**Marília da Silveira Gouveia Barandas**

Licenciatura em Ciências de Engenharia Biomédica

# Range of Motion Measurements Based on Depth Camera for Clinical Rehabilitation

Dissertação para obtenção do Grau de Mestre em
Engenharia Biomédica

Orientador : Hugo Gamboa, Professor Auxiliar, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Co-orientador : José Manuel Fonseca, Professor Auxiliar, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Júri:

Presidente: Prof. Doutor Mário Secca

Arguente: Profª. Doutora Isabel Nunes

Vogais : Prof. Doutor Hugo Gamboa
Prof. Doutor José Manuel Fonseca

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

**Dezembro, 2013**

**Range of Motion Measurements Based on Depth Camera for Clinical Rehabilitation**

*"If A is a success in life, then A equals x plus y plus z. Work is x; y is play; and z is keeping your mouth shut."*

***Albert Einstein***

# Acknowledgements

First, I would like to thank my supervisor, Hugo Gamboa, for welcoming at *PLUX - Wireless Biosignals, S.A* and for agreeing to work with me during the last academic year. This experience gave me different tools regarding the business environment. I am very thankful to Professor José Manuel Fonseca, my co-supervisor, for all the support, motivation and availability that greatly contributed in achieving the proposed objectives for this dissertation. His help was essential! He always found a better solution for my problems, even when I thought it was no longer possible. I also feel the need to thank him for being such an excellent professor.

I am also grateful to all PLUX workers for welcoming in their team and for creating a healthy environment every day. A special thanks to António Jordão for his patience and dedication in helping me whenever I needed. I also want to thank my thesis' colleagues that shared this experience with me. My biggest thanks goes to Mafalda Camara for the friendship and support throughout the development of this work. Thank you for making my days a lot better, without you everything would have been more difficult. I want to thank Inês Machado for making me laugh every day with her jokes.

A special thanks to Ana Trindade for keeping me company on the last 5 years. With her presence everything was easier, a simple phone call was enough to solve an entire day's problem. Thank you for the friendship and for being part of my life. Our stories will forever stay in my mind as one of the best phases of our lives.

I must thank Marcos Jerónimo, Filipe Gouveia, Diogo Capelo, Cristina Viegas, Rui Ventura, Paulo Poção and Luis Pedroso for helping me reviewing this work. Their help was very important for me.

My sincere appreciation to my (*Peniche*) friends for letting me do some work and at

the same time having so much fun during our awesome holidays. I love you all.

*Por fim agradeço à minha família que sempre me apoiou e acreditou em mim. À minha mãe e às minhas irmãs um muito obrigada por fazerem parte da minha vida todos os dias. Não tenho palavras para descrever o quanto gosto de vocês. Ao meu pai tenho também que agradecer pelo orgulho demonstrado por mim e pelo meu trabalho, apesar da distância sei que estás sempre presente à tua maneira. À Bárbara e ao Marcos um obrigada muito especial por serem também uma parte da minha família. Bárbara, estes 17 anos contigo fizeram de mim uma pessoa melhor. Obrigada por seres aquela amiga que eu nunca esquecerei! Marcos, mais que um namorado és um grande amigo, obrigada por toda a ajuda e paciência que tiveste comigo.*

# Abstract

In clinical rehabilitation, biofeedback increases the patient's motivation which makes it one of the most effective motor rehabilitation mechanisms. In this field it is very helpful for the patient and even for the therapist to know the level of success and performance of the training process. The human motion tracking study can provide relevant information for this purpose. Existing lab-based Three-Dimensional (3D) motion capture systems are capable to provide this information in real-time. However, these systems still present some limitations when used in rehabilitation processes involving biofeedback. A new depth camera - the Microsoft Kinect$^{TM}$ - was recently developed overcoming the limitations associated with the lab-based movement analysis systems. This depth camera is easy to use, inexpensive and portable.

The aim of this work is to introduce a system in clinical practice to do Range of Motion (ROM) measurements, using the Kinect$^{TM}$ sensor and providing real-time biofeedback. For this purpose, the ROM measurements were computed using the joints spatial coordinates provided by the official Microsoft Kinect$^{TM}$ Software Development Kit (SDK) and also using our own developed algorithm. The obtained results were compared with a triaxial accelerometer data, used as reference.

The upper movements studied were abduction, flexion/extension and internal/external rotation with the arm at 90 degrees of elevation. With our algorithm the Mean Error (ME) was less than 1.5 degrees for all movements. Only in abduction the Kinect$^{TM}$ Sketelon Tracking obtained comparable data. In other movements the ME increased an

order of magnitude. Given the potential benefits, our method can be a useful tool for ROM measurements in clinics.

**Keywords:** Biofeedback, Depth Camera, Range-of-Motion, Rehabilitation

---

# Resumo

Em reabilitação clínica, o *Biofeedback* aumenta a motivação do paciente, tornando-o um dos mecanismos de reabilitação motora mais eficazes. Neste campo é muito útil para o paciente e mesmo para o terapeuta saber o nível de sucesso e o desempenho do processo de treino. O estudo da captura de movimento humano pode fornecer informações relevantes para esta finalidade. Os sistemas de laboratório existentes de captura de movimento 3D são capazes de fornecer esta informação em tempo real. No entanto, estes sistemas apresentam ainda algumas limitações quando usados em processos de reabilitação envolvendo *biofeedback*. Uma nova câmara de profundidade - o Microsoft Kinect - foi recentemente desenvolvido ultrapassando as limitações associadas com os sistemas de análise de movimento em laboratório. Esta câmara de profundidade é fácil de usar, barata e portátil.

O objectivo deste trabalho é introduzir um sistema para medições da amplitude de movimento na prática clínica, usando o sensor do Kinect e fornecendo *biofeedack* em tempo real. Para este efeito, as medições de amplitude de movimento foram calculadas usando as coordenadas espacias das articulações fornecidas pela Microsoft Kinect SDK oficial e também usando o nosso próprio algoritmo desenvolvido. Os resultados obtidos foram comparados com os dados de um acelerómetro triaxial, usado como referência.

Os movimentos dos membros superiores estudados foram a abdução, flexão/extensão e rotação interna/externa com o braço a 90 graus de elevação. Com o nosso algoritmo o ME foi menos de 1.5 graus em todos os movimentos. Apenas na abdução o algoritmo

da Microsoft obteve dados comparáveis. Nos outros movimentos o ME aumentou uma ordem de magnitude. Tendo em conta os potenciais benefícios, o nosso método pode ser uma ferramenta útil para medições da amplitude do movimento em clínicas.

**Palavras-chave:** *Biofeedback*, Câmara de Profundidade, Amplitude de Movimento, Reabilitação

---

# Contents

# List of Figures

# List of Tables

# Acronyms

**2D** Two-Dimensional

**3D** Three-Dimensional

**AAPB** Association for Applied Psychophysiology and Biofeedback

**ACC** Accelerometry

**AM** Active Marker

**BRS** Biofeedback Research Society

**COF** Centre of Force

**COP** Centre of Pressure

**EMG** Electromyography

**IR** Infrared

**LED** Light-Emitting Diode

**ME** Mean Error

**PM** Passive Marker

**RGB** Red-Green-Blue

**RMS** Root-Mean-Square

**ROM** Range of Motion

**SDK** Software Development Kit

**sEMG** Surface Electromyography

**SD** Standard Deviation

# 1

# Introduction

## 1.1 Motivation

Nowadays, biofeedback is one of the most effective motor rehabilitation mechanisms [7]. Involving the user in his rehabilitation process gives a real perception of his evolution and increases the patient's motivation. This factor alone creates a considerably faster recuperation progress.

The emerging technology offers solutions, like the Microsoft Kinect$^{TM}$, adequate to the evaluation of the human position and the body's movement. Through processing algorithms applied to depth images, it is possible to detect the joints' positions as well as the angles executed during the subject's movement. Nowadays, several technologies exist capable of evaluating the aforementioned parameters. Nonetheless, these systems still present limitations when used in rehabilitation processes involving biofeedback.

Thus, in the context of biofeedback, the Kinect$^{TM}$ sensor offers information of great relevance that can improve the quality of life of people with some incapacity levels. These capabilities of depth cameras should be explored with a real-time application, providing patients details of their own movement and improvement.

## 1.2 Objectives

The main purpose of this thesis is to introduce a simple system in a clinical environment for ROM measurements, since there is currently a lack of practical and cost-efficient methods available for this purpose [8]. Generally, in clinics, these ROM measurements are done manually with a goniometer. However, their results present a 5-10 degrees error [9]. Thus we propose to improve these results with a faster and easier to use method that can provide real-time biofeedback.

In order to fulfil this goal, we needed to use the depth map information provided by a Depth Camera - the Microsoft Kinect$^{TM}$ - and developed an algorithm capable of detecting anatomical landmarks to compute the ROM measurements in glenohumeral movements[1].

## 1.3 State of the art

Human motion tracking for clinical applications has been an active research topic since the 1980s [10]. In this field, human movement analysis has been done with different sensing principles such as magnetic, inertial and optical sensors. However, these systems are of limited use for clinical applications in rehabilitation processes. Optical systems with Active Marker (AM) or Passive Marker (PM) achieve high accuracy and large capture volumes. However, these systems are difficult to use in real-world applications due to their complexity, bulky size and space requirements [2]. Magnetic tracking systems are not bulky and require less space but the rather cumbersome cables carried by the patient and the limited accuracy due to magnetic field distortions caused by large metal objects makes its use very limited in clinics [2]. Inertial measurement systems are easy to use, have high sensitivity and large capture areas making them a very desirable alternative for portable 3D motion capture. However, they are also susceptible to measurement errors mainly due to nearby metals, when magnetometers are present in their constitution [11].

Recently a new sensor based on depth images was developed by a company in Israel, named PrimeSense. This sensor, called Kinect$^{TM}$, gained the attention of many researchers due to its low cost, portability and because it does not require body markers to determine anatomical landmarks. Several studies have been performed using the

[1] Movements of the shoulder joint

Kinect$^{TM}$ sensor and their applications have been diverse.

The 3D depth accuracy of the Kinect$^{TM}$ has been evaluated by several researchers [11][3]. Tilak Dutta [11] conducted a study to determinate the range, field of view and accuracy of the sensor. The gold standard reference used in this study was the Vicon motion capture system (Optical System with PM) that has a positional accuracy of 0.1 mm and angular accuracy of 0.15 degrees [12]. He concluded that the accuracy of a Kinect$^{TM}$ motion capture system would be at least an order of magnitude less than that of a Vicon system. However, this study was based on a static scene. The results obtained from the Kinect$^{TM}$ field of view and range are in agreement with Microsoft's advertising [13]. The study of Khoshelham et al. [3] showed that the Kinect$^{TM}$ can provide accuracy of depth reconstruction from a few millimetres at short distance up to about 4 cm at the maximum range of the sensor.

The depth map information provided by Microsoft Kinect$^{TM}$ has been used for different applications. Oikonomidis et al. [14] presented a novel model based on tracking the full pose of a hand in complex articulation using data acquired by a Kinect$^{TM}$ sensor. Also using depth images from the Kinect$^{TM}$ sensor, Xia et al. [15] proposed a model based approach which detects humans in all poses and provides an accurate estimation of the person's whole body contour. Thus, their algorithm can be an alternative to conventional human detection that is mostly done with regular cameras. In a study conducted by Gama et al. [16] a prototype was developed that recognised wrong movements and when correct therapeutic exercises were done, showed its efficiency. Thus this method, that avoids wrong movements, can prevent lesions and optimises the recuperation.

Like the last study mentioned, numerous attempts have been made to use the Kinect$^{TM}$ sensor for rehabilitation. To use it as a motion capture system a specific software like OpenNI, PrimesenseNITE and the official Microsoft Kinect$^{TM}$ SDK is normally used. Clark et al. [17] developed a method for assessment of the postural control using the Kinect$^{TM}$ sensor in combination with the Microsoft SDK and their results provided comparable data to a Vicon System when assessing the anatomical landmark position and angular displacement data during commonly performed clinical tests of postural control. However, the results of the study conducted by Obdržálek et al. [18], showed that the accuracy of the joint estimation obtained was comparable to the optical system used, but only in a controlled body posture such as standing and exercising arms. In general postures, the variability of their implementation of the pose estimation was about 10 cm,

indicating that the measurements could be used to assess general trends in the movement, but when a quantitative estimation is needed, an improved skeletonization with an anthropometric model is necessary. These results were obtained at a range of 3 m.

Other approaches used the combination of the inertial sensors with the Kinect™, in order to compensate the limitations of each other. Lanari Bó et al. [19] developed a method based on Kalman filtering to estimate joint angles with inertial sensors and its integration with Kinect™. The 3D joint position reconstruction with Kinect™ was based on the module developed by PrimeSense and in their work they studied three movements: sit to stand, squat and shoulder abduction/adduction. They concluded that when larger sensor errors are present, suitable calibration is required and for this reason they used the Kinect™ for conducting brief online calibration periods. Thus the Kinect™ provides the possibility of calibrating the inertial sensors in real-time, enables simpler initialisation procedures and a better visualisation of the estimated angles. However, the Kinect™ showed some limitations such as conditioned workspace and estimation errors due to occlusions which can be compensated by the inertial sensors. With faster movements, the Kinect™ has some problems due its lower frame rate. Another study using these two sensors was conducted by Hondori et al. [20] to monitor the patient's intake gestures and distinguishing between healthy and paralysed body sides. They measured joints positions, angular displacements and the acceleration of the object which is held by the subject. Their research can be used to generate feedback on the patient's health status in the post-stroke telerehabilitation.

The ROM measurements have been studied too. Fernandez-Baena et al. [21] agreed with the previous studies mentioned and used OpenNI and PrimeSenseNITE. To evaluate the accuracy of the Kinect™ they also used an optical system as a reference and obtained ME between 5 and 13 degrees. They claim that the Kinect™ has several advantages and the precision ranks obtained are sufficient to use it for rehabilitation treatments. They developed an application for knee rehabilitation that automatically counts repeated movements and validates the quality of such motion. They also defend that the precision of the Kinect™ can be increased by imposing some restrictions such as fixing length for the bones or working directly with the depth map information. Kitsunezaki et al. [9] studied three applications for the physical rehabilitation using the Kinect™, where one of them was *ROM measurement*. They used the Microsoft Kinect™ SDK and compared

the angles computed with manual values obtained by an assistant with a protractor, resulting in differences from 2% to 9% for the upper limb.

We believe that the algorithms developed by Microsoft and PrimeSense for skeleton tracking are useful for some applications. However, when the required pose occludes some body parts or when the precision of the ROM measurements is required, these algorithms are insufficient. To overcome these problems some researches are working with the depth information to create their own modules. In [22] the authors created their own tracking module using the depth sensing data obtained from the Kinect$^{TM}$. However they only detected and tracked the head, the shoulders and the hands, which is insufficient to do ROM measurements. Thus, we propose to measure them working directly with the depth map information to get a better approximation of the joints' positions.

## 1.4 Thesis Overview

This thesis is organized in six chapters and one appendix, and its structure is represented schematically in Figure 1.1.

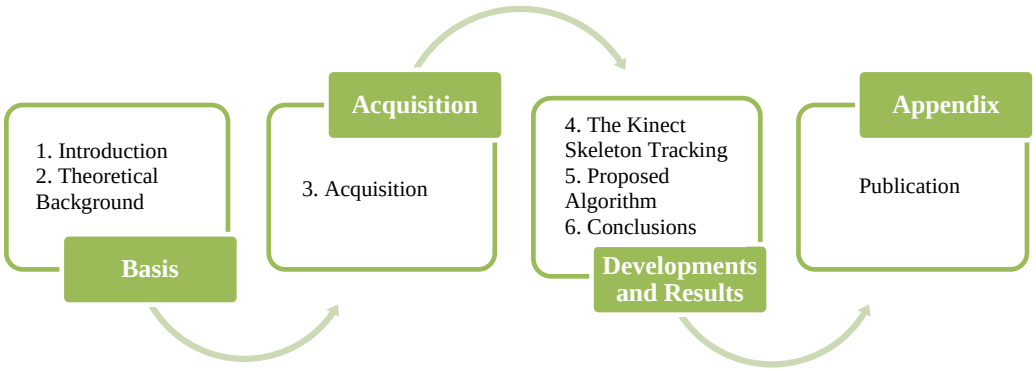



Figure 1.1: Thesis overview.

In the present chapter the theme and motivation that lead to the development of this thesis are introduced, as well as the main objectives and the state of the art. Theoretical concepts are presented in Chapter 2 to contextualize the reader with the main principles involved in this work. These two chapters form the basis for the development of this thesis.

