José Brandão Tavares

Automatic Diagnosis System for Human Gait Pathologies using Computer Vision

Coimbra 2015

· U C ·

UNIVERSIDADE DE COIMBRA

· U C ·

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Automatic Diagnosis System for Human Gait Pathologies using Computer Vision

José Brandão Tavares

Coimbra, September 2015

Automatic Diagnosis System for Human Gait Pathologies using Computer Vision

Supervisor:

Professor António Paulo Mendes Breda Dias Coimbra

Co-Supervisors:

Professor Manuel Marques Crisóstomo
Professor João Paulo Morais Ferreira

Jury:

Prof. Helder de Jesus Araújo (President)

Prof. Jorge Manuel Moreira de Campos Pereira Batista
Prof. Manuel Marques Crisóstomo

Dissertation submitted in partial fulfilment for the degree of Master of Science in Electrical and Computer Engineering.

Coimbra, September 2015

Acknowledgements

I would like to publicly express my gratitude towards everyone that in any way contributed to realization of this thesis.

To my supervisor, Professor A. Paulo Coimbra for all the patience, help and guidance.

To my Co-Supervisors, Professor Manuel Crisóstomo and Professor João Ferreira for all the support, guidance and knowledge.

To Engineer Paulo Ferreira for all the answers to my questions, all the help and patience.

To my parents for always being there for me and for providing me with all the support I needed.

To my brother for always making me go mad.

To my uncle José António for always being there for me.

To my aunt Angelina and uncle Mário for having received me into their home when i moved to college and everything else.

To Natacha P. for all the moment we shared and for the last minute revisions.

To João Santos for being a role model and a good friend without even realizing it and for all the help during the last years.

To Sérgio Zuca for supporting me while I was "on fire" and for the friendship over the years.

To Gonçalo for being a good friend and for all the help and suggestions provided.

To Soraia for always caring.

To Diogo Silva for the all the moral XM.

To Natália for all the moral support, the moments we share and for remembering-me

and to all of those that I didn't mention but made a difference to me.

Resumo

Este trabalho insere-se no projeto "Novas Tecnologias para apoio à Saúde e Qualidade de Vida" e foca-se no melhoramento de um sistema de análise da marcha que usa visão por computador.

Durante a última década o estudo da marcha humana tem sido alvo de bastante interesse para se realizarem diagnósticos médicos, quantificando o estado móvel de determinada perturbação médica e portanto, ajudando na prescrição de tratamento e no acompanhamento [30]. A qualidade da marcha pode ser medida tendo em conta vários parâmetros do passo, tais como a velocidade, o comprimento e cadência que por sua vez podem ser quantificados usando um sistema computacional. O diagnóstico médico é útil e prático, no entanto carece de precisão e objectividade [30] fornecida por dados quantitativos tridimensionais. Embora os computadores sejam menos subjetivos que as pessoas ao realizar a análise da marcha, existem problemas complexos que precisam de ser resolvidos. Alguns autores referem o quão importante é a perspectiva de uma câmara face ao objecto de interesse [28] [6] [18]. Definir uma perspectiva exata no mundo real, além de ser demasiado moroso, é também bastante difícil senão impossível. Resolver este tipo de problema, ou pelo menos minimizar os seus efeitos no resultado final, é altamente desejável.

De forma a melhorar o sistema existente, um novo método para o módulo de alinhamento das câmaras foi aplicado baseado na estimação da pose para uma câmara usando correspondências de pontos 3D para 2D. Utilizando a pose estimada, são dadas instruções ao utilizador para que este consiga corrigir a pose da câmara para a pose desejada. O novo módulo de alinhamento das câmaras é mais rápido, fiável e em conjunto com uma simulação criada foi possível concluir que dadas as condições necessárias para que a câmara seja considerada alinhada, o erro introduzido pelo desvio da pose ainda existente não é significativo.

Palavras-Chave: estimação da pose da câmara, análise da marcha, ângulos da marcha, visão por computador.

Abstract

This work is part of the project "Novas Tecnologias para apoio à Saúde e Qualidade de Vida" (New technologies for healthcare and quality of life support). It focuses on improving a low cost gait analysis system that uses computer vision.

During the last decades the study of the human gait has been a subject of great interest to make medical diagnosis by quantifying the mobility state of a medical disorder and therefore help prescribe treatment and assess its outcome [30]. Gait quality can be measured by taking into account several stride parameters such as walking speed, step length and cadence which can be accurately quantified by a computer system. Caregiver assessment is useful and practical, however they lack the precision and objectivity [30] provided by three dimensional quantitative data. Despite computers being less subjective than people when performing gait analysis there are some complex problems to be dealt with. Some authors have pointed out how important camera perspective is towards the objects of interest [28] [6] [18] and setting an exact camera point of view in real life is time consuming and difficult, if not impossible. Solving this kind of issue or at least minimizing its effects on the final results is highly desirable.

In order to improve upon an existing gait analysis system, a new method for its camera alignment module was applied, which is based on the pose estimation for a calibrated camera using 3D to 2D point correspondences. Using the estimated pose, instructions are given to the user in order to correct the camera into the desired pose. The new camera alignment module is faster, more reliable and in conjunction with a simulation created, it was concluded that for the given alignment conditions, the error introduced by the deviations in the camera pose are not significant.

