.)
O       | 86.4  (perc.)
Q       | 87.3  (perc.)
R       | 81.0  (perc.)
S       | 85.5  (perc.)
T       | 73.1  (perc.)
U       | 78.6  (perc.)
V       | 70.8  (perc.)
U       | 78.6  (perc.)
V       | 70.8  (perc.)
W       | 86.5  (perc.)
X       | 70.8  (perc.)
Y       | 95.1  (perc.)
Z       | 85.5  (perc.)
ONE     | 76.6  (perc.)
TWO     | 92.7  (perc.)
THREE   | 84.3  (perc.)
FIVE    | 88.8  (perc.)
FOUR    | 77.7  (perc.)
SIX     | 86.3  (perc.)
SEVEN   | 75.6  (perc.)
EIGHT   | 86.1  (perc.)
NINE    | 66.9  (perc.)

Gesture with highest probability: Y
```

Table 6.1: Results from recognizing a gesture.

the “Y” gesture. The visual perspective of the gesture is shown in Figure 6.9.

The gesture “Y”, analysed above, has the same finger angular pose relative to the palm.

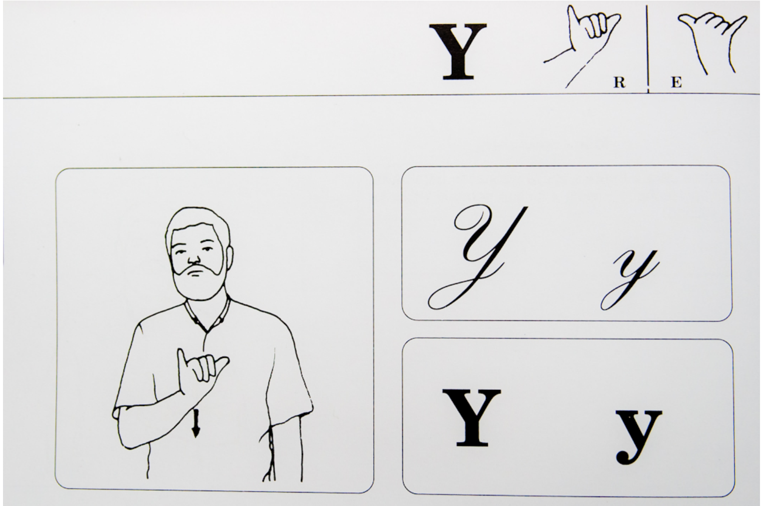


Figure 6.9: The gesture of the letter “Y”.

The only diference is the angular pose of the palm. This is ilustrated in Figure 6.10.

The best test to the recognition process is to test the two similar gestures described in Figure 6.10.

Performing the two gestures separately it was possible to obtain the Table 6.2. This test was conducted several times with the same outcome: the two gestures were always correctly identified.

The values on any of these tables for the estimated probability seem to be always too high but in chapter 4, the normalization of these estimated probabilies were against the *pi* value. Since gestures components are usually much less distant than *pi* it is expectable that the values have a high grade of estimated probability. Yet, the identified gesture