Chapter 3 focuses on the procedures to do the data collection with all systems used and constitute the acquisition part of the thesis. A triaxial accelerometer was used as reference to validate the two approaches to compute ROM measurements, the Kinect$^{TM}$ Skeleton Tracking and our proposed algorithm.

The last three chapters address the development and results obtained for both approaches analysed. Chapter 4 presents the development of an application to receive the joints' positions provided by the Microsoft Kinect$^{TM}$ SDK and compute the angles executed during the movement. The results demonstrated the need for creating a new algorithm able to overcome the limitations of the previous. In Chapter 5, a new algorithm to compute ROM measurements is proposed as well as its validation with a triaxial accelerometer. The last chapter presents an overview of the developed work, results and some future work suggestions.

There is one additional appendix that contains the paper submitted in the context of this research work.

The application to receive the Kinect$^{TM}$ information was developed in C# using the $Visual\ Studio$ 2010 and the official drivers provided by Microsoft. The proposed algorithm was developed in $Python$ and the image processing techniques required the use of the $OpenCV$ library.

# 2

# Theoretical Background

In this chapter, the main theoretical concepts that were used in this research are presented. Contextual information about Biofeedback, the systems that can provide human motion tracking and the image processing techniques will be addressed.

## 2.1 Biofeedback

In 1969, the term biofeedback was officially defined at the first meeting of the Biofeedback Research Society (BRS). In 1988, BRS became the Association for Applied Psychophysiology and Biofeedback (AAPB) and defined biofeedback as "a process that enables an individual to learn how to change physiological activity for the purposes of improving health and performance. Precise instruments measure physiological activity such as brainwaves, heart function, breathing, muscle activity, and skin temperature. These instruments rapidly and accurately 'feed back' information to the user. The presentation of this information — often in conjunction with changes in thinking, emotions, and behaviour — supports desired physiological changes. Over time, these changes can endure without continued use of an instrument".

The purpose of this research is to use the biofeedback in rehabilitation to improve certain motor functions, so we are focusing only in motor biofeedback from now on.

In cases of rehabilitation, it is very helpful for the patient and even for the therapist to

know the level of success and performance of the training process. This performance is often derived from afference and reafference and can also be described as intrinsic feedback. Thus, the feedback can be divided into two categories, intrinsic feedback and extrinsic feedback. The intrinsic feedback is generated by movement itself, proprioception or vision of the moving limb, but also sound of footsteps. On the other hand, extrinsic or augmented feedback may be provided additionally by an outside source, such as a therapist [7]. We are interested to improve the extrinsic biofeedback, since it is important for learning some motor tasks. Furthermore, when patients know about their progress, usually this is translated in increasing their motivation, and some researches claim that a growth in motivation is translated into a greater effort during task practice [7]. Pursuing and achieving goals are also reasons to keep the patients motivated. So, the required measurements to compare the current status with the desired goal, are achieved by a biofeedback equipment.

The clinical applications of biofeedback are increasingly being used in rehabilitation medicine for the treatment of many varied disorders by providing visual or audio feedback. The most usual biofeedback technique is the electromyographic that supplies information about muscle contraction. However, there are other techniques that are used for feedback purposes in rehabilitation medicine as force, position and joint angle monitors for recording body posture and movement [23].

### 2.1.1 Electromyography feedback

Electromyography (EMG) was employed as a primary biofeedback source to down-train activity of a hyperactive muscle or up-train recruitment of a weak muscle, thus improving muscular control over a joint [24]. The EMG signal can be recorded in two different ways. With one of them, it is necessary to insert a needle electrode into the muscle tissue, that is called intramuscular EMG, the other, known as sEMG, uses an electrode placed on the skin over a muscle (see Figure 2.1). In biofeedback only sEMG is used and thus in this research the EMG signals are all acquired using this non-invasive method.

When a muscle needs to begin its activity, the muscle cells are activated, electrically or neurologically, and an electrical signal is generated that can be detected by the electrodes of the EMG. This specific signal is called the electromyographic signal and is generated by the motor neurons. A motor unit is constituted by a motor neuron and all the muscles fibres that it innervates. Thus, to contract a muscle, a group of motor units becomes active

and causes its own electric signal which leads to a contraction.

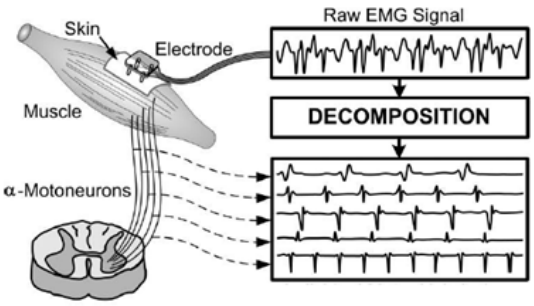



Figure 2.1: sEMG recording with an electrode placed on the surface of the skin above the muscle and an example of a raw signal with the signal decomposition to obtain individual motor unit action potentials [1].

### 2.1.2 Pressure or ground reaction force feedback

Force platforms or force plates were often used as feedback sources during balance retraining programs, providing the locus of the Centre of Force (COF) or Centre of Pressure (COP) to patients, as well as training protocols to enhance stance symmetry, steadiness and dynamic stability. Researches were made with these platforms indicating that the patients who received the biofeedback training were having a faster progress than the patients who received a traditional physical therapy [25].

There are two types of force platforms that are widely used, one of them based on piezo-electric transducers and the other one based on strain-gauge transducers. A standard force platform consists of four triaxial force sensors measuring the three orthogonal components of the applied forces, the vertical torque and the anterioposterior or mediolateral coordinates of the COP. Figure 2.2 shows a ground reaction force and moment outputs for a force platform that is based on strain-gauge transducers.

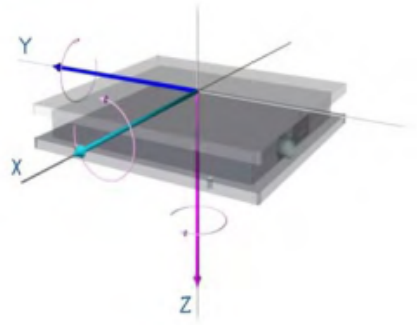



Figure 2.2: Ground reaction force and moment outputs for the AMTI force platform [2].

### 2.1.3 Angular or positional feedback

Angular or positional biofeedback was used to improve patients' ability to self-regulate the movement of a specific joint. To measure joint angles, electrogoniometer devices are used. These produce an output voltage proportional to the angular change between the two attached surfaces. The angles can be measured in a single plane or for bi-planar movements and do not require attachment to joint centres, having instead attachments for the two segments spanning the joint. There are different kinds of electrogoniometers such as potentiometric and flexible electrogoniometers and fiber-optic goniometers. The most commonly used in medicine is the flexible electrogoniometer, shown in Figure 2.3, that provides free joint's motion.

Figure 2.3: Biaxial electrogoniometers from Biometrics Ltd.

## 2.2 Human motion tracking

The ability to analyse human movement is an essential tool of biomechanical analysis for clinical applications and has been an active research topic since the 1980s. Human movement tracking systems are expected to generate real-time data dynamically representing the changes in pose of a human body, or a part of it, based on well developed motion sensing technologies [10].

There are three main types of 3D motion analysis systems: magnetic, inertial and optical systems. In the following sections there is a description of the these systems.

### 2.2.1 Magnetic systems

Magnetic tracking systems, as the name implies, use magnetic fields and include a transmitter and a receiver. The transmitter emits magnetic fields and the receiver detects the magnetic fields emitted by the transmitter. This system has sensors and each one of them returns six degrees-of-freedom, three coordinates and three angles, in real-time. The size,

sampling rate and the fact that line of sight between sensor and receptor is not required constitute advantages of these kind of systems. In several studies [10][26] [27], the main cited disadvantages are the short operating ranges, the system accuracy when a large volume of operation is desired and the magnetic field itself, which gets distorted by several kinds of metals. However, the field of operation is actually comparable with, or better than, some optical systems [2], but magnetic fields lose strength as the receiver moves away from the transmitter. The accuracy of these systems should be confirmed in every setting because different physical environments will affect the magnetic field differently. The current research is focused on tackling the effect of the distortion of magnetic fields and several studies show that some algorithms are able to minimise this effect [10].

The two main commercially available magnetic systems are produced by Ascension and Polhemus. Zachmann et al. [26] compared the susceptibility of the Polhemus's Fastrak and the Ascension's Flock-of-Birds and the results pointed to the Ascension system as less susceptible to ferro-magnetic metals than Polhemus.

### 2.2.2 Inertial systems

Inertial sensors are constituted by accelerometers and gyroscopes and provide information about position and acceleration of the body part where the sensor was placed. Some of them also include magnetometers. Besides their low cost and ease of use, other advantages of the inertial systems are their high sensitivity and large capture areas that make their use frequent for full-body human motion detection. However, the data from the sensor can not be correctly determined due to the fluctuation of offsets and measurement noise, leading to integration drift. Therefore, designing drift-free inertial systems is the main target of the current research [10].

Zhou et al. [28][10] wrote several papers on this topic and presented in 2010 [29] a new inertial-sensor-based monitoring system for measuring the movement of human upper limbs using two wearable inertial sensors with triaxial accelerometers and gyroscopes. Their results showed that the measurement drift in segment orientation is dramatically reduced after a Kalman filter is applied to estimate inclinations using accelerations and turning rates from gyroscopes.

Instead of using a system that integrates a set of accelerometers, gyroscopes and magnetometers, a single triaxial accelerometer is widely used for movement kinematic analysis. Vibration and tilt analysis, obtaining motion patterns of various tasks and posture

sensing are examples of applications of Accelerometry (ACC). These sensors are capable of estimating the acceleration along the tri-axis and detecting the acceleration's magnitude or direction change rate. In order to calibrate the gain and the zero offset, the propriety that an accelerometer at rest with its sensitive axis pointing toward the centre of the Earth will have an output equal to 1 g is commonly used [30].

The two main commercially available systems with inertial sensors are MT9 of Xsens Motion Tech and G-link of MicroStrain. These systems are similar and both have wireless properties, indicating that they are not limited in space. The MT9 based system is a digital measurement unit that measures 3D rate-of-turn, acceleration and earth-magnetic field and the G-Link system is a high speed, triaxial accelerometer node, designed to operate as part of an integrated wireless sensor network [10].

### 2.2.3 Optical systems

Optical systems can be further divided into passive and active systems depending on the type of the markers that each system uses. There are also cameras based on depth images that do not need any kind of markers .

#### 2.2.3.1 Passive Marker

PM systems use reflective markers attached to the patients, enabling them to reflect the light by an external source. Reflections from the markers are tracked using multiple video cameras. Infrared (IR) flash illuminators surround each camera lens sending out pulses of IR light that are reflected back into the lens from the markers. This system requires at least 2/3 cameras but 6/8 is the recommended minimum because the markers are often obscured from one or more cameras or their trajectories cross [2]. So, the space required for the operation of these systems is a big limitation on their use by clinics. Furthermore, marker slippage, markers leaving the volume, bad volume calibration, ghosting, bad threshold level, stray light and cost are some of the problems of PM systems.

Still, PM systems are the most used 3D motion analysis systems. The main advantages of these systems are the highly configurable marker setup achievable, large capture volumes and the fact that no cables or battery packs are required.

#### 2.2.3.2 Active Marker

In the case of AM systems, each marker is a IR Light-Emitting Diode (LED), so the markers generate their own light, unlike the PM. This system also needs markers attached to the patient. One of the major advantages over PM is that, with only one marker flashing at any one time, this system can automatically identify and track each marker. However, after sampling the first marker, it must sample all others before it can sample the first one again. Thus, the sample rate reduces as the number of markers increases [2]. AM uses three cameras mounted in a rigid housing to track the light emitted by the markers. These systems have accuracy and noise similar to the PM and their main problems are occlusion of markers and markers passing out of the capture volume; even so, AM systems provide easy setup and calibration, excellent spatial resolution (as low as 0.1 mm) and the ability to place markers close together make their use more suitable in certain situations.

There are many commercially available optical systems and J. Richards [31] conducted a study to compare some of them. PM systems reviewed in this study included the Ariel system, BTS's ElitePlus system, Motion Analysis' HiRes system, Peak Performance's Motus system, Qualisys' ProReflex system and Vicon's 360 system. Charnwood Dynamic's CODA system was the sole AM system that was tested. The results of this study revealed that Peak Motus, Vicon and Motion Analysis systems were the fastest PM systems providing the 3D data. However, for the AM system 3D data was immediately available following data collection. Furthermore, all passive optical systems confused marker identifications when markers moved within 2 mm of each other in a 3 mm long volume.

#### 2.2.3.3 Depth Cameras

Depth cameras go by many names such as ranging camera, flash lidar, time-of-flight camera and RGB-D camera. These cameras have been developed for several years and the PMDTec, Mesa Imaging, 3DV Systems and Canesta were the companies driving their development [32]. The operation principle of these cameras can be pulsed light or continuous wave amplitude modulation. Both illuminate the scene with IR light and measure the time-of-flight. The pulsed light measures very short time intervals in order to achieve a resolution which corresponds to a few centimetres in depth, and the other principle

avoids this by measuring the phase shift between emitted and received modulated light which directly corresponds to the time-of-flight. The main limitations of these cameras are low resolution, low sensitivity resulting in high noise levels and background light. Until 2010, year that Microsoft and PrimeSense released the Kinect$^{TM}$ sensor, the laser scanners and structured light approaches were not able to provide high frame rates for full images with a reasonable resolution.

Structured-light 3D scanning is a method that involves highly accurate and expensive scanners and it is based on the projection of a narrow band of light onto a 3D shaped surface producing a line of illumination. Using the deformation of the band when seen from a point of view different from the source to measure the distance from each point to the camera and thus reconstitute the 3D volume. A faster method, with a high number of samples simultaneously, is achieved with a projection of patterns consisting of many stripes of light at the same time. However, the Kinect$^{TM}$ system is a little different and the measurement of depth is described by the inventors as a triangulation process. Instead of projecting stripes of visible light, the IR projector emits a single beam which is split into multiple beams by a diffraction grating to create a constant pattern of speckles which bounces on the objects and is captured by the IR camera [33][3].

To convert the light coding image to a depth map, a chip (PrimeSense's PS1080 chip [33]) is necessary. This chip compares the image received with a reference image stored in the chip's memory as the result of a calibration routine performed on each device during the production process [33]. So, each Kinect$^{TM}$ is calibrated to know exactly where each dot from its projector appears when projected against a flat wall at a known distance. When a dot is projected on an object whose distance to the sensor is smaller or larger than that of the Kinect$^{TM}$'s calibration, the position of the dot in the IR image will be shifted in the direction of the baseline between the IR projector and the perspective centre of the IR camera. These shifts are measured for all dots by a simple image correlation procedure, which yields a disparity image. In every part of the image that the Kinect$^{TM}$ captures from the IR camera, each dot will be a little out of position from where the Kinect$^{TM}$ was expecting to see it [34][3].

Contrary to other depth cameras, the Microsoft Kinect$^{TM}$ has a very low price, making it a very desirable system. Moreover, the Kinect$^{TM}$ has a good working range supported by a large depth range (16-bits depth pixel), a reasonable resolution and a maximum frame rate of 30 Hz. Most importantly, it provides reliable depth measurements

under a large variety of conditions [35]. The technical specifications provided by the Microsoft Kinect$^{TM}$ sensor are provided in Table 2.1 and 2.2 and in Figure 2.4 the Kinect$^{TM}$ depth range is presented. Only in the Kinect for Windows sensor the near range represented in Figure 2.4 is available [13].

In comparison with other systems mentioned in the previous sections, the Microsoft Kinect$^{TM}$ has some benefits like portability, low cost and not requiring markers to determine anatomical landmarks. For these reasons, this system gained the attention of several researches.

| **Field Of View** | |
|---|---|
| Horizontal field of view | 57 degrees |
| Vertical field of view | 43 degrees |
| Physical tilt range | $\pm$ 27 degrees |
| Depth sensor range | 0.8 m - 4 m |

Table 2.1: The Kinect$^{TM}$ technical specifications of Field of View.

| **Data Streams** | | |
|---|---|---|
| Image Stream | Resolution | Rate Frame |
| Depth | 80x60 | 30 frames/sec |
| | 320x240 | 30 frames/sec |
| | 640x480 | 30 frames/sec |
| Color | 640x480 | 30 frames/sec |
| | 1280x960 | 12 frames/sec |

Table 2.2: The Kinect$^{TM}$ technical specifications of Data Streams.