Keywords: camera pose estimation, gait analysis, gait angles, computer vision

“There can be no life without change, and to be afraid of what is different or unfamiliar is to be afraid of life."

— Theodore Roosevelt

Contents

List of Acronyms

FPS	Frames per Second
CMOS	Complementary Metal–Oxide Semiconductor
VGA	Video Graphics Array
QREN	Quadro de Referência Estratégica Nacional
i.e.	Latin expression "*id est*" ("that is")
e.g.	Latin expression "*exempli gratia*" ("for example")
CW	Clockwise
USB	Universal Serial Bus
HD	High Definition
FOV	Field Of View
CG	Computer Generated
OpenCV	Open Source Computer Vision
CPU	Central Processing Unit
RAM	Random Access Memory
DOF	Degrees Of Freedom
RGB	Red, Green and Blue
PnP	Perspective-n-Point

List of Figures

List of Tables

Chapter 1

Introduction

This document describes the work done in "Novas Tecnologias para apoio à Saúde e Qualidade de Vida" (New technologies for healthcare and quality of life support) project financed by Quadro de Referência Estratégica Nacional (QREN) - Mais Centro[1], which took place in the Electrical and Computer Engineering department from the Faculty of Sciences and Technology of the University of Coimbra and at the Physical Rehabilitation Centre of the University of Coimbra Hospitals. The main goal of this project is the development of a low cost human gait acquisition and analysis system using computer vision techniques as an alternative to Vicon Bonita[2] and Optotrak Certus[3].

Regarding this master's thesis work, since some of the software was already created and the equipment chosen, it was divided into three main phases: acquaintance with the already developed software, workflow, equipment and human gait acquisition techniques used in this project; improvement of the camera alignment process which is required to do each time the equipment is moved and can take a considerable amount of time; study of the implications of not having the cameras properly aligned and how to overcome that problem since in the real world it is impossible to place each camera exactly on the desired place and orientation.

This chapter presents this thesis motivation, main objectives and the outline.

1.1 Thesis Motivation

During the last decades the study of the human gait has been a subject of great interest in diverse areas such as biomechanics [3], psychology [15] [32], medical sciences [37] [34] and

[1] http://www.qren.pt/

[2] http://www.vicon.com/products/camera-systems/bonita

[3] http://www.ndigital.com/msci/products/optotrak-certus/

others[1] since the information that can be gathered from human gait analysis has immense applications and goes far beyond the ability to recognise a set of moving dots as a human walker. It is a pattern of human locomotion and can be described by kinetic or kinematics characteristics [2]. It can be used to determine the gender of a person [4] [35], to identify a subject [34], to make athletic performance analysis [1] or even to make medical diagnosis by quantifying the mobility state of a medical disorder and therefore help prescribe treatment and assess its outcome [30].

Gait quality can be measured by taking into account several stride parameters such as walking speed, step length and cadence which can be accurately quantified by a computer system. Caregiver assessment is useful and practical, however they lack the precision and objectivity [30] provided by three dimensional quantitative data. Despite computers being less subjective than people when performing gait analysis there are some complex problems to be dealt with other than the task of interpreting the data itself, which is not an easy task. For example, during a single stride, multiple body joints and segments contained on several planes have to be analysed as a function of time and the motions they perform are coupled with other joints and segments and across other planes. It is therefore, an interconnected system and a problem present in a specific joint will most certainly manifest its effects across several parts of the body. Regarding the system described in this report, one of those problems relates to the requirement of having the equipment placed in a specific manner (e.g. position, orientation) so the data acquired by both cameras in multiple analysis can be processed in a consistent way, i.e. it would make no sense to interpret joint positions in the sagittal plane of a patient recovering from some sort of disorder if the data gathered several months after the first test, is acquired with the cameras in a new configuration. The data wouldn't be coherent and the joint angles wouldn't be perceived as the same even if patient walked exactly as in the previous test.

Some authors have pointed out how important camera perspective is towards the objects of interest [28] [6] [18] and setting an exact camera point of view in real life is time consuming and difficult, if not impossible. Solving this kind of issue or at least minimizing its effects on the final results is highly desirable.

1.2 Main objectives

The first objective of this work was to improve the camera alignment software module since placing each camera in the correct position and orientation (pose) was a time consuming task and very prone to introduce some error in the data to be collected. The second objective was to find out the outcome of having a camera with an incorrect pose, regarding the joint angles to be measured by it. The last objective was, if needed, to implement some corrections to compensate, to some extent, the error introduced by each camera incorrect pose considering it is impossible to have them exactly in the correct place in a real world scenario.

1.3 Dissertation Outline

This document is divided in six main chapters. In the first are introduced the context of the work, the motivation that lead to it and its main objectives. The second Chapter presents some theory of support which explains in a succinct form some of the concepts that are used. Third chapter describes the overall Gait Analysis System in order to highlight some of the issues that will be worked on. Chapter four presents a simulation where projections from various camera poses are used to assess the importance of having the image plane of a camera parallel to the patient's sagittal plane during the acquisition process and a possible method to compensate, to some extent, the error that might be introduced by not having the cameras correctly aligned. Chapter five describes the practical implementation of the improvements applied to the Gait Analysis System as a result of this work. In Chapter six are discussed the conclusions and future work.