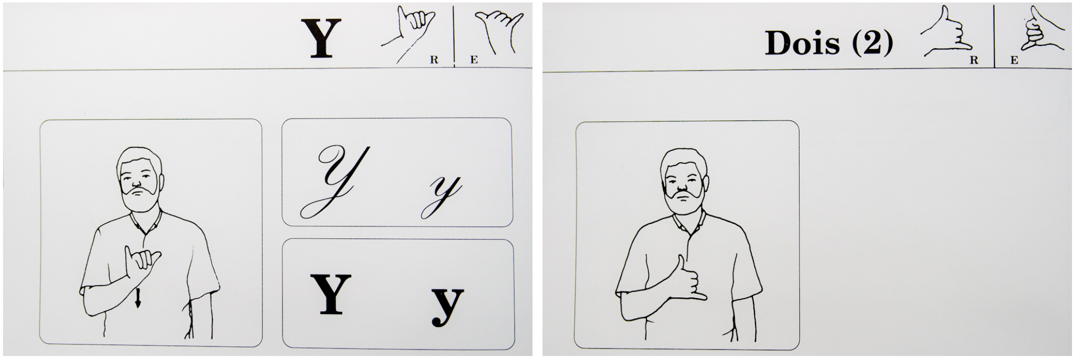
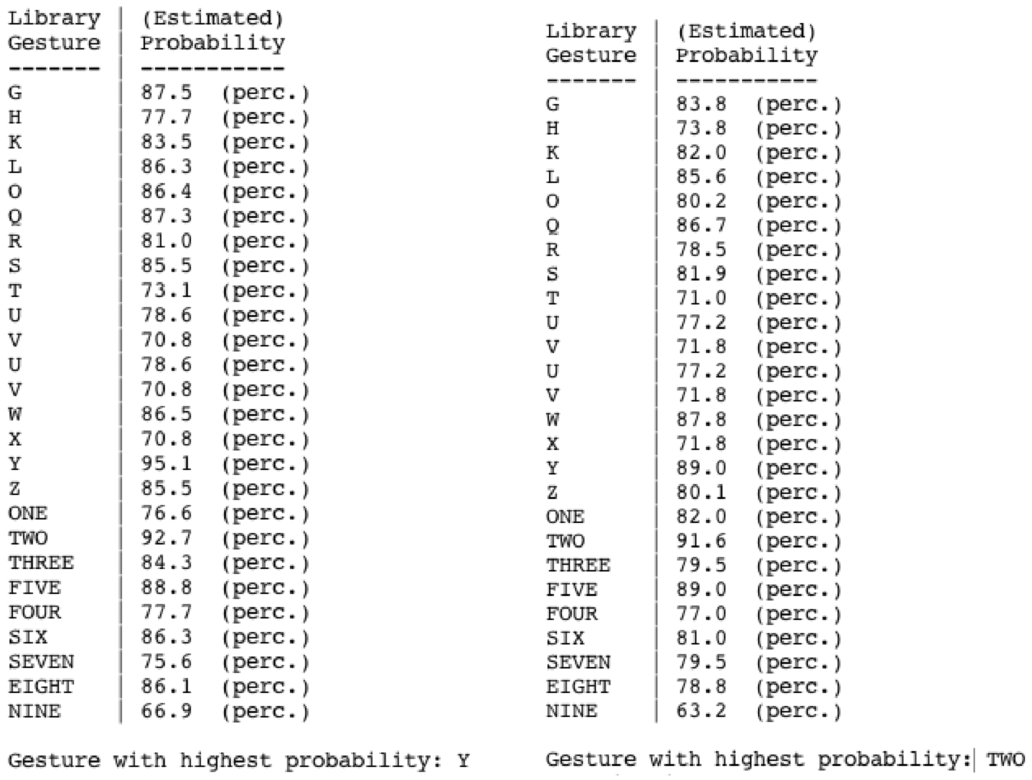


Figure 6.10: Gestures with the same relative angular finger pose to the palm but with a different palm angular pose.



```
Library  | (Estimated)
Gesture  | Probability
-------  | -----------
G        | 87.5  (perc.)
H        | 77.7  (perc.)
K        | 83.5  (perc.)
L        | 86.3  (perc.)
O        | 86.4  (perc.)
Q        | 87.3  (perc.)
R        | 81.0  (perc.)
S        | 85.5  (perc.)
T        | 73.1  (perc.)
U        | 78.6  (perc.)
V        | 70.8  (perc.)
U        | 78.6  (perc.)
V        | 70.8  (perc.)
W        | 86.5  (perc.)
X        | 70.8  (perc.)
Y        | 95.1  (perc.)
Z        | 85.5  (perc.)
ONE      | 76.6  (perc.)
TWO      | 92.7  (perc.)
THREE    | 84.3  (perc.)
FIVE     | 88.8  (perc.)
FOUR     | 77.7  (perc.)
SIX      | 86.3  (perc.)
SEVEN    | 75.6  (perc.)
EIGHT    | 86.1  (perc.)
NINE     | 66.9  (perc.)

Gesture with highest probability: Y
```

```
Library  | (Estimated)
Gesture  | Probability
-------  | -----------
G        | 83.8  (perc.)
H        | 73.8  (perc.)
K        | 82.0  (perc.)
L        | 85.6  (perc.)
O        | 80.2  (perc.)
Q        | 86.7  (perc.)
R        | 78.5  (perc.)
S        | 81.9  (perc.)
T        | 71.0  (perc.)
U        | 77.2  (perc.)
V        | 71.8  (perc.)
U        | 77.2  (perc.)
V        | 71.8  (perc.)
W        | 87.8  (perc.)
X        | 71.8  (perc.)
Y        | 89.0  (perc.)