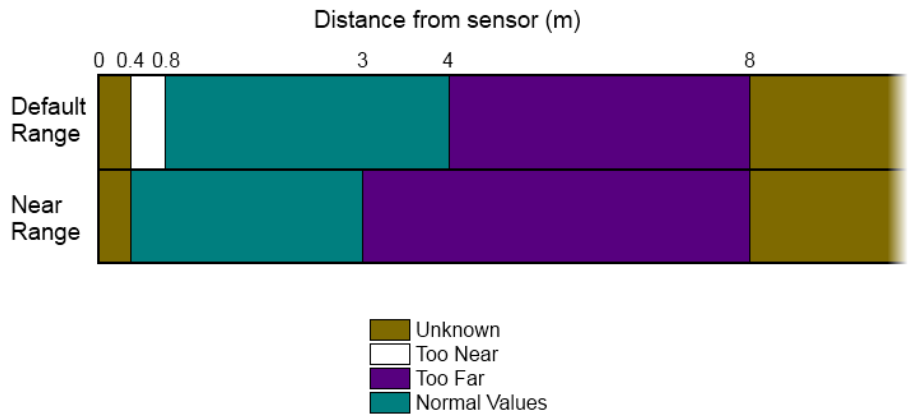



Figure 2.4: The Kinect$^{TM}$ depth range.

The Kinect™ consists of a RGB camera, an IR camera and an IR projector. The RGB camera collects the light that bounces off of the objects in front of them and turns this light into a Two-Dimensional (2D) image that resembles what we see with our own eyes. The IR projector shines a grid of IR dots over everything in front of it and it is possible to capture a picture of these dots using an IR camera. An IR camera works in the same way as the RGB camera but instead of using visible light, it uses the IR light to create an image (a depth image) that captures not what objects look like, but where they are in space. Therefore, the camera records the distance of the objects that are placed in front of it.

The Kinect™ still has four microphones that capture the sound as well as being able to locate the sound within the room, but they will not be used in this work. Inside the Kinect™'s plastic base is a small motor and a series of gears. By turning this motor, the Kinect™ can tilt its cameras and speakers up and down. The motor gives the Kinect™ the ability to aim itself at the best point for capturing people [34]. Lastly, a tri-axis accelerometer configured for 2 g range with a 1 degree accuracy upper limit is also present, where g is the acceleration due to gravity. Thus, accelerometer data can help detect when the Kinect™ sensor is in an unusual orientation [13]. Figure 2.5 shows the Kinect™ sensor with the components mentioned before.

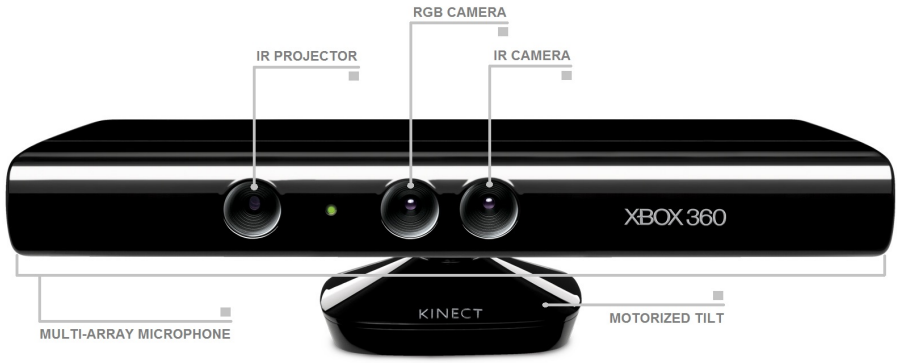



Figure 2.5: The Kinect™ Sensor.

K. Khoshelham et al. [3], in 2012, presented an investigation of the geometric quality of depth data acquired by the Kinect™ sensor and obtained the basic mathematical model for derivation of depth from the observed disparity represented by the following equation (2.1),

$$Z_k = \frac{Z_o}{1 + \frac{Z_o}{fb} d} \tag{2.1}$$

where $Z_k$ denotes the distance (depth) of the point $k$ in object space, $Z_o$ is the distance from the flat wall used in calibration to the sensor, $b$ is the base length, $f$ is the focal length of the IR camera and $d$ is the observed disparity in image space.

The equation 2.1 is obtained by the similarity of triangles represented in Figure 2.6. In this figure the $Z$ axis is orthogonal to the image plane towards the object, the $X$ axis perpendicular to the $Z$ axis in the direction of the baseline $b$ between the IR camera centre and the laser projector, and the $Y$ axis orthogonal to $X$ and $Z$ making a right-handed coordinate system.

The planimetric object coordinates of each point can be calculated from its image coordinates and the scale by the following equations (2.2 and 2.3),

$$X_k = -\frac{Z_k}{f}(x_k - x_o + \delta x) \tag{2.2}$$

$$Y_k = -\frac{Z_k}{f}(y_k - y_o + \delta y) \tag{2.3}$$

where $x_k$ and $y_k$ are the image coordinates of the point, $x_o$ and $y_o$ are the coordinates of the principal point, and $\delta x$ and $\delta y$ are corrections for lens distortion, for which different models with different coefficients exist.

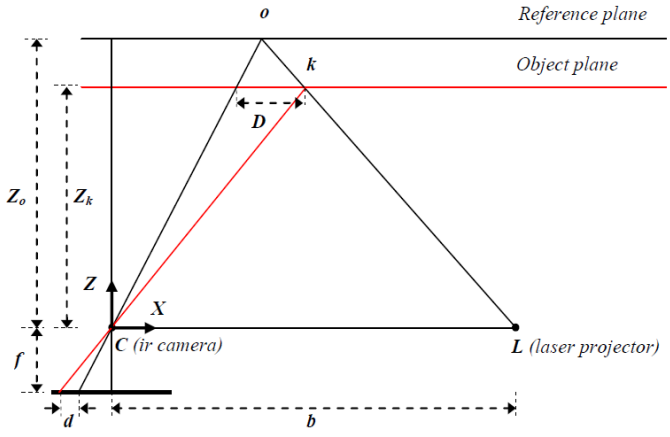



Figure 2.6: Relation between the distance of an object point $k$ to the sensor relative to a reference plane and the measured disparity $d$. If the object is shifted closer to the sensor then the location of the speckle on the image plane will be displaced in the $X$ direction [3].

Depth images are much easier to process than colour images and their possible uses are detecting and tracking individual people, locating their individual joints and body parts. Combining this information with colour images can provide real-time video of

movements performed by adding more detailed information as joints' detection, angles made by them during the movement and dimensions of members.

However, the method used by Kinect™, like any optical method, has some problems with certain objects' surfaces depending on their reflectivity. The dark surfaces absorb light emitted from the IR projector and this light is not reflected back to the IR camera resulting in a gap in the point cloud. Shiny surfaces cause specular reflection and rough surfaces may also be blind to the Kinect™ if the angle of incidence of incoming light is too large. In these cases, 3D reconstruction also has gaps from the respective surfaces.

In order to access the Kinect™ data streams for rehabilitation purposes, drivers are necessary to connect the Kinect™ to a computer. The first open source drivers made were Libfreenect Drivers developed by the OpenKinect community. This community continues to improve and maintain the drivers to this day. It currently supports access to the RGB and depth images, the Kinect™ motor, the accelerometer and the LED [33].

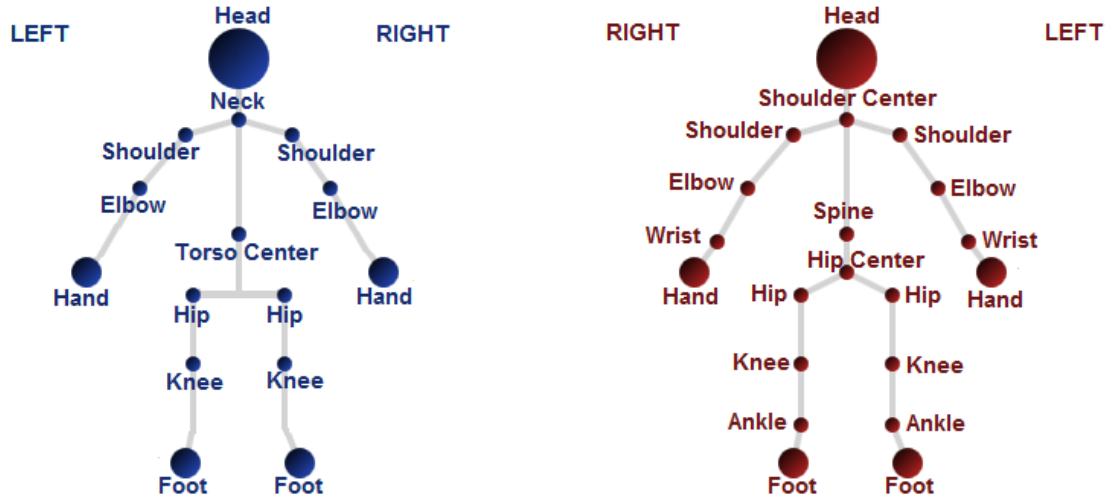



Figure 2.7: Skeleton's front side with NITE Tracked Joints (blue) and Microsoft Tracked Joints (red).

Due to the interest of the use of the Kinect™, PrimeSense which is the company that developed the technology behind Kinect™ 3D imaging, released their software to work with the Kinect™. Their drivers added the possibility to detect users and locate the position of their joints in three dimensions. They called their system OpenNI, for "Natural Interaction". This system is separated into two pieces of software. The first is the OpenNI framework, which is a license similar to the OpenKinect but does not provide

access to motor or the accelerometer because these two components are developed by Microsoft. The optics and the microchip were developed entirely by PrimeSense, so OpenNI framework allows access only for these components. The second module, which is called NITE, is the most attractive part, however NITE is not available under an open source license. PrimeSense provides a royalty-free license that can be used to make projects that use NITE with OpenNI, but it is not currently clear if this license can be used to produce commercial projects [34]. NITE gives access to algorithms for gesture recognition, feature detection and joint tracking.

Six months after PrimeSense released its drivers, Microsoft announced the release of the official Microsoft Kinect™ SDK. This SDK allows the access to all the Kinect™ sensor capabilities plus hand/skeleton tracking. However one of the big limitations for programmer communities is that drivers are only available for Windows, leaving out the Linux and Mac OSX. Moreover, the development of applications is limited to C++, C#, or Visual Basic using Microsoft Visual Studio [33]. Nevertheless, Microsoft Kinect™ SDK offers several important advantages over the OpenNI that are referenced in Table 2.3. The number of tracked joints is also different between these drivers as it can be seen in Figure 2.7.

| | OpenNI | Microsoft SDK |
|---|---|---|
| Raw depth and image data | Yes | Yes |
| Joint position tracking | Yes | Yes |
| API-supported gesture recognition | Yes | No |
| Save raw data stream to disk | Yes | No |
| Joint tracking without calibration | No | Yes |
| Development in C# | No | Yes |
| Audio processing including speech recognition | No | Yes |
| Easy installation | No | Yes |
| Number of joints available | 15 | 20 |
| Quality of documentation | Adequate | Excellent |

Table 2.3: Comparison of toolkits for interfacing with the Kinect™ [6].

## 2.3 Image Processing Techniques

Image processing is the study of the representation and manipulation of pictorial information. An image may be defined as a 2D function, $f(x, y)$, where $x$ and $y$ are spatial

coordinates in a plane, and the amplitude $f$ at any pair of coordinates $(x, y)$ is called the intensity or gray level of the image at that point. When $x$, $y$ and $f$ are all finite, discrete quantities, the image is called digital image and its processing is called digital image processing. A digital image is composed of a finite number of elements, named pixels, and each of those has a particular location and value. Digital image processing is performed on digital computers that manipulate images as arrays or matrices of numbers [5].

Image processing is any kind of method whose input and output are images and image analysis is a method whose inputs may be images, but whose outputs are attributes extracted from those images.

The methods used in this research were chosen in order to minimise the processing time during the acquisition, since the final objective is to develop a real-time application. The sequence of steps to apply in image processing techniques depend on the final objective, without specific rules. In Table 2.4 a sequence of techniques utilized in this thesis is presented.

Image segmentation is an essential concept related with image processing. If it is not done correctly, the rest of the image processing analysis can be affected. In this step, a grey-level image is converted in a bilevel image. A bilevel image is a monochrome image only composed by black and white pixels [36]. Therefore, irrelevant information for the image analysis is excluded, assigning a value of 1 to the useful information and a value of 0 to the background. With this operation the amount of data is significantly reduced, making the processing easier. In this work, a background subtraction is used for this purpose. This method is widely used for detecting moving objects in videos from static cameras. The differences between the current frame and the reference frame are computed to isolate the objects of interest. Thus the moving objects are classified as white (1) and the background as black (0). Other methods, like thresholding, exist and are able to do the segmentation of the image. The most essential thresholding operation will be the selection of a *single threshold value*. All the grey levels below this value are classified as black, and those above it white. However, most of the time it is impossible to segment an image into objects and background with a single value because of noise and illumination effects. Other thresholding methods like *Mean Value*, *P-Tile*, *Edge Pixel*, *Iterative* and *Fuzzy* can be used to solve the problem of noise and illumination effects [36].

| Step | Technique | Description |
|---|---|---|
| 1 | Segmentation | Conversion between a grey-level image and a bilevel image |
| 2 | Morphological Operations | Improve the image |
| 3 | Morphological Algorithm | Object detection |
| 4 | Attributes Extraction | Proprieties extraction of the detected object (Area) |
| 5 | Classification | Classification of the detected object |

Table 2.4: Image processing techniques utilized.

With a binary image a group of mathematical operations can be applied to the set of pixels to enhance or highlight specific aspects of the shape. This process is often called Morphological Image Processing. The morphological operations used in this research were dilation and erosion. These methods are the two primitive operations [5]. One of the simplest applications of dilations is for bridging gaps and of erosion is for eliminating irrelevant detail from a binary image. Thus, dilation expands an image and erosion shrinks it. There are two other important morphological operations that are the combination of the previous operations: opening and closing. Opening is the combination of an erosion followed by a dilation, referring to the ability of this combination to open up gaps between just-touching features. This sequence is one of the most commonly used for removing fine lines and isolated pixel noise from binary images. On the other hand,

closing is a combination of a dilation followed by an erosion [4]. This operation generally fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps in the contour [5]. In Figure 2.8 a result of the application of the two primitive operations in a different order is represented.

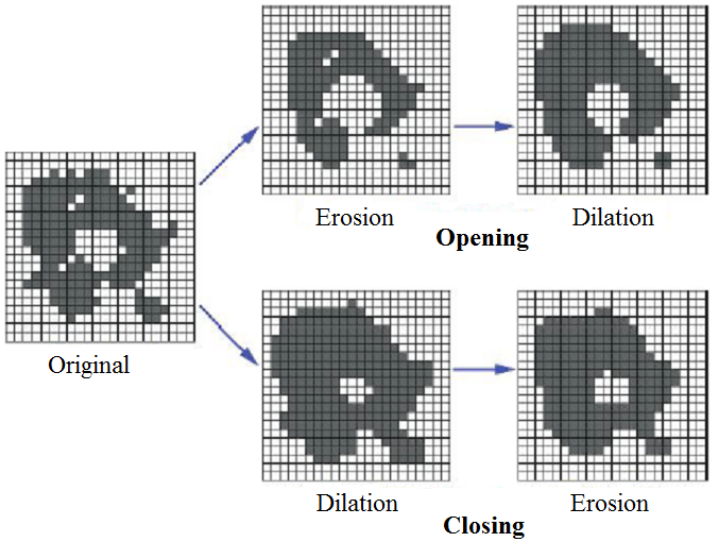
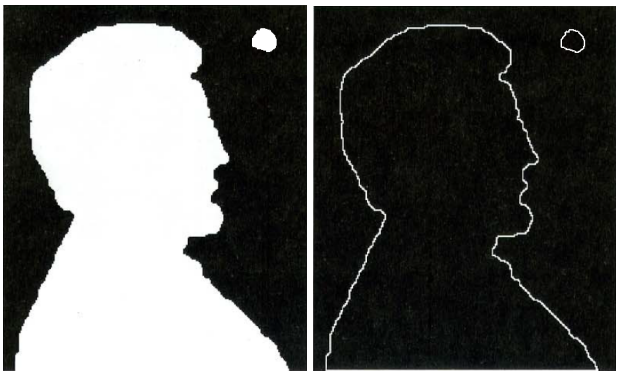



Figure 2.8: Combining erosion and dilation to produce an opening or a closing. The result is different depending on the order of application of the two operations [4].