Chapter 2

Theory Of Support

This chapter explains in a generic way some of the concepts that are the basis of this project. It starts by describing how various coordinate systems (or reference frames) and its points can be mapped into another. It also introduces the use of Homogeneous Coordinates and the advantages of using them in Projective Geometry. Lastly, using the concepts introduced, it is explained the process of the image formation and the notation used.

2.1 Coordinate System Changes and Rigid Transformations

When there are multiple reference systems being considered at the same time it is convenient to refer to a point using a notation that indicates in which reference system that point is being referred to. In this report the following notation will be used to represent the coordinate vector of a point P in a frame A:

$$ {}^{A}P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \overrightarrow{{}^{A}OP} \tag{2.1}$$

If another reference system is considered, let's say, B, and the distance from its origin to the origin of A is known, point P can be represented in relation to the new frame as ${}^{B}P$. If both reference frames have their axes versors parallel to each others and are only shifted by some distance, the rigid body transformation that relates them is a pure translation. The Figure 2.1 shows a representation of a pure translation between two frames.

Mapping the point ${}^{A}P$ into the reference frame B can be done by means of a simple

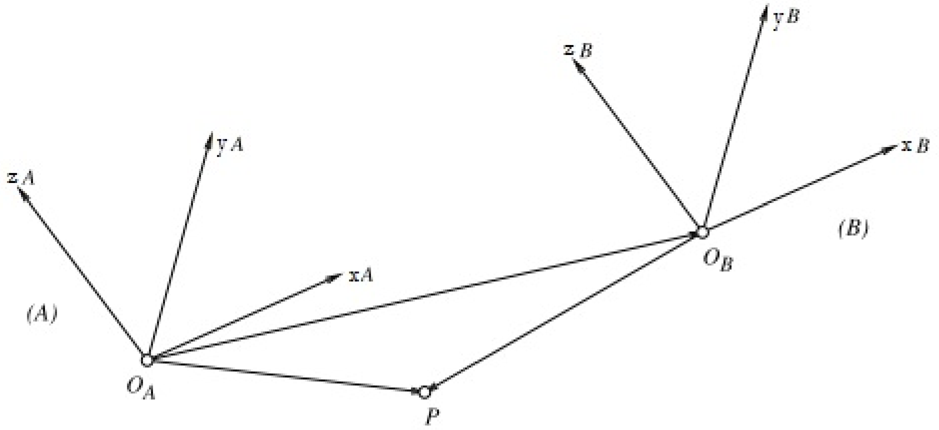


Figure 2.1: Pure translation between two reference frames [11]

vector operation:

$$^{B}P =^{A} P +^{B} O_A \tag{2.2}$$

Now let's consider the case where both reference frame's origins coincide, that is, $O_A = O_B$ and the versors of their axes are not parallel as shown in figure 2.2. The transformation that relates them is given by a rotation matrix defined by:

$$^{B}_{A}R = \begin{bmatrix} x_A\cdot x_B & y_A\cdot x_B & z_A\cdot x_B \\ x_A\cdot y_B & y_A\cdot y_B & z_A\cdot y_B \\ x_A\cdot z_B & y_A\cdot z_B & z_A\cdot z_B \end{bmatrix} \tag{2.3}$$

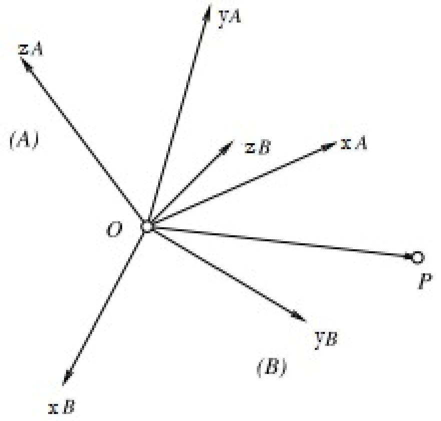


Figure 2.2: Pure rotation between two reference frames [11]

The rotation matrix from (2.3) describes the frame (A) in the coordinate system (B).

For the case of a pure rotation the point ${}^{A}P$ can be mapped in the frame B by premultiplying it by the rotation matrix that relates both frames:

$$ {}^{B}P = {}^{B}_{A} R\, {}^{A}P \tag{2.4} $$

A parametrization of a rotation matrix in terms of three elementary rotations about coordinate axis is possible [8][31] in such a way that $R_{zyx} = R_z(\gamma) \cdot R_y(\beta) \cdot R_x(\alpha)$.
α, β and γ are known as the Euler angles and each elementary rotation is described by:

$$ R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix} \tag{2.5} $$

for a rotation around the x axis by $\alpha°$

$$ R_y(\beta) = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix} \tag{2.6} $$

for a rotation around the y axis by $\beta°$

$$ R_z(\gamma) = \begin{bmatrix} cos\gamma & -sin\gamma & 0 \\ sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7} $$

for a rotation around the z axis by $\gamma°$.