Z        | 80.1  (perc.)
ONE      | 82.0  (perc.)
TWO      | 91.6  (perc.)
THREE    | 79.5  (perc.)
FIVE     | 89.0  (perc.)
FOUR     | 77.0  (perc.)
SIX      | 81.0  (perc.)
SEVEN    | 79.5  (perc.)
EIGHT    | 78.8  (perc.)
NINE     | 63.2  (perc.)

Gesture with highest probability:| TWO
```

Table 6.2: Table showing the results of the recognition of letter "Y" and number "2", which correspond to two gestures with very similar pose.

clearly stands out from the others, as seen in Table 6.2.

6.3 Results of Visualization

The visualization was structured to allow the representation of the hand pose processed in the recognition process and also directly from a realtime connection to the Acceleglove. The visualization of the results that came from Matlab were possible to represent because Matlab outputed the Roll Pitch and Yaw values for each sensor it processed. Having blender prepared to read those values and running the Python routines defined in blender a correct pose from blender was possible to represent. Figure 6.11 shows a render of the pose from the data shown in Figures 6.6 and 6.7. Again, the relatively small error in calculating the hand pose allowed the visual representation as seen in this figure.

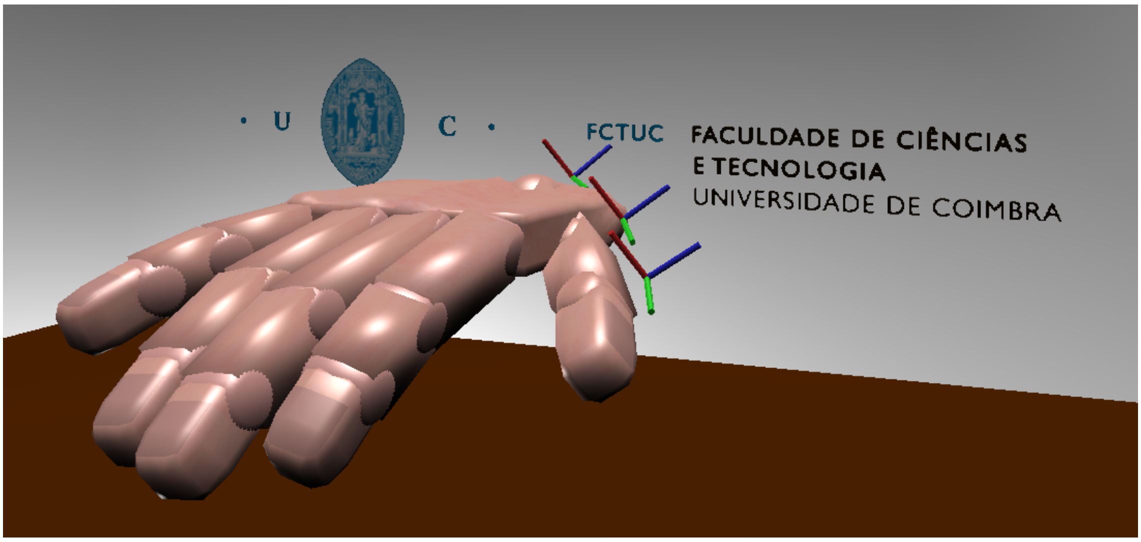
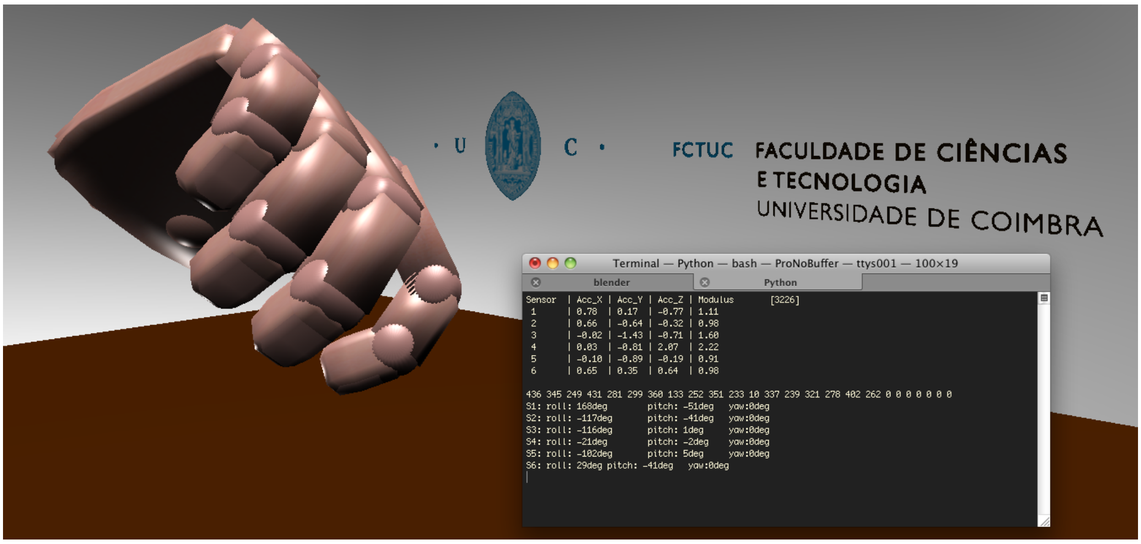


Figure 6.11: A render from Blender showing the pose defined by the recognition step in Matlab. (Refered to the same dataset of Figures 6.6 and 6.7)



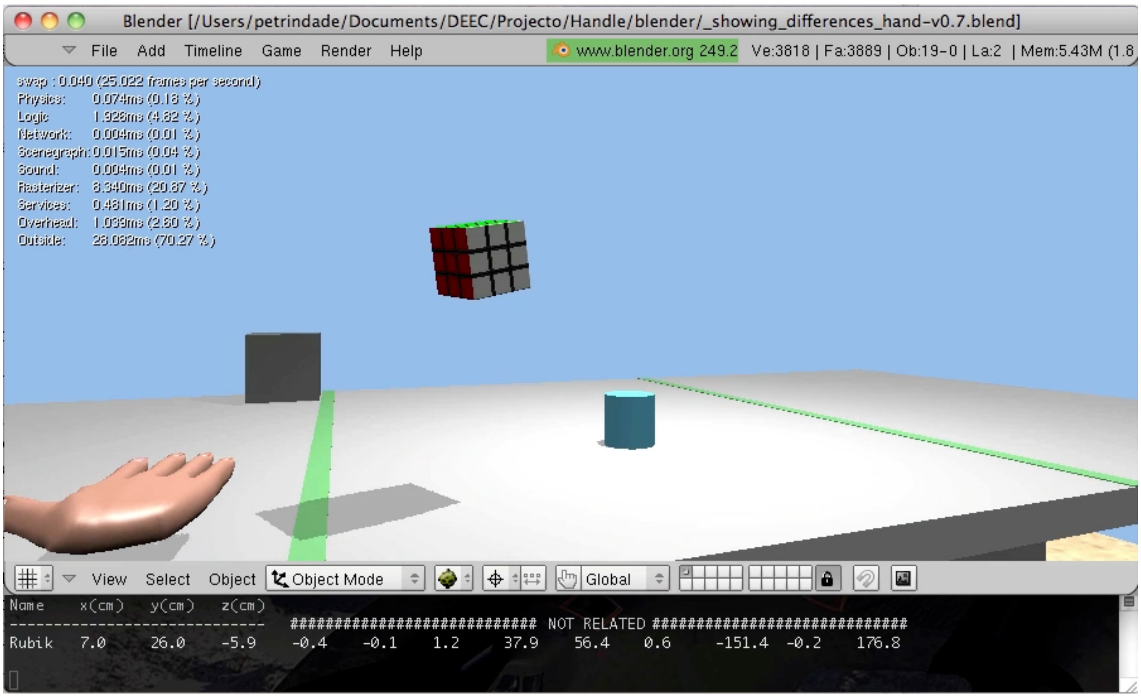
```
Sensor  | Acc_X | Acc_Y | Acc_Z | Modulus       [3226]
 1      | 0.78  | 0.17  | -0.77 | 1.11
 2      | 0.66  | -0.64 | -0.32 | 0.98
 3      | -0.02 | -1.43 | -0.71 | 1.60
 4      | 0.03  | -0.81 | 2.07  | 2.22
 5      | -0.10 | -0.89 | -0.19 | 0.91
 6      | 0.65  | 0.35  | 0.64  | 0.98

436 345 249 431 281 299 360 133 252 351 233 10 337 239 321 278 402 262 0 0 0 0 0 0 0
S1: roll: 168deg        pitch: -51deg   yaw:0deg
S2: roll: -117deg       pitch: -41deg   yaw:0deg
S3: roll: -116deg       pitch: 1deg     yaw:0deg
S4: roll: -21deg        pitch: -2deg    yaw:0deg
S5: roll: -102deg       pitch: 5deg     yaw:0deg
S6: roll: 29deg pitch: -41deg   yaw:0deg
```

Figure 6.12: real-time rendering in Blender with an external script performing the real-time connection and processing of the information from Acceleglove.