The main application of morphology is extracting image components that are useful in the representation and description of shape. For this purpose, morphological algorithms were developed for extracting boundaries, connected components, the convex hull and the skeleton of a region. As in this work the objective was isolating only one component in the image, an algorithm for boundary extraction was used. Thus, the boundaries in an image can be obtained by first eroding the original image and then performing the set difference between the original image and its erosion. This algorithm retrieves all contours of the binary image. In Figure 2.9 is shown a result of the application of this algorithm.

After applying the algorithm for boundary extraction, the features extraction is made. In this work, only one characteristic is necessary to identify the object of interest: the area. Thus, the contour with the largest area in the image is chosen and the binary image is composed by 1's inside of the contour and 0's outside of it. If this decision was applied to the image represented in Figure 2.9, only the subject would be used in further processing.

Finally, after the image processing is complete, the image projections of the binary image are computed. These projections are important to detect elements of interest in the



(a) (b)

Figure 2.9: Morphological algorithm: (a) original test image; (b) Result of using a algorithm for boundary extraction. Adapted from [5].

image. The vertical projection is the sum of the pixels in each column of an image. On the other hand, the horizontal projection is the sum of the pixels in each row of an image. In Figure 2.10 a binary image with the respective vertical and horizontal projections is shown.

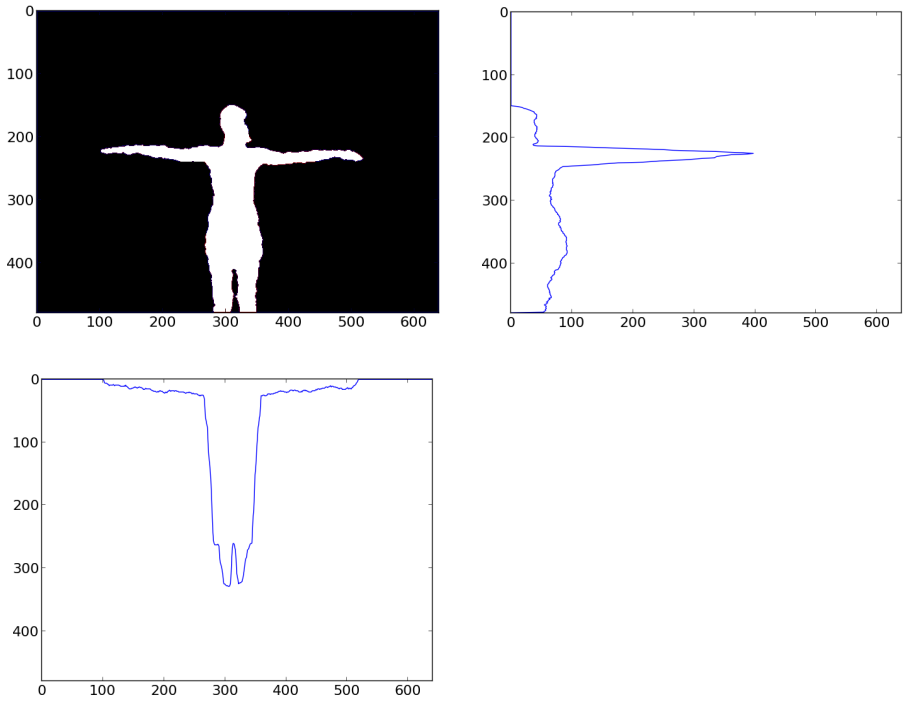



Figure 2.10: Representation of the projections of the binary image. In the right side and in the bottom of the image are represented the horizontal and the vertical projections, respectively.

# 3

# Acquisition

This chapter exposes the data acquisition procedure to validate the Kinect$^{TM}$ Skeleton Tracking as well as the proposed algorithm, using as reference a triaxial accelerometer. The data correspondence between both systems is also presented.

## 3.1 Material and Equipment

In this research, we have compared data from the Kinect against triaxial accelerometer data. Therefore, in the following points, there is a description of both used systems.

### 3.1.1 Triaxial Accelerometer

To validate the Kinect$^{TM}$ Skeleton Tracking and the proposed algorithm, a reference system was needed. The sensor chosen for this purpose was a triaxial accelerometer. This sensor was used to measure arm movements, like abduction and flexion, and it was validated in a study developed by Eva Bernmark et al. [37] with an optoelectronic measuring system. Their results showed that the correspondence between the two systems was almost perfect when the movements were done without influence of dynamic acceleration. The triaxial accelerometer values were on average 1 degree lower than the optoelectronic values.

In this research, the device presented in Figure 3.1, the MotionPlux [38], was used as

an inclination sensor, to measure the inclination of a body segment in relation to the vertical line (the line of gravity). With this device it is possible to define 3D vectors in space and measure the angle between them with constant sensibility over 360 degrees [39]. The MotionPlux is a device that collects and digitalizes signals from a triaxial accelerometer, transmitting them via bluetooth to the computer where the signals are shown in real-time. This system is portable, small sized and has a sampling frequency of 800 Hz.


Figure 3.1: The MotionPlux system.

### 3.1.2 Depth Camera

As mentioned before, the depth camera used in this research was the Kinect™ sensor and its description is presented in section 2.2.3.3. However, for the next sections it is important to know relevant informations related to the recorded frames. The depth information is encoded in an IFTImage as 16 bits per pixel encoded as Little Endian [13]. Thus, each pixel is represented by 2 bytes but only 12 bits are used to represent the distance from camera (in millimetres) to object perceived at pixel. The data format for these 16 bits is represented in Figure 3.2. The most significant bit (Bit 15) is unused in depth

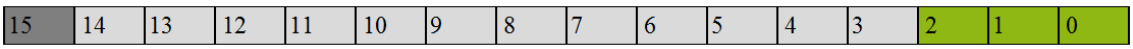



Figure 3.2: Depth data format from the Kinect™.

calculations, the bits from 14 to 3 contain the depth data and the last three bits (Bits 2:0) contain the PlayerID that owns the depth data. Although the depth data uses 12 bits, in this thesis only 8 bits were used for the calculations. Therefore, the last and less significant 4 bits were discarded. These bits represent only 15 mm, therefore, each measured value can have a maximum error of 15 mm. This decision was made because the image manipulation is easier with images with 8 bits per pixel and due to the noise level of the Kinect, the bits discarded do not represent useful information for the calculations (empirically tested).

The coordinate system for all image frames used in this thesis is shown in Figure 3.3. Its origin is in the Kinect$^{TM}$ plane with the $Z$ axis orthogonal to the image plane. In the image plane the $X$ axis is the horizontal and the $Y$ axis is the vertical. Figure 3.3 also represents the Kinect$^{TM}$ field of view, $57\,^{\circ}$ horizontal and $43\,^{\circ}$ vertical, as well as the depth space range in the default range where the distances from 800 millimetre to 4000 millimetres are considered normal values.

The objective of this research is to compute ROM measurements in glenohumeral movements, so the angle was calculated between the arm and a defined reference. This reference can be the world or the subject's thorax. As the Kinect$^{TM}$ has a motor with a tilt range of $-27\,^{\circ}$ to $27\,^{\circ}$, if the user chooses to use the world as reference, the motor angle needs to be considered. In Figure 3.3 the Kinect$^{TM}$ motor angle is $0\,^{\circ}$, so the Kinect$^{TM}$ plane is parallel to the flat wall and in this case the world reference was defined as a vertical straight line that goes through to the joint of interest. However, in Figure 3.4 the Kinect$^{TM}$ plane is not parallel to the flat wall, therefore the distances between them are greater in the top of the image than in the bottom of the image. Thus, in this case the reference used was a straight line that goes through to the joint of interest with the same inclination of the Kinect$^{TM}$ motor.

Unlike in depth space, skeleton space coordinates are expressed in metres. The X, Y and Z axes are the body axes of the depth sensor as shown in Figure 3.5. The skeleton space coordinate system places the Kinect$^{TM}$ at the origin with the positive z-axis extending in the direction in which the Kinect is pointed. The positive Y axis extends upward, and the positive X axis extends to the left [13].

The Kinect$^{TM}$ sensor was used to record the joints' spatial coordinates provided by the official Microsoft Kinect$^{TM}$ SDK and to record the depth frames.

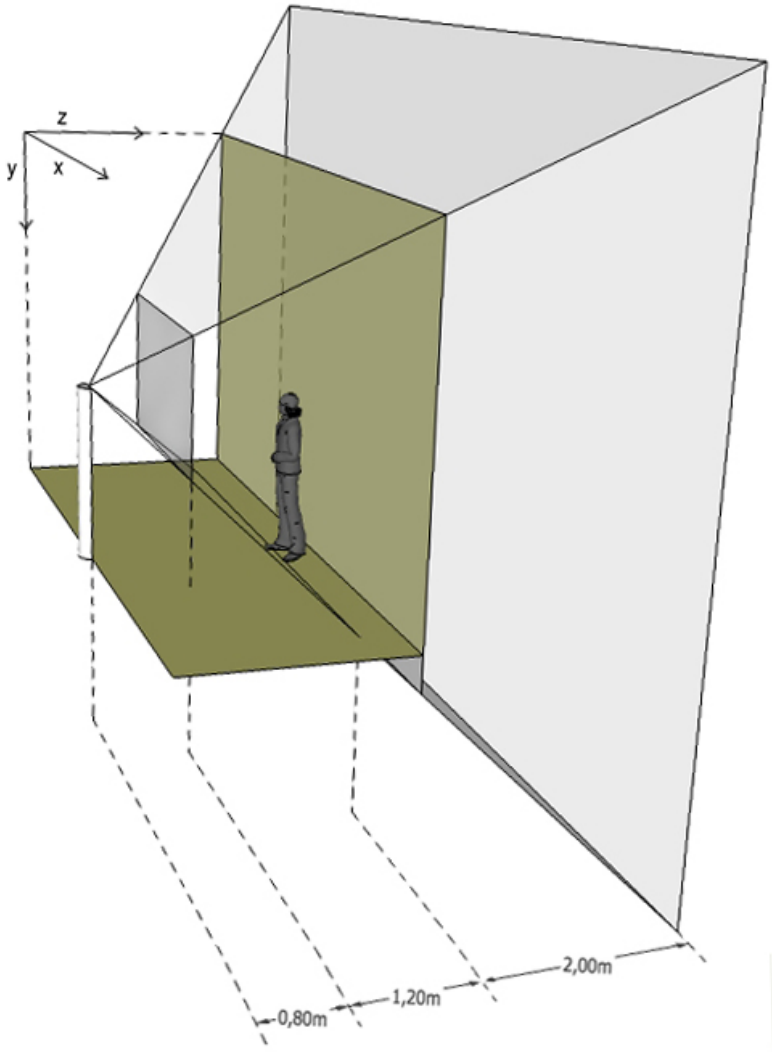



Figure 3.3: Representation of the coordinate system used, the Kinect$^{TM}$ field of view and the depth sensor range.

## 3.2 Procedures

In the tests executed to validate the algorithms, the Kinect$^{TM}$ sensor was positioned in front of a flat wall with its plane parallel to the flat wall and the subject at an approximate distance of 2 meters from it. Although the flat wall is the best environment to do the acquisitions, one may also use other kinds of environments as long as their surfaces are not dark, shiny and rough. At it is known, dark, shiny and rough surfaces may affect the 3D reconstruction creating undesirable gaps in the depth map. The MotionPlux was placed in the subject's arm, aligned with the humerus and with the positive $Y$ axis pointing the roof.

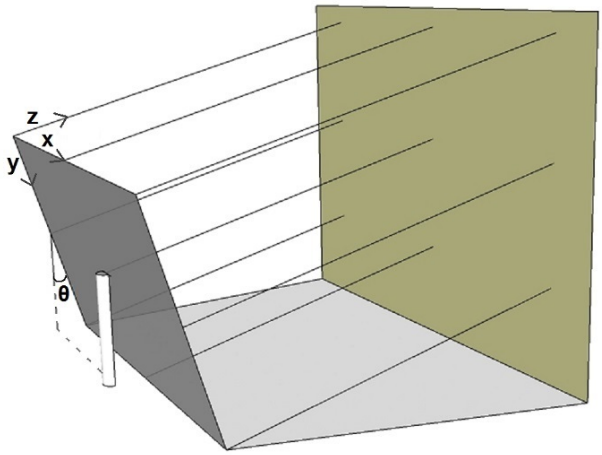



Figure 3.4: Representation of the Kinect$^{TM}$ plane with $\theta$ degrees in the angle of its motor.

The movements studied in the upper body (Figure 3.6) belong to common physical therapies and they were abduction, flexion, extension and internal and external rotation with the arm at 90 degrees of elevation. In abduction movement, the subject needs to stand face to face with the Kinect$^{TM}$ and, in the other movements, the subject needs to be aside to the Kinect$^{TM}$ with the arm that will make the movement closer to the Kinect$^{TM}$.

In the tests performed, the angles of the movement executed were measured with the MotionPlux. This system recorded the raw data and applied it a 1 Hz (empirically tested) low pass Butterworth filter to reduce the influence of the dynamic acceleration. After this procedure the initial vector of the acceleration is saved and used as reference to measure the angles between this vector and the next vectors acquired. Thus the result of this operation is a variation of the movement executed. For this reason, and because the high rates influence the measures from the triaxial accelerometer [37], it was required

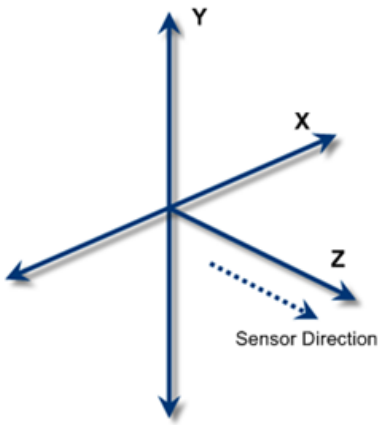
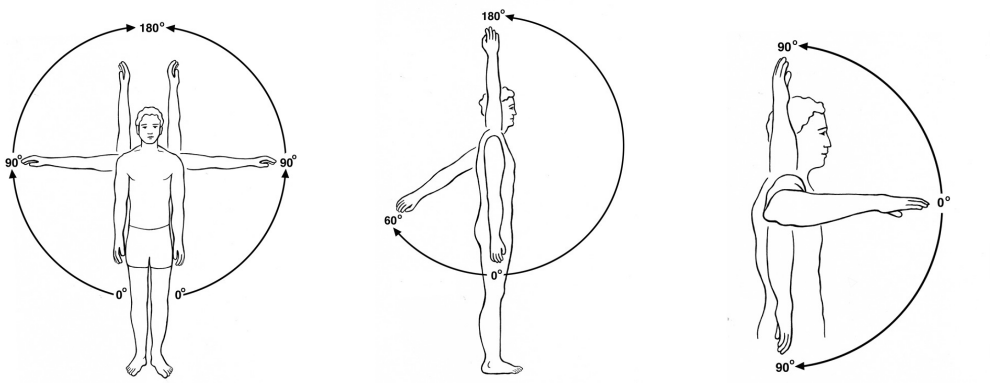



Figure 3.5: Skeleton space coordinate system of the Kinect$^{TM}$.




(a) Abduction (b) Flexion/Extension (c) Internal/External Rotation

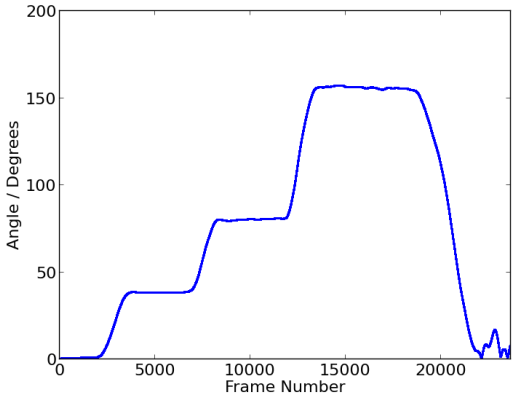
Figure 3.6: Representation of the upper movements studied.

that the subjects stayed 3 seconds in the initial and final positions. In all movements the subjects did three repetitions: the first with the arm at less than 90 degrees, the second with the arm at approximately 90 degrees and the last with the maximum that they could achieve.

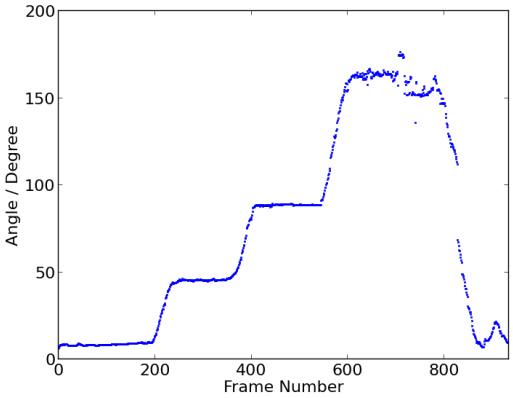
For the validation, this work required the participation of ten healthy subjects, six females and four males. The mean age was 31 years (ranging from 21 to 55 years) and mean height was 170 cm (ranging from 159 to 180 cm). During the acquisitions, the males wore shirts and the females fitting clothes. The lower body clothes were not relevant, but clothes with shiny surfaces, like belts, were avoided.