When performing such operations there are some important notions to be taken into account like the order in which each rotation is applied to a frame for it matters. Applying a rotation around the x axis of a frame followed by a rotation around the z axis is not the same as doing it the other way around since it yields different results: $R_z(\gamma) \times R_x(\theta) \neq R_x(\theta) \times R_z(\gamma)$. In another note, an example of an useful concept about rotation matrices is that ${}^{A}_{B}R^{T} = {}^{B}_{A} R$. This is pretty useful to describe the rotation that brings the frame (A) into (B) and vice versa.

Now that pure rotation and pure translations have been described it's time to interpret the case where both reference systems do not share the same origin nor their axes versors are parallel, as represented by Figure 2.3. This is called a general Rigid Body Transformation.

Combining Equations 2.4 and 2.2 we get

$$ {}^{B}P = {}^{B}_{A} R\, {}^{A}P + {}^{B} O_A \tag{2.8} $$

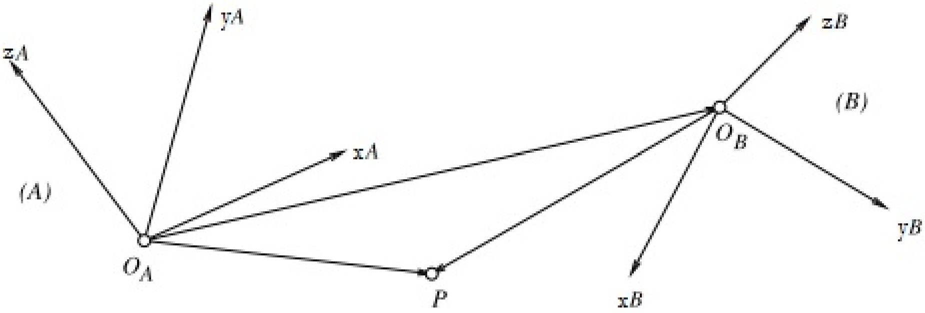


Figure 2.3: A general rigid transformation between two frames [11]

Equation 2.8 can be written in matrix form and re-arranged giving origin to

$$\begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} {}^{B}_{A}R & {}^{B}O_A \\ 0^T & 1 \end{bmatrix}}_{{}^{B}_{A}T} \begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} \tag{2.9}$$

${}^{B}_{A}T$ is called a Transformation Matrix and describes all the transformations that map the point ${}^{A}P$ into the frame (B) giving origin to ${}^{B}P$. Note the use of the Homogeneous coordinates by appending the 1 at the end of vector P. Homogeneous coordinates have some advantages over Cartesian coordinates and are widely used in projective geometry and will be discussed in section 2.2.

2.2 The Projective Geometry and Homogeneous Coordinates

Homogeneous coordinates have played an important role in the development of projective geometry [24] [23] and are used extensively in computer graphics [26] [29]. They play a key role in the representation of three-dimensional objects commonly specified by the use of three Cartesian coordinates denoted by x, y, and z. Homogeneous coordinates augment this description introducing a fourth coordinate which can be viewed as a scale factor and are denoted by wx, wy, wz, w. In general, an homogeneous representation is a description of any n-dimensional space by an $(n+1)$-dimensional space problem. The motivation for its transformation into a higher dimension relies on the fact that the solutions for a lot of geometrical problems are simpler in the higher dimensional space and can then be projected back into its original representation. By transforming a given system into an higher dimensional space, in fact what happens, is that all the primitives are being increased by one dimension, i.e.

points become lines an lines become planes.

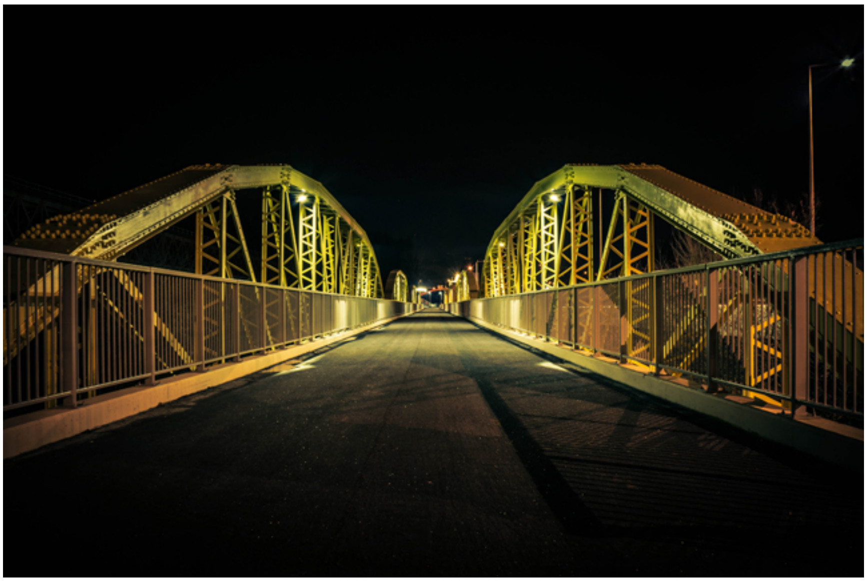


Figure 2.4: A perspective view of a bridge

For a better understanding of the advantages of using homogeneous coordinates and its relation with projective geometry let's consider the image shown in Figure 2.4. The fences on each side of the road are parallel if described in Euclidean geometry, however by looking at the image, that is, the projection of the world elements into an image plane, those lines seem to intersect at a point at infinity, also known as a vanishing point. This is a result of projective geometry since it does not preserve some of the properties of Euclidean geometry: distances between points are not preserved nor angles between lines. However lines are still mapped into lines. [36]. In Euclidean geometry the infinity has no satisfactory representation but in projective geometry it can be easily represented using homogeneous coordinates by $w = 0$. As mentioned before, the conversion between Cartesian and Homogeneous coordinates can be easily done. For simplicity reasons the two-dimensional space will be considered for the following examples.