When this visualization was performed in real time, an external (to Blender) script was running. This script would provide Blender the pose information for each sensor. Figure 6.12 shows the script window displaying the information that it's processing and sending it to Blender. At the same time it is possible to see blender making rendering at a framerate of about 29fps. At every frame, blender updates the pose of every joint in the hand and renders it. This performance was achieved in a Macbook Pro 2.5GHz from 2009.

One of the goals with visualization was to make it expandable to allow its integration in the HANDLE project. This was also achieved by creating in Blender other objects to allow a broader integration of sensors. Such sensors can be any of those exposed in Figure 6.1. A capture screen from a working integration can be seen in Figure 6.13 where the Rubik Cube, attached to a Polhemus sensor has its real movement mapped in real time in the Blender environment.


Blender [/Users/petrindade/Documents/DEEC/Projecto/Handle/blender/_showing_differences_hand-v0.7.blend]
File
Add
Timeline
Game
Render
Help
www.blender.org 249.2
Ve:3818 | Fa:3889 | Ob:19-0 | La:2 | Mem:5.43M (1.8
View
Select
Object
Object Mode
Global

```
swap : 0.040 (25.022 frames per second)
Physics:     0.074ms (0.18 %)
Logic        1.928ms (4.82 %)
Network:     0.004ms (0.01 %)
Scenegraph:  0.015ms (0.04 %)
Sound:       0.004ms (0.01 %)
Rasterizer:  8.340ms (20.87 %)
Services:    0.481ms (1.20 %)
Overhead:    1.039ms (2.60 %)
Outside:     28.082ms (70.27 %)
```

```
Name    x(cm)   y(cm)   z(cm)
-----------------------------   ############################## NOT RELATED ################################
Rubik   7.0     26.0    -5.9    -0.4    -0.1    1.2     37.9    56.4    0.6     -151.4  -0.2    176.8
```

Figure 6.13: Results of visualization in HANDLE project.

6.4 Summary

This chapter provided the experimental results of the work performed in this thesis, namely the reuslts of implementation of methods like the fingers' relative angular pose estimation and hand gesture recognition. The measuments of the reprojection error allowed to validate these methods. Also it was shown the correct use of the software tools such as Matlab, Python, Blender and C to do the processing of the algorithms and allow the visualization of the gesture. The visualization software was shown to be useful for the HANDLE project, which was also an objective of this visualization.

Chapter 7

Conclusions

The recognition of gestures is a vast area of research. It is an area with enough dimension to have an IEEE anual conference dedicated to the subject. This thesis presented a novel approach by using distributed accelerometers in the hand and recognize the gestures solely by the measurement of acceleration.

In this work a simple gesture recognition method was created. By using an inteligent distribution of accelerometers on the hand it was possible to create an algorithm to recognize a hand gesture. Our approach overcomes some limitations from using gravity as the reference. Gravity provided the information to define the pose but without the information in relation to a rotation in gravity's axis. To overcome this limitation, a novel approach, based on the work of [Lobo, 2007] on relative pose calibration between inertial sensors and cameras, allowed to find the exact 3D angular pose of the fingers in relation to the palm. The recognition is based on static gestures. As future work this recognition should be extended to allow dynamic gesture to be recognized as well.

In this thesis it was also proposed a 3D visualization tool for the hand pose. By creating an inteligent structure this tool was already capable of fulfilling a broader project, like the HANDLE project [Handle, 2010].

The approach presented in this work can be improved in many ways, by addressing dynamic gestures, applying a probabilistic approach for learning the gestures, adding more sensors, and can be an interesting research topic for a future Ph.D. work.

Bibliography

[Acceleglove, 2010] AnthroTronix Acceleglove http://www.acceleglove.com/.

[Barbour, 2003] Barbour, N.M., *Inertial Navigation Sensors*, Sensors & Electronics Technology Panel, Advances in Navigation Sensors & Integration Technology. Held in London, UK; Ankara, TUR; Paris, FRA, 10/20/2003 to 10/28/2003, pp. 2-1 to 2-18. Sponsored by: North Atlantic Treaty Organization (NATO). (Draper Report no. P-4151).