## 3.3 Data Correspondence

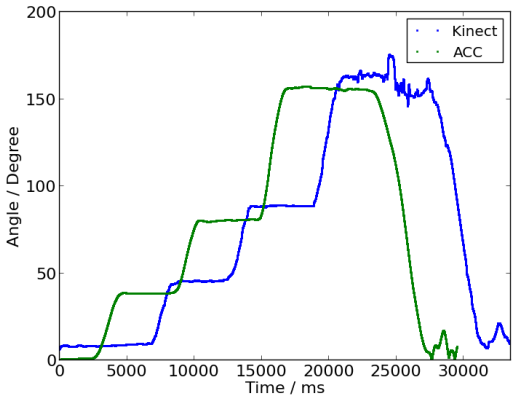
As mentioned before, the sampling frequency of the MotionPlux is 800 Hz and Kinect$^{TM}$'s is approximately 30 Hz. So, after computing the angles with each system, Figure 3.7(a) and 3.7(b), a linear interpolation was done with the computed data from the Microsoft Kinect$^{TM}$ SDK to obtain the same (interpolated) sampling frequency of the MotionPlux, Figure 3.7(c). Then, a correlation with both data was done to synchronize them, ensuring that the data would share a synchronous time reference, Figure 3.7(d). Finally, as the depth frames were recorded at the same time that the joints' spatial coordinates provided by the Microsoft Kinect$^{TM}$ SDK, there was no problem to ensure that the moment analysed is the same in both methods. Thus, for each system an average of the angles highlighted in grey in Figure **??** was done and the results were compared.



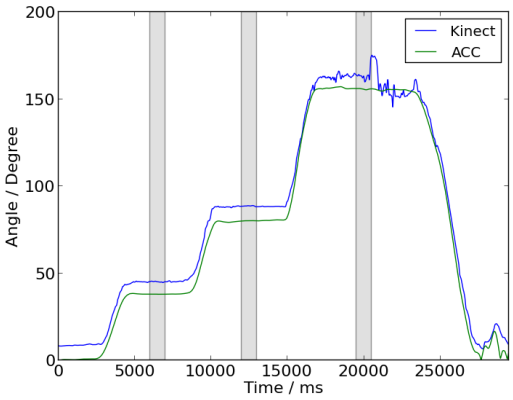

(a) Computed Angles with a triaxial accelerometer




(b) Computed Angles with the joints' spatial coordinates provided by Microsoft




(c) Result of the interpolation done with the values of the subfigure (b) overlaid on subfigure (a)




(d) Result of the correlation done with representation of the instants analysed highlighted in grey

Figure 3.7: Representation of the steps to do a data correspondence between each system analysed.

# 4

# The Kinect Skeleton Tracking

In this chapter, the procedures to analyse and validate the Kinect™ Skeleton Tracking are presented, as well as the main limitations for its use in rehabilitation purposes.

## 4.1 Application

In order to study the Kinect™ Skeleton Tracking for rehabilitation purposes a short application to receive the information from the Kinect™ sensor had to be done. The application was developed in C# using the Visual Studio 2010 and the access of the data streams was made with the official drivers developed by Microsoft, the Microsoft Kinect™ SDK. With this drivers it is possible to use the Microsoft methods to access the anatomical landmarks, however the software is proprietary and closed source. Thus, in the application the joints positions were found and shown in the stream of the RGB image with circles on the subject's joints. Figure 4.1 shows one RGB frame collected from the Kinect™ with all tracked joints available for the upper body.

In the application the user can choose the left and/or right arm to see the angle executed during the movement in real-time. The joints' spatial coordinates and the depth frames can be recorded for posterior analysis.

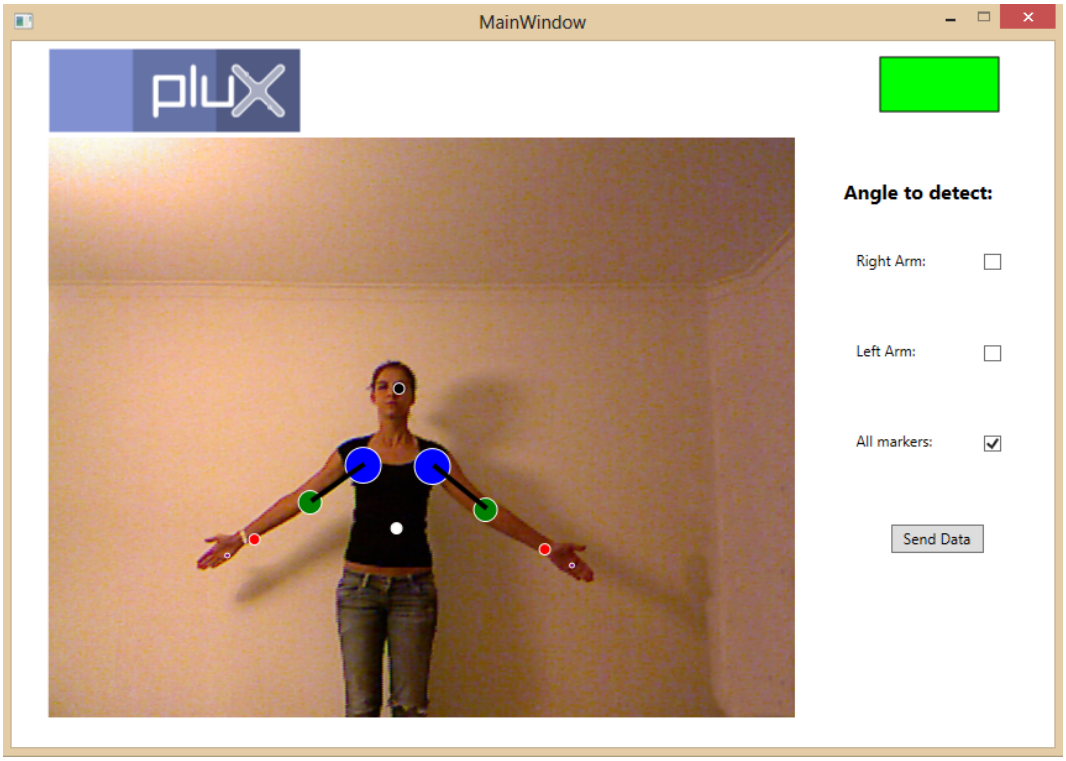



Figure 4.1: Application developed with a RGB frame with all tracked joints available represented with circles.

## 4.2 Data Analysis

With the joints' spatial coordinates it is possible to compute ROM measurements. In this thesis only glenohumeral movements were analysed, so the joints of interest were the shoulder, elbow and wrist. For abduction, flexion and extension movements the shoulder and elbow positions were used and for internal and external rotation with the arm at 90 degrees of elevation instead of using the shoulder position, was used the wrist position. To compute ROM measurements it was necessary to define two vectors and measure the angle between them. If the movement was abduction or flexion/extension, the first vector was composed by the shoulder and elbow spatial coordinates. Otherwise, the vector was composed by the elbow and wrist spatial coordinates. The second vector was the reference and its calculation is explained in section 3.1.2. The equation 4.1 was used to computed the angles performed by the subject.

$$\theta = arccos\left(\frac{A \times B}{\|A\|\|B\|}\right) \tag{4.1}$$

where $\theta$ represents the angle made in degrees, $A$ the first vector mentioned before and $B$ the reference vector.

## 4.3 Validation

To evaluate the performance of the Kinect$^{TM}$ Skeleton Tracking a data comparison with a triaxial accelerometer, the MotionPlux, was done. As previously mentioned the measures were done using variations instead of the maximum angle recorded.

To measure the accuracy the ME, the Standard Deviation (SD) and the Root-Mean-Square (RMS) error were computed. The results obtained between the Kinect$^{TM}$ Skeleton Tracking and the MotionPlux, for all movements, are summarized in Table 4.1.

| **Movement** | **ME / °** | **SD / °** | **RMS / °** |
|---|---|---|---|
| Abduction | 2.91 | 2.51 | 3.34 |
| Flexion | 13.13 | 17.11 | 17.60 |
| Rotation | 18.02 | 19.34 | 19.52 |

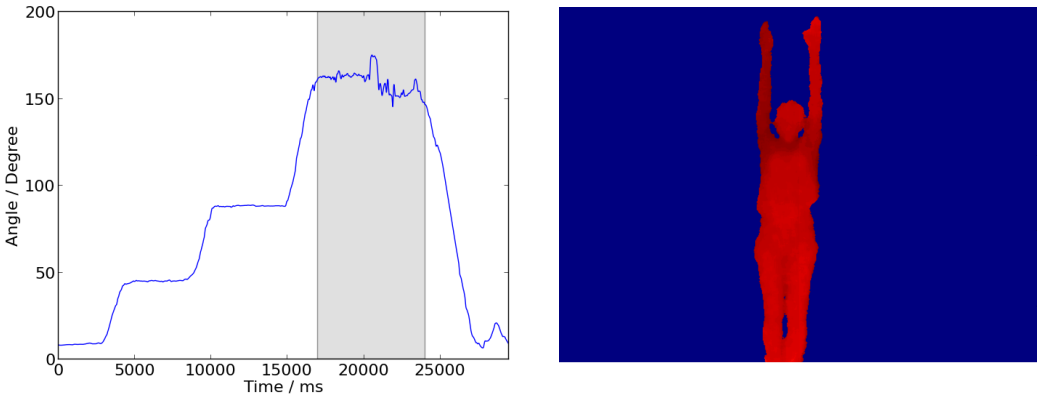
Table 4.1: Comparison between the Kinect$^{TM}$ Skeleton Tracking and the MotionPlux.

Only in abduction the Kinect$^{TM}$ Skeleton Tracking provided results that are considered good enough for rehabilitation purposes with a ME of 2.91 °. The results from flexion/extension and internal/external rotation presented a ME higher than 10 degrees ranging from 1 to 30 degrees. This situation occurs because in these two movements the subject is aside to the Kinect$^{TM}$ and some body parts are occluded, making the tracking more difficult. With the subject face to face with the Kinect$^{TM}$, the difficulties are the same because in certain movement phases the arm is perpendicular to the Kinect$^{TM}$ and the shoulder, elbow and wrist positions are occluded.

In abduction and flexion, the large angles, between 170 and 180 degrees, were always unstable because the shoulders positions were not effectively detected. In the left side of Figure 4.2(b) the angles executed during the movement are represented with the troubling instants highlighted in grey, and in the right side a depth frame recorded in one of these instants after the background subtraction.

Another limitation found was the environment. It was observed that the skeleton tracking struggles with objects in the scene which may be a problem in clinical uses.

Due to limitations mentioned we believe that the Kinect$^{TM}$ Skeleton Tracking has significant potential for rehabilitation purposes but only in a controlled body posture,




(a) (b)

Figure 4.2: (a) Representation of the angles executed during an abduction movement with the troubling instants highlighted in grey; (b) Depth frame acquired in one of these troubling instants.

like abduction/adduction. As the Microsoft Kinect™ SDK is proprietary and closed source, it is not possible to adapt it to our needs nor extend it or improve its abilities.

# 5

# Proposed Algorithm

In this chapter a new algorithm based on the depth map information from the Kinect™ sensor is proposed to overcome the limitations of the Kinect™ Skeleton Tracking. The frames were recorded using the application developed in the previous chapter and the algorithm was developed using the Python 2.7.3. The development of the proposed algorithm can be divided in four main blocks. The first block is composed by the image processing techniques needed to obtain a binary image required for the rest of processing. The second block is used for identifying the subject's movement and it is named automatic calibration. In the third block the detailed explication for identifying anatomical landmarks is presented and in the fourth block the way that the ROM measurements are done is shown. This sequence is systematized on Appendix A highlighting the main events in each block. The ROM measurements validation using the MotionPlux as reference and the algorithm performance are also presented in this chapter.

## 5.1 Image Processing

To be able to use the image frames from the Kinect™ for ROM measurements, image processing techniques must be implemented. The library utilized for this purpose was the OpenCV 2.4.5[1]. The first objective is to isolate the subject from the background, which

[1] http://docs.opencv.org/2.4.5/modules/refman.html

can be done in two different ways, thresholding or background subtraction. The thresholding is simple to apply, however it can be used only in a regular environment, like a flat wall, and in this case it is possible to define a threshold value based on the image histogram. The background subtraction is a more robust technique and it can be applied in any environment. In this approach the differences between the current frame and the reference frame are computed to isolate the subject. However, as the subject's movements are very small, only the arm presents a considerable movement, an image in the same environment without the subject is used as reference frame. After subtracting directly these frames, it is necessary to define a threshold near to the extreme values, 0 and 255 (image with 8 bits per pixel). As it can be seen in Figure 5.1(c) the result of the subtraction is not necessarily 0 in the static scene due to the noise level of the Kinect$^{TM}$ sensor. Thus, all pixels in the image near to the values 0 and 255 are classified as black (0), and the others are classified as white (1).

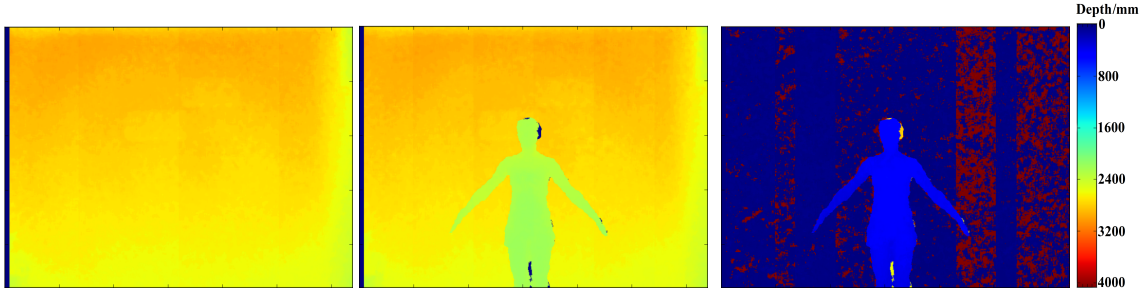
In the next step an opening is applied (combination of an erosion followed by a dilation) for removing fine lines and isolated pixel noise from the binary image. At this stage it is important to extract only a component of the image that is useful: the subject. To do this, a morphological algorithm for boundaries extraction is applied. The method used was $findContours$ [2] which retrieves all contours of the binary image. With this information the contour with the largest area is chosen and the binary image is composed by 1's inside of the contour and 0's outside of it. In Figure 5.1 the sequence of events mentioned before is represented.

## 5.2 Automatic Calibration

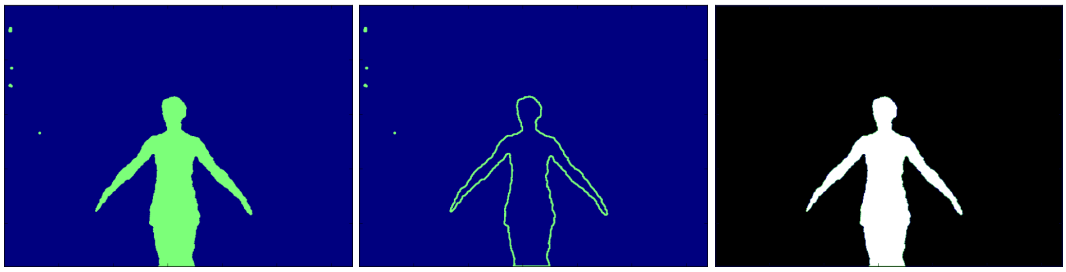
The calibration process is useful for recording some important measures of the subject. These measurements, represented in Figure 5.2, are needed to find the shoulder, elbow and hand joint positions, as well as the head and the thorax inclination during any movement position.