$$\underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{Cartesian} \Rightarrow \underbrace{\begin{bmatrix} wx \\ wy \\ w \end{bmatrix}}_{Homogeneous} \tag{2.10}$$

and *vice versa*

$$\underbrace{\begin{bmatrix} x \\ y \\ w \end{bmatrix}}_{Homogeneous} \Rightarrow \underbrace{\begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}}_{Cartesian} \tag{2.11}$$

When converting Homogeneous to Cartesian coordinate an important property is noticed: Homogeneous coordinates are scale invariant.

$$\underbrace{\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}}_{Homogeneous} \Rightarrow \underbrace{\begin{bmatrix} \frac{1}{1} \\ \frac{2}{1} \end{bmatrix}}_{Cartesian} ; w = 1 \tag{2.12}$$

$$\begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{2}{2} \\ \frac{4}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{1} \\ \frac{2}{1} \end{bmatrix} ; w = 2 \tag{2.13}$$

$$\begin{bmatrix} 3 \\ 6 \\ 3 \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{3}{3} \\ \frac{6}{3} \end{bmatrix} = \begin{bmatrix} \frac{1}{1} \\ \frac{2}{1} \end{bmatrix} ; w = 3 \tag{2.14}$$

All the Homogeneous points shown in (2.12), (2.13) and (2.14) represent in fact, the same Cartesian point, hence the name Homogeneous. The Figure 2.5 shows a graphical representation of the previous example.

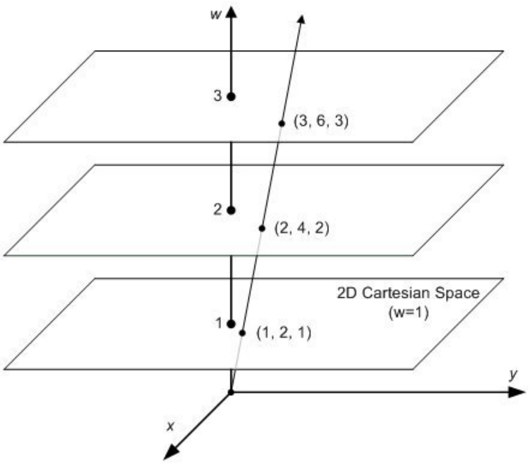


Figure 2.5: The projective space

Drawing a line passing from each Homogeneous point and through the origin creates a set of lines that are collinear and represent the same Cartesian point. i.e.: every single line through the origin can be written as $k \times [x, y, w]^T$ as long as $[x, y, w]^T \neq [0, 0, 0]^T$. By convention, when converting from Cartesian to Homogeneous the new coordinate usually has the value $w = 1$ and is also known as the normalized projection plane in the projective space. [12]

2.2.1 The Pinhole Camera Model

The pinhole model is the most common geometric model of an intensity camera. It describes the process in which points of a scene are mapped into the corresponding image points by means of a projection. It is also known as the perspective camera model model. Figure 2.6 shows a physical representation of it.

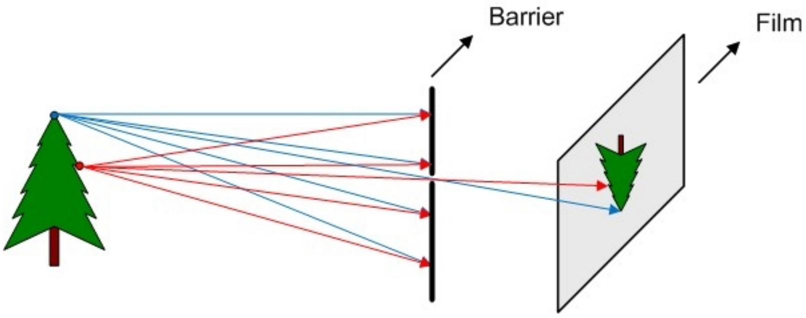
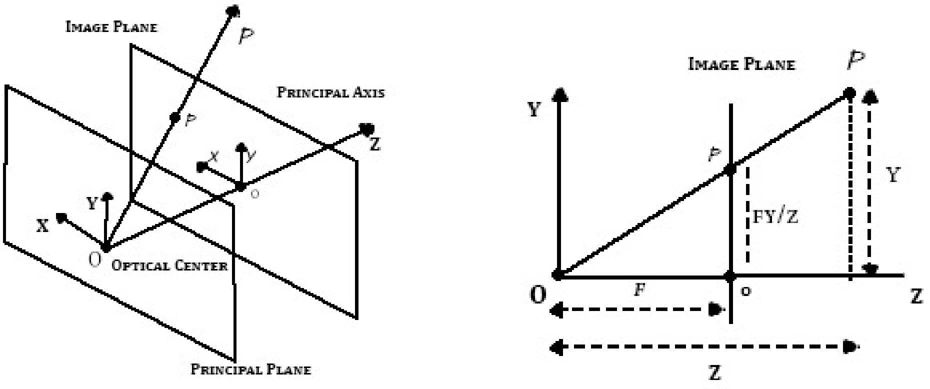


Figure 2.6: The pinhole camera model

Red and blue lines represent the light being reflected from distinct points of the object. If the hole is big enough and a ray coming from the same point of the object hits the film it will be recorded several times in different places of the film thus making the resulting image out of focus, however for the geometric model description it is considered the barrier has a small hole that only lets some rays of light pass through it. The resulting image is a 2D ideal representation of the 3D object. The hole in the barrier is known as the optical centre, the plane where the 2D image is formed is the image plane and the distance between them is the focal length F, in meters.