[Baudel, 1993] T. Baudel and M. Beaudoin-Lafon, "*Charade: remote control of objects using free-hand gestures*," Communications of the ACM, vol. 36, no. 7, pp. 28- 35, July 1993.

[Fraden, 2006] Fraden, J., *Handbook of Modern Sensors: Physics, designs, and applications*, 3rd Edition, Springer, 2003.

[Freescale MMA72] Freescale Semiconductor MMA720QT webpage http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MMA7260QT.

[Gestuário-LGP, 1997] Ferreira, A. et al., *"Gestuário - Língua Gestual Portuguesa"*, Secretariado Nacional para a Reabilitação e Integração das Pessoas com Deficiência, 1997.

[Handle, 2010] HANDLE website, 2010 - http://www.handle-project.eu.

[Horn, 1987] B.K.P Horn. *Closed-Form Solution of Absolute Orientation Using Unit Quaternions.* Journal of the Optical Society of America, 4(4):629–462, April 1987.

[Jones et al., 2006] Jones, L., Lederman, S., "*Human Hand Function*", Oxford University Press, 2006.

[Kang Li et al., 2010] Kang Li, I-Ming Chen, Song Huat Yeo, *"Design and Validation of a Multi-finger Sensing Device Based on Optical Linear Encoder"*, 2010 IEEE International Conference on Robotics and Automation Anchorage Convention District May 3-8, 2010, Anchorage, Alaska, USA.

[Kapandji, 1970] Kapandji, I. A. (1970). *The physiology of the joints: Vol. 1. Upper limb (2d ed., L. H. Honore, Trans.)*. Edinburgh: E & S Livingstone.

[Khan et al., 2009] S Khan, G Gupta, D Bailey, S Demidenko, C Messom, *Sign Language Analysis and Recognition: A Preliminary Investigation*, Image and Vision Computing New Zealand, 2009. IVCNZ '09. 24th International Conference.

[Lobo, 2006] Jorge Nuno de Almeida e Sousa Lobo, "*Integraton of Vision and Inertial Sensing*", PhD Thesis, Supervisor: Jorge M. M. Dias, December 2006.

[Lobo, 2007] Lobo, J., Dias, J., *Relative Pose Calibration Between Visual and Inertial Sensors*, Int. J. Rob. Res., vol. 26, p.561-575, 2007.

[MMA72 Technical Data] *MMA7260T Technical Data: ±1.5g - 6g Three Axis Low-g Micromachined Accelerometer*, Freescale Semiconductor, Rev 5, 3/2008.

[Mullen, 2007] Mullen, T., *Introducing Character Animation with Blender*, Wiley Publishing Inc., 2007.

[Mullen, 2008] Mullen, T., *Bounce, Tumble and Splash! Simulating the Physical World with Blender 3D*, Wiley Publishing Inc., 2008.

[Niezen, 2008] Niezen, G., *The Optimization of Gesture Recognition Techniques for Resource-Constrained Devices,* MSc Thesis, University of Pretoria, 2008.

[Oliver, 2008] Oliver, J., *Buffering Bergson: Matter and Memory in 3D Games*, 2008.

[Oxford, 2000] *Oxford Advanced Learner's Dictionary,* Sixth Edition, Oxford University Press, 2000.

[Takayuki et al., 2007] Takayuki, H., Ozake, S., Shinoda, H., *Three-Dimensional Shape Capture Sheet Using Distribuited Triaxial Accelerometers,* Fourth International Conference on Networked Sensing Systems, 2007.

[Taylor et al., 1970] Taylor, C. L., & Schwarz, R. J. (1970). *The anatomy and mechanics of the human hand.* In Selected Articles From Artificial Limbs, January 1954-Spring 1966 (pp. 49–62). Huntington, NY: Kreiger.

[Wang et al., 2009] Robert Y. Wang and Jovan Popovic, *Real-Time Hand-Tracking with a Color Glove*, ACM Transaction on Graphics (SIGGRAPH 2009), 28(3), August 2009.

[Weinberg et al., 2006] Weinberg, M. S., Kourepenis, A., *Error Sources in In-Plane Silicon Tuning-Fork MEMS Gyroscopes*, Journal of Microelectromechanical Systems, vol. 15, No. 3, June 2006.