In order to do these measurements, it is necessary to first find the head and the shoulders positions before the subject begins the movement. The head $y$-coordinate is the first non-zero point in the horizontal projection and its $x$-coordinate is the average of the $x$ positions in the horizontal line of the $y$-coordinate, where the pixel value is 1. The

[2] http://docs.opencv.org/2.4.5/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html#findcontours




(a) Reference Frame (b) Current Frame (c) Background Subtraction

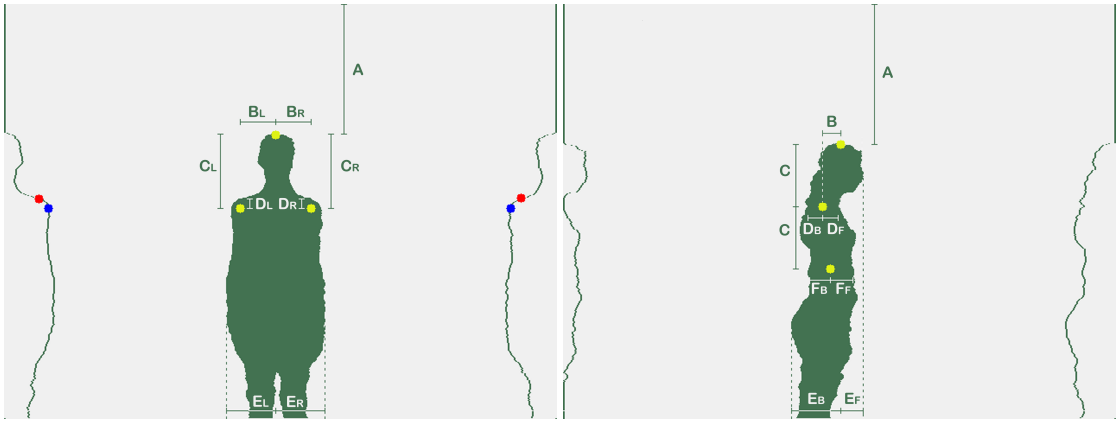


(e) Morphological Operations (f) Morphological Algorithm (g) Final Binary Image

Figure 5.1: Representation of the image processing techniques used.

shoulders detection depends on the subject's position. When the subject is face to face with the Kinect$^{TM}$, the shoulders are detected calculating the maximum difference along the points of the horizontal projection (red point in Figure 5.2(a)) and found, after this point, the first point when the difference is approximately constant (blue point in Figure 5.2(a)). Thus, the shoulder position is the intersection of the vertical straight line that goes through to the red point with the horizontal straight line that goes through to the blue point. In the other position, the shoulder is detected based on the difference of depth between the head and the remnant body. Firstly, the minimum depth value in a line after a reasonably large number of pixels after the head $y$-coordinate is found (adding a constant to the head $y$-coordinate). The value of this constant is not important and serves only to ensure that the region intersected is the arm and not the head. Knowing the $x$ position of this point, the shoulder is found through this vertical line until the depth values decrease less than 31 millimetres (variation of one unit - the 4 bits less significant are unused - see section 3.1.2). The value found represents the transition between the head and the shoulder. Therefore, the shoulder $y$-coordinate is the $y$ position of this value. Its $x$-coordinate is computed finding the average of the $x$ positions with the pixel value equal to the depth value found and in the horizontal line of the shoulder $y$-coordinate.

After detecting the head position, the algorithm is able to detect which movement the




(a) Face to face with the Kinect™ (b) Aside to the Kinect™

Figure 5.2: Representation of the subject measures in two frames from the Kinect™ after processing them. In the left and right side of the images are represented the horizontal projections of the binary image of half of the body.

patient does. First of all, the algorithm identifies if the subject is in the position represented in Figure 5.2(a) or 5.2(b), based on the differences between the left and the right horizontal projections of the binary image of half of the body. These projections are obtained dividing the subject's body by a vertical straight line that goes through the head position and then their differences are computed with the variance of the quotient between both projections. The variance is calculated with the following equation (5.1),

$$\sigma^2 = \frac{1}{N}\sum_{i=0}^{N}(x_i - \mu)^2 \quad , \quad x = \frac{P_1}{P_2} \tag{5.1}$$

where $\sigma^2$ is the population variance, $N$ is the number of elements in the population, $x_i$ is the $i$ element from the population, $\mu$ is the population mean, $P_1$ is the horizontal projection of half of the body of the side on which the movement is executed (left or right) and $P_2$ is the other projection.

As it can be seen in Figure 5.2 the general shape of the two projections when the subject is face to face with the Kinect™ is the same. However, when the subject is aside to the Kinect™ the differences are obviously visible. Therefore, the variance obtained when the subject is face to face with the Kinect™ is shorter than the variance obtained in the other position and it is possible to define a threshold to differentiate both positions.

Thus, if the subject is face to face with the Kinect™, the movement is abduction, otherwise it is necessary to choose between flexion/extension and internal/external rotation. At this stage, the measures presented in Figure 5.2 begin to be recorded during

approximately one second (30 frames). The measure chosen for the rest of the processing is the median of all measures. After this extraction, the subject receives the information that can begin the movement. If the subject is in the position represented in Figure 5.2(b), the algorithm starts to recognize which movement the subject is doing (flexion or rotation). The main difference between flexion/extension and internal/external rotation is the proximity of the arm with the body.

Therefore, with the depth information provided by the Kinect$^{TM}$, an average with all depth values in the subject's body is calculated. Then, a depth threshold is established that is a considerable percent of the depth average calculated before. If there is an object with a considerable area, near the head position and with the depth values smaller than the threshold, the movement is rotation. In flexion almost the entire body has depth values greater than the threshold.

In the position represented in Figure 5.2(b), the detection of the thorax inclination is also needed. It is assumed that in calibration the subject is not compensating with the body. Thus, the inclination measure is the reference of that subject. To find the centre of the subject, the average of the $x$ positions in the horizontal line calculated with the $C$ and $D$ measures, where the pixel value is 1, is computed. For the internal/external rotation this process is repeated for more 4 lines, 2 above and 2 below with a distance of 10 pixels between them, and the respective measures are recorded.

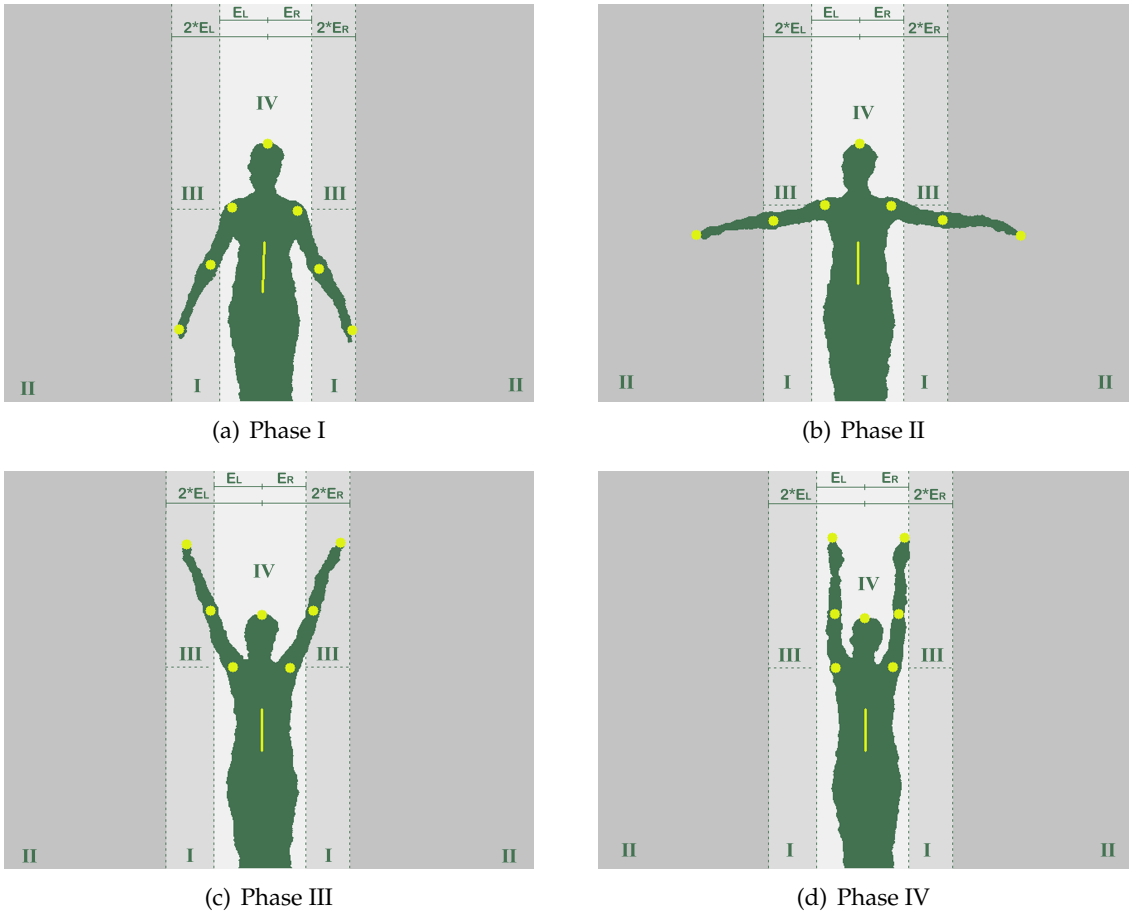
## 5.3 Anatomical Landmarks

To compute the angles done during the movements, few anatomical landmarks need to be detected. In the following sections, the way that they are detected is described. As their detection depends on the movement performed they are described separately for each movement. To simplify this description it is assumed that the subject is always doing the movement with the left arm. For the right arm the determination processes are analogous.

### 5.3.1 Abduction

In abduction the processing to find the anatomical landmarks is always done with the information of the projections of the binary image. In this movement the depth information is only used to compute the 3D coordinates. During processing it is necessary to

recognize the movement phase, because the way that the algorithm finds the anatomical landmarks varies with it. Thus, the differences between the first and the last non-zero values of the vertical projection in each frame are compared with the $E_L$ and $E_R$ measures from the calibration, Figure 5.2(a), identifying if the subject is in phase $II$, Figure 5.3(b), or not. If the differences computed indicate that the subject's arms are not in phase $II$, it is necessary to check if the arms are above or below the head with the information of the $A$ measure from the calibration. In Figure 5.3 is represented the main movement phases and in the following points are described the detection of each element for abduction.




(a) Phase I

(b) Phase II

(c) Phase III

(d) Phase IV

Figure 5.3: Representation of the main movement phases of abduction.

- The **thorax inclination** is the first detected element. Its first point is computed in a horizontal line obtained with a sum of the $A$ measure, the result of the average of the $C_L$ and $C_R$ measures and a constant. If the movement phase is $II$ (with the arms doing more than 90 degrees), $III$ or $IV$ it is only necessary to do an average of the $x$ positions in the subject's body, where the pixel value is 1, to find the subject's centre.

This procedure is repeated for 6 horizontal lines with a distance between them of 10 pixels. After this, a linear regression with all points is done. The other movement phases needed to have more precautions, because the arms cross the vertical lines where the thorax inclination is computed and their influence in its calculation is not desired. Thus, to compute the average of the $x$ positions in each horizontal line, a difference of the consecutive points in this line is done. As the image is binary the result of this operation is 0's in all positions except in the boundary of the subject that is 1 or -1. With this information the body position is known and it is possible to find its centre. In phase $I$, if the subject's arms are united with the body, the thorax inclination is not calculated because the arms influence can not be removed.

- The **shoulders** are the next detected element and their positions are calculated with the information of the thorax inclination combined with the distances from the calibration. Thus, the $B$ and $D$ measures from the calibration are predetermined by the thorax inclination. The straight lines parallel to the thorax represented in Figure 5.4, are computed with the $B_R$ and $B_L$ measures from the calibration. In each straight line the first non-zero point is found and represents the beginning of the subject's body. After that the $D_R$ and $D_L$ measures from the calibration are used in the same direction of the straight line, to find the shoulder position.

- The **head** is detected after the shoulders and if the movement phase is $I$ or $II$ (with the arms below the head), the first non-zero point in the horizontal projection is its $y$-coordinate. Otherwise, as the height of the subject in the image is known ($A$ measure from the calibration), the first non-zero point in the horizontal projection is shorter than the value of the head in calibration. Therefore, it is assumed that the head $y$-coordinate is the same of the calibration. For the $x$-coordinate, the average of the $x$ positions, where the pixel value is 1, in the horizontal line between the shoulders $x$-coordinates is calculated.

- The left **hand** is found from the extremes of the projections. The use of the vertical or horizontal projection depends on the movement phase. In phase $III$ and $IV$ the hand $y$-coordinate is the first non-zero point in the left horizontal projection of the half of the body. Its $x$-coordinate is the average of the $x$ positions until the head $x$-coordinate, where the pixel value is 1. In the other movement phases the hand $x$-coordinate is the first point in the vertical projection, and its $y$-coordinate is the

average of the $y$ positions in the respective hand $x$-coordinate.

- The left **elbow** is computed assuming that the humerus length corresponds to 40 percent of the distance between the shoulder and the extremity of the hand [40]. Thus, the distance between the two positions is calculated and 60 percent of that is added to the value of the left hand $x$-coordinate. The $y$-coordinate is calculated with the average of the $y$ positions in the line of the elbow $x$-coordinate. When the subject achieves approximately 180 degrees or more, phase $IV$, the elbow $y$-coordinate is the value of the head $y$-coordinate [40], and the $x$-coordinate is the average of the $x$ positions until the head $x$-coordinate.

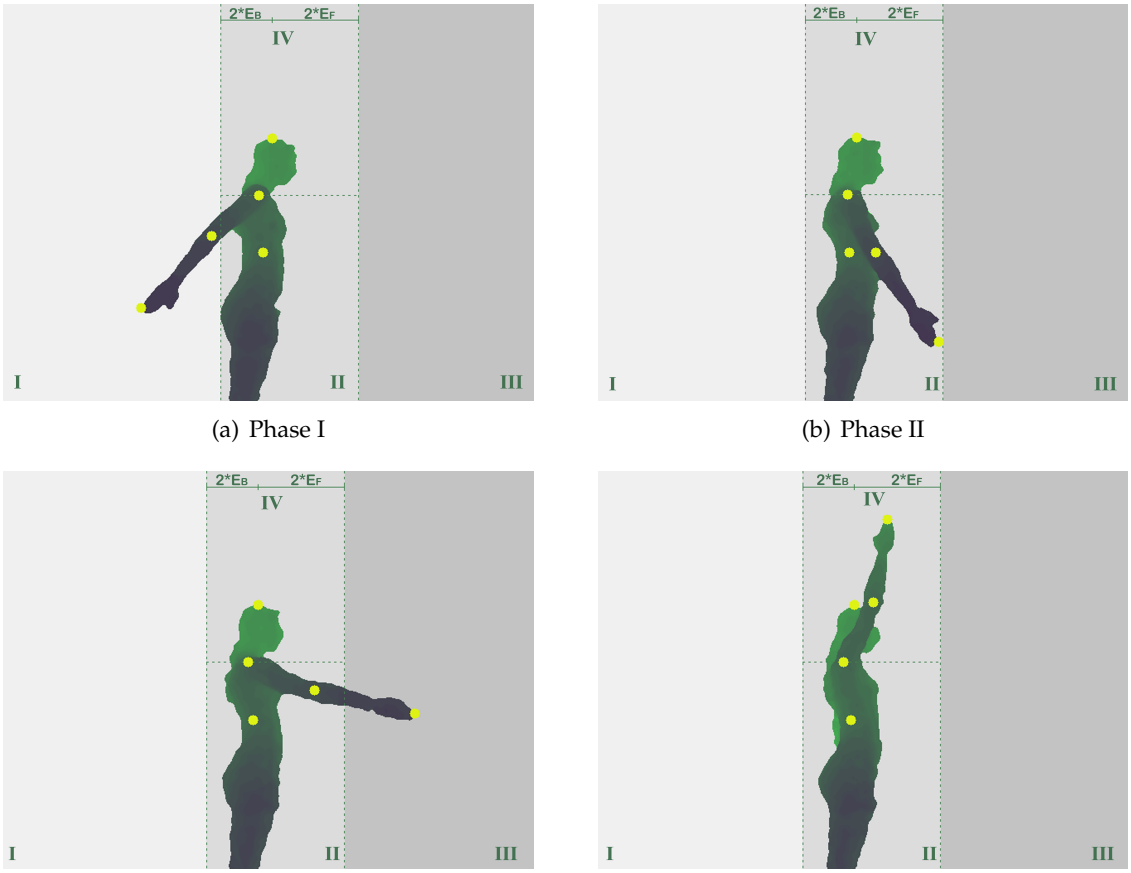

Figure 5.4: Representation of the shoulder detection based on the thorax inclination.

### 5.3.2 Flexion/Extension

In flexion/extension the anatomical landmarks are found with the depth information combined with the projections of the binary image. In this movement the way that movement phases are detected is equal to the previous movement and their representation is presented in Figure 5.5.