The image plane can be drawn in front of the Optical Centre as shown in Figure 2.7a at a distance F. This way the corresponding image points won't be mirrored as in Figure 2.6 and the projected point p is located in the intersection from the image plane and the ray passing between P and the Optical Centre (O). The optical centre serve as the camera frame reference origin and, by convention, has the Z axis pointing in the direction of the



(a) 3D perspective camera model

(b) YZ plane: Triangle similarity

Figure 2.7: The Perspective Camera Geometry: Figure 2.7a illustrates the projection of point P into the image plane; Figure 2.7b illustrates the same situation in the YZ plane.

scene. The image plane is therefore perpendicular to it and at their intersection forms the principal point o. The Y and Y axes are also perpendicular to the Z axis, however neither of them has the constraint of pointing into any particular direction apart from having to be orthogonal in respect to each others and collinear with the principal plane.

The coordinates of point P are given in respect to the camera frame previously discussed and have the form $[X, Y, Z]^T$. They are projected into the Image Plane giving origin to p which is now a 2D point in the form of $[x, y]^T$ in respect to the principal point (o). By looking at the geometric model from a 2D point of view such as the plane YZ (Figure 2.7b) a noteworthy relation is noticed between both points and their coordinates: they form similar triangles and therefore, their sides can be related.

$$\frac{x}{F} = \frac{X}{Z} \tag{2.15}$$

$$\frac{y}{F} = \frac{Y}{Z} \tag{2.16}$$

Rewriting (2.15) and (2.16) gives origin to:

$$x = F\frac{X}{Z} \tag{2.17}$$

$$y = F\frac{Y}{Z} \tag{2.18}$$

To note that Equations 2.17 and 2.18 are non-linear and do not preserve distances between points nor angles between lines. Here is where the use of Homogeneous coordinates comes in handy, in fact Equations 2.17 and 2.18 can be rewritten again giving origin to

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{F}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{2.19}$$

and using Homogeneous coordinates (2.19) has the matrix form

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} F & 0 & 0 & 0 \\ 0 & F & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.20}$$

where $[\lambda x, \lambda y, \lambda]^T$ are the Homogeneous image plane coordinates and λ is the scaling factor. If P is given in meters and in the camera coordinate frame, then p is the corresponding 2D image point in meters too and the non-linear problem is no longer an issue. There are a few extra steps to be taken since the desired projected points in the camera are represented by pixels and their coordinates are in respect to the top left corner of the image instead of their centre. This next step transforms the image plane coordinates into pixel coordinates. Given the pixel size S_x, S_y, in meters, and the image centre o_x, o_y of the camera sensor, in pixels, Equations 2.17 and 2.18 can be re-written again to contain these parameters becoming

$$x = \underbrace{\frac{F}{S_x}}_{f_x} \frac{X}{Z} + o_x \tag{2.21}$$

$$y = \underbrace{\frac{F}{S_y}}_{f_y} \frac{Y}{Z} + o_y \tag{2.22}$$

where f_x, f_y is the focal length in pixels. In matrix form we get

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{K|0^3} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.23}$$

and the Image Coordinates $[x, y]^T$ are now in pixels. Ideally all the camera sensors are centred in the camera body but in practice, due to the fabric process, they can be slightly offset. The same principle applies to the pixel size. They should be square, that is, $S_x = S_y$ but in a real physical camera that might not be the case [12]. There are more camera parameters that should be taken into account like lens distortion, however for this case it is considered that the model uses an ideal camera.

Matrix K is called the intrinsic parameters matrix (M_{int}). If a given point is not referenced in the camera frame, it has to be mapped before applying Equation 2.23. The set of transformations that relates both frames is described by the external earameters matrix (M_{ext}). The whole process is described in 2.24

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}_{M_{int}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{M_{ext}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{2.24}$$

Figure 2.8 resumes the process of image formation into a simple Diagram.