- The **head** detection is equal to the previous movement, however in phase $IV$, Figure 5.5(d), the $B$ measure from the calibration needs to be used, because the subject's arm influences the result of the average of the $x$ positions with a pixel value of 1.

- The **shoulder** is the next detected element and its $y$-coordinate is calculated adding the $C$ measure from the calibration to the head $y$-coordinate. In this horizontal line




(a) Phase I

(b) Phase II

(c) Phase III

(d) Phase IV

Figure 5.5: Representation of the main phases of flexion/extension.

the algorithm finds the first non-zero point and sums the $D_B$ measure. Only in movement phase $I$, extension, the $D_F$ measure from the calibration is used, and instead of using the first non-zero point the algorithm uses the last non-zero point to subtract it.

- In all movement phases the **hand** is detected differently. In phase $I$ the hand $x$-coordinate is the first non-zero point in the vertical projection and in phase $III$ is the last non-zero point in the same projection. The hand $y$-coordinate is the average of the $y$ positions in its $x$-coordinate, where the pixel value is 1. In phase $IV$ the hand $y$-coordinate is the first non-zero point in the horizontal projection and the hand $x$-coordinate is again the average of the $x$ positions in its $y$-coordinate. In the movement phase $II$, if the hand position coincides with the body position, the hand is not detected, otherwise its detection is equal to the phase $I$ or $III$.

- The **elbow** is computed following the same principle mentioned in section 5.3.1.

Thus, in phases $I$, $III$ and $IV$ the procedure is equal. However, in phase $II$ the algorithm needs to use the depth map information to recognize the body part closer to the Kinect$^{TM}$. If the hand was detected, the algorithm finds the minimum depth value in the horizontal line correspondent to the 40 percent of the distance between the shoulder and the hand. After this, an average of the $y$ positions with the minimum depth value is computed. Otherwise, the elbow $y$-coordinate is computed adding a constant to the shoulder $y$-coordinate. In this horizontal line an average of the $x$ positions with the minimum depth value is computed to find its $x$-coordinate. This constant, after the first movement, can be replaced with the 40 percent of distance between the shoulder and the hand, calculated in other movement phase.

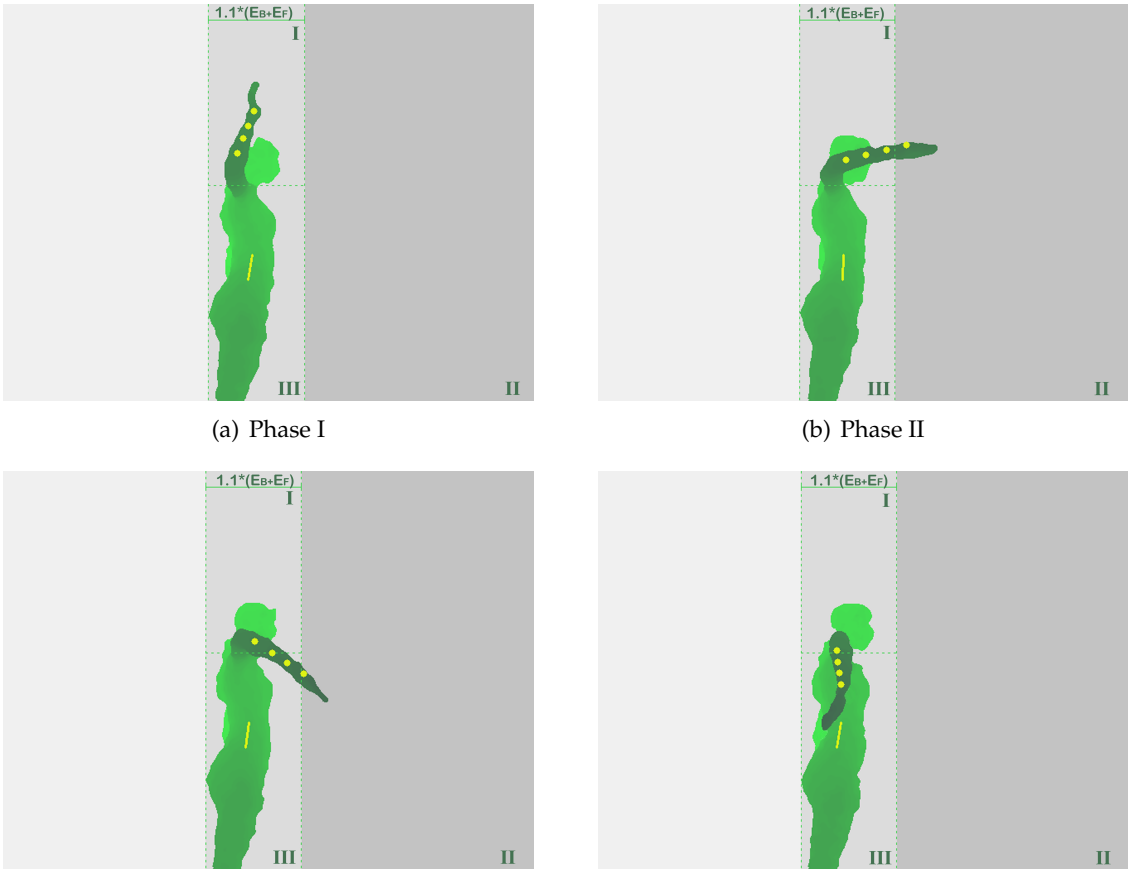
- To calculate the **thorax inclination** only one point in the centre of the subject's body is computed. In movement phases $II$, $III$ and $IV$ the algorithm finds the first non-zero point and adds the $F_B$ measure from the calibration. To the movement phase $I$ the algorithm finds the last non-zero point and subtracts the $F_F$ measure.

### 5.3.3 Internal/External Rotation

Internal/external rotation with the arm at 90 degrees of elevation needs the information of the depth map and the projections of the binary image. In this movement, the head and the shoulder are not detected because their detection is difficult and they are not required for ROM measurements. Unlike the other movements, the exact position of the anatomical landmarks is not necessary because the ROM measurements are computed in a different way, that is explained in next section (5.4). The movement phases are represented in Figure 5.6 and they are obtained with a percent of the sum of the $E_B$ and $E_F$ measures.

- The approximate **elbow** position is calculated with the sum of the $A$ and $C$ measures from the calibration, Figure 5.2(b), because in this movement the subject's arm is at 90 degrees of abduction. Thus, the $x$ and $y$ coordinates of the shoulder are approximately the same of the elbow.

- Similarly with the other movements, the **hand** $y$-coordinate in the movement phase $I$ is the first non-zero point in the horizontal projection and its $x$-coordinate is the average of the $x$ positions in this line, where the pixel value is 1. In the movement phase $II$ the hand $x$-coordinate is the last non-zero point in the vertical projection




(a) Phase I

(b) Phase II

(c) Phase II

(d) Phase III

Figure 5.6: Representation of the main phases of internal/external rotation.

and its $y$-coordinate is the average of the $y$ positions in this vertical line, where the pixel value is 1. Finally in phase $III$ the exact position of the hand is not calculated, but the algorithm finds the minimum depth value in a horizontal line after the elbow $y$-coordinate, adding a constant. Its $x$-coordinate is the average of the $x$ positions where the depth value is equal to the minimum.

- To compute the **thorax inclination** the algorithm finds the first non-zero points in the horizontal lines calculated with the $C$ measure from the calibration and adds the $F_B$ measures. However, in phase $III$, only if the difference between the hand and the first non-zero point in the vertical projection was shorter than the difference between the hand and the last non-zero point, the $F_F$ measures are used. These measures are subtracted to the positions of the last non-zero points. After this procedure, a linear regression with all points is computed.

## 5.4 ROM Measurements

The way that the ROM measurements are computed depends on the movement performed. In abduction and flexion/extension the procedures are the same, however as the detection of the anatomical landmarks in internal/external rotation is not precise, it was necessary to find another way to compute it.

In abduction and flexion/extension the ROM measurements are computed in 3D coordinates. As the $x$ and $y$ coordinates of the shoulder and elbow are in pixels and the $z$ coordinates are in millimetres, the conversion of the $x$ and $y$ coordinates from pixels to millimetres is necessary. For this purpose the equations 2.2 and 2.3 are used. To compute the angle that the subject is doing during the movement it is necessary to define two vectors and measure the angle between them. The first vector is composed by the spatial coordinates of the shoulder and elbow joints and the other depends on the reference that the user wants to use. If the reference is the world, the calculation of the vector is explained in the section 3.1.2. However if the user wants to consider the body movements, the use of the thorax inclination as reference is very useful because if the subject compensates the movement with the body the ROM measurements are not affected, unlike the use of the world as reference. Thus, the vector in this case is composed by two points of the thorax straight line, if the movement was abduction, or by the shoulder point and the body centre, if the movement was flexion/extension.

For internal/external rotation with the arm at 90 degrees of elevation the ROM measurements are computed in 2D coordinates, because the detection of the rotation is done with the forearm position. If the forearm is not perpendicular to the arm, the rotation is not influenced. Therefore, the $z$ coordinate information would influence the measurement wrongly. Thus, in this movement, the shortest depth values are found in 4 perpendicular straight lines between the vector composed by the hand and the elbow position. With these values a linear regression is done and to compute the ROM measurements a vector can be defined with two points of the linear regression. The other vector needed is defined in the same way that is done in abduction and flexion/extension.

As mentioned before, when the subject is aside to the Kinect$^{\text{TM}}$ the thorax inclination is calculated in calibration to know the reference of each subject. Thus, after the ROM are computed the difference between the thorax inclination in calibration and in each frame is added to the final angle.

## 5.5 Validation

Like the section 4.3 the validation of the proposed algorithm was done with the MotionPlux and the process and the parameters evaluated were the same. Thus, in Table 5.1 are presented the ME, SD and RMS between the proposed algorithm and the MotionPlux.

| Movement | ME / ° | SD / ° | RMS / ° |
|---|---|---|---|
| Abduction | 1.14 | 1.39 | 1.40 |
| Flexion | 1.23 | 1.53 | 1.56 |
| Rotation | 1.23 | 1.47 | 1.54 |

Table 5.1: Comparison between the proposed algorithm and the MotionPlux.

The ME was less than 1.5 degrees for all movements, with no significant difference between them. In Figure 5.7 the error bars for all movements studied and with both methods used are represented. As it can be seen in this figure, flexion/extension and internal/external rotation improved an order of magnitude when compared with the same results from the Kinect™ Skeleton Tracking.

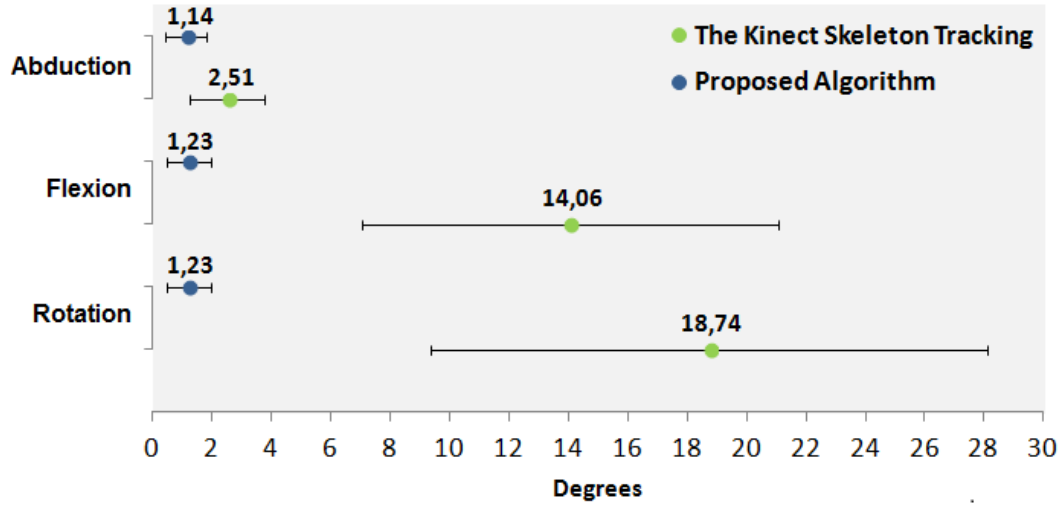



Figure 5.7: Error bars for all movements studied in which the blue points represent the proposed algorithm and the red points the Kinect™ Skeleton Tracking.

The results obtained suggest that the proposed algorithm is good enough for rehabilitation purposes, since in clinics the ROM measurements are done manually with a goniometer and its results present a 5-10 degrees error [9]. Furthermore, the addition of biofeedback using the Kinect™ sensor is another advantage in clinical practice.

## 5.6 Real-Time

The proposed algorithm was done using the Python programming language, however to receive the information from the Kinect$^{TM}$, the official drivers developed by Microsoft were used in Visual Studio 2010. Therefore, the processing is not done in real-time. Although, the Python language has several advantages, like intuitive object orientated and extensive standard libraries, doing a real-time application using it, may cause some performance problems, since that is the most cited limitation of Python. In Tables 5.2 to 5.5 the performance results of each part of the proposed algorithm are shown. The values in the tables were obtained doing an average of the elapsed time with 30 frames, 10 of each movement analysed, and with all movement phases. The tests were done using a Samsung 700Z computer with 8 GB ram, 2.40 GHz quad core i7 processor running the Windows 8 operating system.

As it can be seen, the image processing and the calibration are the slowest parts. In the calibration the user does not receive any information, then there is no objection for not doing real-time processing. However, if the movement was aside to the Kinect$^{TM}$ the number of the frames processed should decrease, because in these movements the processing is too slow. The image processing can bring more problems once every single frame needs to pass this step. As the frame rate of the Kinect$^{TM}$ is 30 Hz, to do an ideal real-time application with it would be necessary that all processing took less than 33 ms. Although the anatomical landmarks and the ROM measurements take at most 10 ms, the image processing done in Python hinders the ideal real-time processing. However, in the clinical environment it is not necessary to process every single frame received from the Kinect, being acceptable for a real-time application to refresh the information with a frame rate of 10 Hz. Therefore, the frames processing could reach approximately 100 ms, which is the time that our algorithm takes to process the frames. Still, the real-time application is not entirely achieved because the usage of the Microsoft drivers does not allow its information to be received in Python. So, the transmission time between C# and Python needs to be added to the aforementioned 100 ms.

In spite of Python language being very useful to find the best way to perform image processing, in this state of the algorithm, it is important to get higher performance in order to achieve a real-time application, i.e, developed a rehabilitation application using biofeedback. As the code to receive the information from the Kinect$^{TM}$ was done in

| Image Processing | Time / ms |
|---|---|
| Image Reading | 12.00 |
| Background Subtraction | 49.99 |
| Morphological Operations | 23.00 |
| Boundary Detection | 5.00 |
| **Cumulative Time** | **89.99** |

Table 5.2: Performance of the image processing.

| Calibration | Time / ms | Cumulative Time / ms |
|---|---|---|
| Abduction | 9.69 | 99.68 |
| Flexion | 959.19 | 1050.18 |
| Rotation | 1045.19 | 1135.18 |

Table 5.3: Performance of the calibration.

| Anatomical Landmarks | Time / ms | Cumulative Time / ms |
|---|---|---|
| Abduction | 9.99 | 99.98 |
| Flexion | 4.00 | 93.99 |
| Rotation | 8.00 | 97.99 |

Table 5.4: Performance of the anatomical landmarks.

| ROM measurements | Time / ms | Cumulative Time / ms |
|---|---|---|
| Abduction | < 0.01 | 99.98 |
| Flexion | < 0.01 | 93.99 |
| Rotation | < 0.01 | 97.99 |

Table 5.5: Performance of the ROM measurements.

C# and this language has a better performance than Python, a survey to realize if it is possible to convert all code developed in Python to C# was done. The main packages

used in Python were the OpenCV, SciPy[3], Matplotlib[4] and NumPy[5]. The OpenCV is a image processing library and despite not having a C# interface, there is a wrapper, named EmguCV[6], that allows to call functions from OpenCV. Thus, in this library case, the functions used were the same. The SciPy was essentially used to do a linear regression using the $linregress$ function from the statistics modulo ($stats$). In C# there are several libraries with statistics modules that can be used for this purpose, such as ALGLIB[7], Meta.Numerics[8] and Extreme Optimization[9]. The Matplotlib package was applied to the image display, however, in C# the image display is not required and it is replaced by the video stream, which is achieved using the $System$ namespace. Finally, the NumPy package was used to do basic mathematical operations. Thus, in C# is possible to use the modulo $Math$ from the $System$ namespace or the libraries mentioned before for the linear regression. Therefore, the conversion of the code is possible for a real-time application.