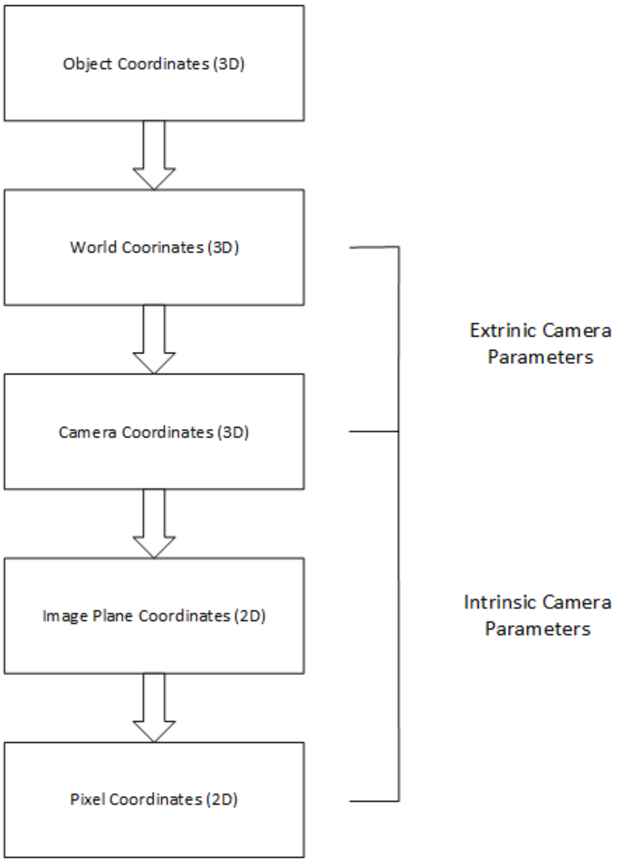


Figure 2.8: The image formation process

Chapter 3

Gait Acquisition and Analysis System Description

This chapter describes in short, the equipment used and the software developed and used for the gait acquisition and analysis system as well as its workflow. To avoid some ambiguity, the person who operates the software will be referred as the user whilst the person whose joint data will be recorded is the patient. The workflow is, by definition, the set of relationships between all the activities in a project, from start to finish [9] and can be divided into three main phases that can be run independently: Equipment set-up, data acquisition and finally, data processing. For the purposes of this chapter the workflow will be described as a continuous process.

3.1 Physical Setup

The data acquisition system is composed by two Logitech C270 webcams, each one placed on either side of a treadmill and connected to a computer. They are rotated 90º Clockwise (CW) and placed on top of a tripod at a distance of 2.35 m of each treadmill's side border at an height of 1.1 m facing each other with their optical axes being collinear. This way each camera captures the movements of each side of the patient during the acquisition process while having its image plane parallel to the patient's sagittal plane (Figure 3.1).

Some markers are also placed on both cameras and treadmill. They are used to help positioning each camera in the correct place and orientation during the alignment process and later on, to help determine the 3D placement of the acquired data by functioning as known 3D points which can be detected and related to. In Figure 3.2 is shown the left

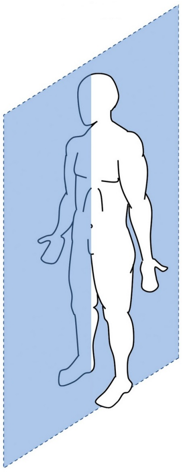


Figure 3.1: Human sagittal plane

camera used with its markers placed, which serve as a visual guide during the alignment process. The right camera looks similar, as well as its markers.

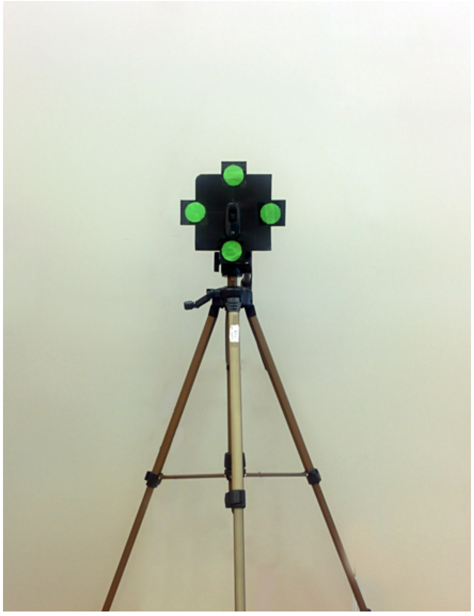


Figure 3.2: Left side Webcam and its markers

The treadmill used is a Newfit, model JS E4002, and has 6 markers placed on it: Four delimiting its deck area, around the belt, and two on the top near the hand bars. Figure 3.3 shows a 3D Computer Generated (CG) representation of the setup described with all the relevant dimensions, including the markers placed on the treadmill and their number designations.

The markers used measure 26 mm in diameter and are green. This project setup has in consideration the requirement of being able to move the equipment to other facilities (e.g.: a room in an hospital or clinic) which can have distinct background colours and illumination.