[3] http://docs.scipy.org/doc/scipy/reference/
[4] http://matplotlib.org/1.3.0/contents.html
[5] http://docs.scipy.org/doc/numpy/reference/index.html
[6] http://www.emgu.com/wiki/files/2.4.2/document/Index.html
[7] http://www.alglib.net/translator/man/manual.csharp.html
[8] http://www.meta-numerics.net/Documentation/index.html
[9] http://www.extremeoptimization.com/Documentation/

# 6

# Conclusions

In this last chapter, an overview of the general contributions that this research provide in the rehabilitation field, are exposed. The main results obtained and the future work are also presented.

## 6.1 General Results

A new algorithm is proposed to do ROM measurements in clinical practice using the Kinect$^{TM}$ sensor. The first intention would be to do an application using the official drivers developed by Microsoft, however the tests performed showed some limitations that could be a problem in clinical applications. Abduction was the movement that obtained better results, with a ME of 2.91 degrees using a triaxial accelerometer as reference. However in this movement and in flexion, the large angles, between 170 and 180 degrees, were always very unstable because the shoulders positions were not effectively detected by the Microsoft drivers. In internal/external rotation case and even in flexion/extension the Microsoft Kinect$^{TM}$ SDK was not always able to track the movement because in these positions some body parts are occluded. Another limitation found was the environment. It was observed that the skeleton tracking struggles with objects in the scene which may be a problem too in clinical uses.

Thus, we believe that the Kinect$^{TM}$ Skeleton Tracking has significant potential for rehabilitation purposes but only in a controlled body posture, like abduction/adduction, and in an environment without objects in the scene. To overcome these limitations a new algorithm based on the depth map information was developed and proved to be a helpful tool when movements like flexion/extension and internal/external rotation are required or the general trends in the movement are insufficient. The proposed algorithm obtained a ME less than 1.5 degrees in all movements studied. However, the developed algorithm has limitations related with the calibration process which needs to be improved. Firstly, in the decision between flexion/extension and internal/external rotation, the subject needs to begin the movement within a limited time. Otherwise, the movement may be wrongly detected. Secondly, once a movement is detected, the algorithm does not identify a new one if the movement was changed during the acquisition. Thus, the application must be restarted. Both problems can be solved adding periodic evaluations of each movement being performed. The real-time application required for using it with biofeedback is also a limitation that needs to be improved in future work.

Using the Kinect$^{TM}$ sensor for ROM measurements provides benefits when compared to our reference system, a triaxial accelerometer. The major benefit is that it does not need anything attached to the patient. Other benefit is that when the accelerometer is used, the subject needs to perform the arm movements at low rates, because dynamic acceleration influences the measures [37]. Moreover, perfect alignment with the bone is difficult. Therefore, to minimise the errors, the measure of variations instead of the maximum angle performed is done, using this way a measure of relative angles, i.e, absolute angles are not possible. With our algorithm we can measure the angles using the thorax as reference. Thus, if the patient tilts the body because it makes it easier to achieve a higher angle, the ROM measurements will not increase for that and the system can alert the patient to not compensate with the body. With the triaxial accelerometer it is possible to use the thorax as reference only if another triaxial accelerometer was attached in the subject's thorax.

Along this research, a paper that presents the development of the new algorithm was submitted to an international conference.

In conclusion, the Kinect$^{TM}$ Skeleton Tracking can be useful for some applications where the general trends are sufficient. Our algorithm, that proved to have comparable data with the triaxial accelerometer, is able to overcome the limitations of the Kinect$^{TM}$

Skeleton Tracking and the triaxial accelerometer, being a very promising technology for ROM measurements.

## 6.2 Future Work

To introduce the developed algorithm in clinical practice a few improvements should be carried out. In the following list those tasks are presented.

- **Real-time processing:** In this research, the use of the Python programming language showed limitations related to the performance in real-time applications. Thus, it is proposed to continue the conversion from Python to C# and therefore to introduce a system using our method in the rehabilitation field, using biofeedback.
- **To introduce the ROM measurements in a biofeedback system:** The *PLUX - Wireless Biosignals* has a biofeedback system named physioPlux. This system is capable of analysing several biosignals as the EMG. Once the patients with muscular abnormality have normally problems with their ROM, it would be relevant to introduce a new biofeedback protocol to use, at the same time, the ROM and EMG information. Therefore the patients would see their evolution in both measures.
- **Further validation:** Although the different scenarios in the validation performed were tested, only subjects without shoulder pain were used. Thus, a more exhaustive validation with subjects with any muscular abnormality should be done, to prove that their possible body differences were not influencing the precision of the proposed algorithm.
- **To study other body parts:** This thesis only studied the ROM measurements applied to the glenohumeral movements. However, the introduction of the same principles used in this research to compute ROM measurements should be extended to other body parts.

# Bibliography


[1] Rodolfo Telo Martins de Abreu. *Algorithms for information extraction and signal annotation on long-term biosignals using clustering techniques*. PhD thesis, Faculdade de Ciências e Tecnologia, 2012.

[2] Rezual Begg and Marimuthu Palaniswami. *Computational Intelligence for Movement Sciences: Neural Networks and Other Emerging Techniques*. Idea Group Publishing, 2006.

[3] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[4] John C. Russ. *The Image Processing Handbook*. Taylor and Francis Group, LLC, sixth edition, 2011.

[5] Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing*. Prentice-Hall, Inc., second edition, 2002.

[6] Kathryn LaBelle. *Evaluation of Kinect Joint Tracking for Clinical and In-Home Stroke Rehabilitation Tools*. PhD thesis, Thesis, Indiana, 2011.

[7] Lars Lünenburger, Gery Colombo, and Robert Riener. Biofeedback for robotic gait rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 4(1):1, 2007.

[8] Gregorij Kurillo, Jay J Han, Štěpán Obdržálek, Posu Yan, Richard T Abresch, Alina Nicorici, Ruzena Bajcsy, et al. Upper extremity reachable workspace evaluation with kinect. *Studies in health technology and informatics*, 184:247–253, 2012.

[9] Naofumi Kitsunezaki, Eijiro Adachi, Takashi Masuda, and Jun-ichi Mizusawa. Kinect applications for the physical rehabilitation. In *Medical Measurements and Applications Proceedings (MeMeA), 2013 IEEE International Symposium on*, pages 294–299. IEEE, 2013.

[10] Huiyu Zhou and Huosheng Hu. Human motion tracking for rehabilitation—a survey. *Biomedical Signal Processing and Control*, 3(1):1–18, 2008.

[11] Tilak Dutta. Evaluation of the kinect$^{TM}$ sensor for 3-d kinematic measurement in the workplace. *Applied ergonomics*, 43(4):645–649, 2012.

[12] Vicon system. http://www.vicon.com/_pdfs/Vicon_Tracker_lr.pdf, Accessed:25/02/2013.

[13] Msdn library. http://msdn.microsoft.com/en-us/library/jj131033.aspx, Accessed:10/02/2013.

[14] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis Argyros. Efficient model-based 3d tracking of hand articulations using kinect. *BMVC, Aug*, 2:1–11, 2011.

[15] Lu Xia, Chia-Chih Chen, and JK Aggarwal. Human detection using depth information by kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 15–22. IEEE, 2011.

[16] A Da Gama, Thiago Chaves, Lucas Figueiredo, and Veronica Teichrieb. Poster: Improving motor rehabilitation process through a natural interaction based system using kinect sensor. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 145–146. IEEE, 2012.

[17] Ross A Clark, Yong-Hao Pua, Karine Fortin, Callan Ritchie, Kate E Webster, Linda Denehy, and Adam L Bryant. Validity of the microsoft kinect for assessment of postural control. *Gait & Posture*, 36(3):372–377, 2012.

[18] Štepán Obdrzálek, Gregorij Kurillo, Ferda Ofli, Ruzena Bajcsy, Edmund Seto, Holly Jimison, and Michael Pavel. Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population. *Engineering in Medicine and Biology Society*, pages 1188–1193, 2012.

[19] Antônio Padilha Lanari Bó, Mitsuhiro Hayashibe, and Philippe Poignet. Joint angle estimation in rehabilitation with inertial sensors and its integration with kinect. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 3479–3483. IEEE, 2011.

[20] Hossein Mousavi Hondori, Maryam Khademi, and Cristina Videira Lopes. Monitoring intake gestures using sensor fusion (microsoft kinect and inertial sensors) for smart home tele-rehab setting. In *2012 1st Annual IEEE Healthcare Innovation Conference*, 2012.

[21] Adso Fern'ndez-Baena, Antonio Susin, and Xavier Lligadas. Biomechanical validation of upper-body and lower-body joint movements of kinect motion capture data for rehabilitation treatments. In *Intelligent Networking and Collaborative Systems (INCoS), 2012 4th International Conference on*, pages 656–661. IEEE, 2012.

[22] Vangelis Metsis, Pat Jangyodsuk, Vassilis Athitsos, Maura Iversen, and Fillia Makedon. Computer aided rehabilitation for patients with rheumatoid arthritis. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 97–102. IEEE, 2013.

[23] CK Fernando and JV Basmajian. Biofeedback in physical medicine and rehabilitation. *Applied Psychophysiology and Biofeedback*, 3(4):435–455, 1978.

[24] Huang He and He Jiping. Recent developments in biofeedback for neuromotor rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 3, 2006.

[25] Deborah S Nichols. Balance retraining after stroke using force platform biofeedback. *Physical Therapy*, 77(5):553–558, 1997.

[26] Gabriel Zachmann. Distortion correction of magnetic fields for position tracking. In *Computer Graphics International, 1997. Proceedings*, pages 213–220. IEEE, 1997.

[27] Judd S Day, Duncan J Murdoch, Genevieve A Dumas, et al. Calibration of position and angular data from a magnetic tracking device. *Journal of biomechanics*, 33(8):1039, 2000.

[28] Huiyu Zhou, Thomas Stone, Huosheng Hu, and Nigel Harris. Use of multiple wearable inertial sensors in upper limb motion tracking. *Medical Engineering and Physics*, 30(1):123, 2008.

[29] Huiyu Zhou and Huosheng Hu. Reducing drifts in the inertial measurements of wrist and elbow positions. *Instrumentation and Measurement, IEEE Transactions on*, 59(3):575–585, 2010.

[30] Neuza Filipa Martins Nunes. *Algorithms for time series clustering applied to biomedical signals*. PhD thesis, Faculdade de Ciências e Tecnologia, 2011.

[31] James G Richards. The measurement of human motion: A comparison of commercially available systems. *Human Movement Science*, 18(5):589–602, 1999.

[32] Benjamin Langmann, Klaus Hartmann, and Otmar Loffeld. Depth camera technology comparison and performance evaluation. In *Pattern Recognition Applications and Methods, 2012 International Conference on*, pages 438–444. SciTePress, 2012.

[33] Enrique Ramos Melgar and Ciriaco Castro Diez. *Arduino and Kinect Projects*. Springer, 2012.

[34] Greg Borenstein. *Making Things See*. O'Reilly Media, first edition, 2012.

[35] Fabrizio Pece, Jan Kautz, and Tim Weyrich. Three depth-camera technologies compared. *Engineering in Medicine and Biology Society*, pages 1188–1193, 2012.

[36] Alberto Martin and Sabri Tosunoglu. Image processing techniques for machine vision. *Miami, Florida*, 2000.

[37] Eva Bernmark and Christina Wiktorin. A triaxial accelerometer for measuring arm movements. *Applied Ergonomics*, 33(6):541–547, 2002.

[38] J Beckert, F Silva, and S Palma. Inter-rater reliability of the visual estimation of shoulder abduction angles and the agreement of measurements with an acelerometer. *Proceedings of ECSS2009, Oslo, Norway*, 2009.

[39] STMicroelectronics. Tilt measurement using a low-g 3-axis accelerometer. Technical report, AN3182, 2010.

[40] Andrew Loomis. *Figure drawing for all it's worth*. Penguin Group (USA) Incorporated, 1943.

# Flowchart

In this appendix a flowchart that systematized the sequence of events of the developed algorithm is presented.

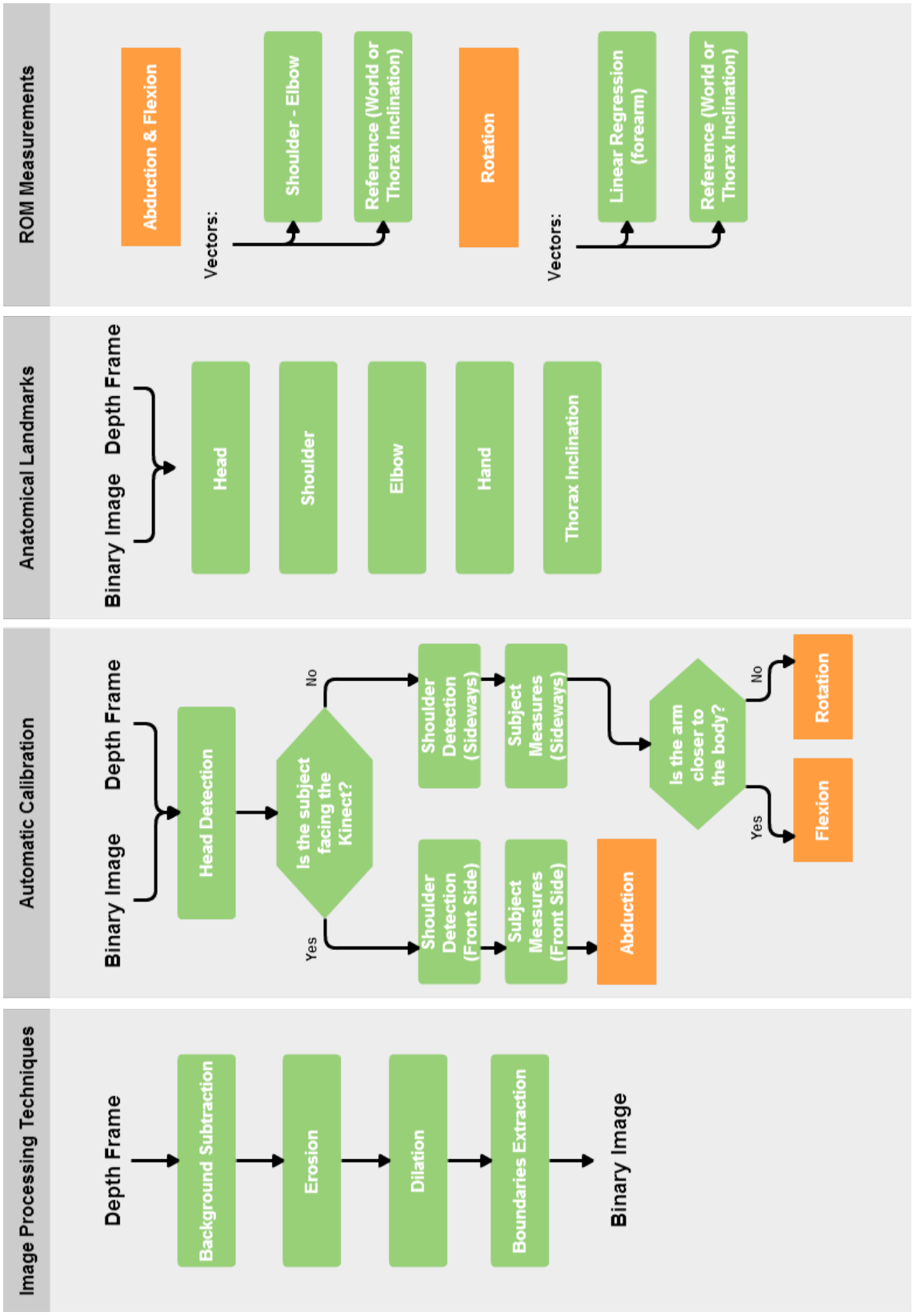


Image Processing Techniques
Depth Frame
Background Subtraction
Erosion
Dilation
Boundaries Extraction
Binary Image
Automatic Calibration
Binary Image
Depth Frame
Head Detection
Is the subject facing the Kinect?
No
Yes
Shoulder Detection (Sideways)
Subject Measures (Sideways)
Shoulder Detection (Front Side)
Subject Measures (Front Side)
Abduction
Is the arm closer to the body?
No
Yes
Rotation
Flexion
Anatomical Landmarks
Binary Image
Depth Frame
Head
Shoulder
Elbow
Hand
Thorax Inclination
ROM Measurements
Abduction & Flexion
Vectors:
Shoulder - Elbow
Reference (World or Thorax Inclination)
Rotation
Vectors:
Linear Regression (forearm)
Reference (World or Thorax Inclination)