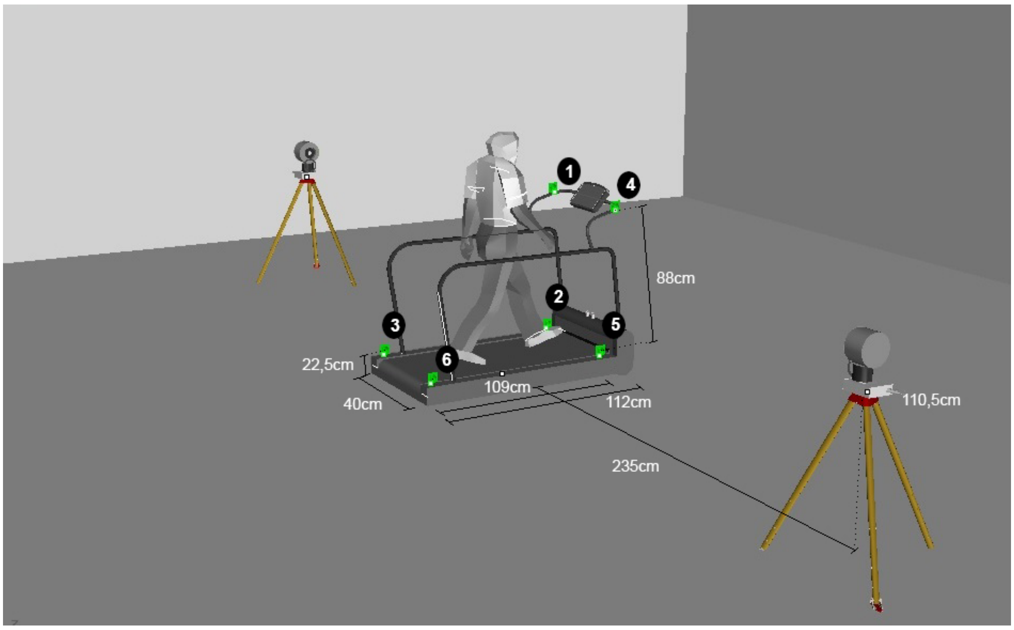


Figure 3.3: CG scenario

For those reasons the marks used can be replaced by another colour and the software can be easily readjusted to detect them.

3.2 Cameras

There are two USB 2.0 Logitech C270 webcams fitted with a Complementary Metal–Oxide Semiconductor (CMOS) sensor capable of acquiring video up to HD definition (1280x720 pixels), however this particular camera model is only capable of capturing video at 30 Frames per Second (FPS) at a resolution up to 640x480 pixels and for that reason it is the image resolution used in this project.

Figure 3.4 shows the form and shape of the reported model without any markers on it. It has a fixed focal length of $4.0mm$ and a Field Of View (FOV) of 60. Any other technical specifications are not known, such as the CMOS size and pixel size or other relevant parameters. The manufacturer has been contacted via e-mail to check if it could provide more information about this camera model, however the reply stated that apart from the info listed on their website [33], none was available. Each camera was calibrated using the Camera Calibrator Application [21] present in Computer Vision System Toolbox from MATLAB and a checkerboard with 29 mm sized squares.

Figure 3.4: Logitech C270 webcam as presented by its manufacturer [33]

3.3 Used Resources

The software was created and developed using MATLAB 2014a and its Computer Vision System Toolbox. It was also used Open Source Computer Vision (OpenCV) library[1] V2.4.9 for windows compiled for MATLAB using MEXOPENCV development kit [2]. The computer systems used were a Desktop with Windows 7 Professional 64 bit powered by an Intel™ Core i7-4774k CPU clocked at 3.50 GHz, with 16 GB of RAM and a laptop from ASUS™, model N61JQ, using Windows 8, 64 bit.

3.4 System Workflow

This section describes in a brief way each phase of the capture and analysis system before any modification developed during this thesis was applied. It's given a particular emphasis on the camera alignment module for its design has been redone and improved as it was one of the main objectives of this work.

3.4.1 Initial Placement of the Equipment

This phase is only done when the equipment is moved to another location (e.g.: room, facility) or when it is apparent that it is not in its intended configuration (e.g.: a camera pointing in the wrong direction or moved out of place). If any of these conditions is verified the equipment, specially the cameras, need to be placed and oriented accordingly to the

[1] `http://opencv.org/`

[2] `http://vision.is.tohoku.ac.jp/~kyamagu/software/mexopencv/`

setup described in 3.1. It may be required the use of measuring tape to set the position and height of each camera. The next step consists of connecting them to the computer and running the software.

3.4.2 Software and Camera Initialization

The software was created and developed by engineer Paulo Ferreira. When it is initiated a menu appears where the user can select from a drop-down menu, which camera is the left and the right one. There is also a button that activates a preview window that shows an image obtained from each camera. After the selection process, a new window appears, the user can either start the calibration process using the Camera Calibrator Application from MATLAB or load an existing file containing the information obtained from a previous calibration. Calibration parameters are used in several stages of the work e.g.: everytime an image is obtained from any of the cameras the image distortion is corrected using the distortion coefficients from the calibration parameters. The system is now ready to start the camera alignment process.

3.4.3 Camera Alignment Module

The following process will be referred as camera alignment for the reason that its main purpose is to offer some guidance so the user can set the camera in the intended pose (camera position and orientation), that is, with both camera optical axes being collinear, passing through each cameras optical centre and perpendicular to the human sagittal plane. This way the image plane will be parallel to it. They are also set at an width that divides the treadmill floor sides in half and at the same height of the two superior markers of the treadmill.

To start the alignment process a button named "Ajustar Câmara" must be selected. The window now presents in a preview box, a sequence of frames from the camera and starts detecting the marks both from the treadmill and the opposite side camera. The markers placed on the treadmill act as 3D points with known coordinates that can be related to and are mainly used to provide some visual aid during the alignment process. The algorithm used to detect the markers is described in 3.4.4 in greater detail and their corresponding image points are called detected points. Figure 3.5 shows the camera alignment module elements that will be described in this section.

Since the desired position of each element is known, using the focal distance provided

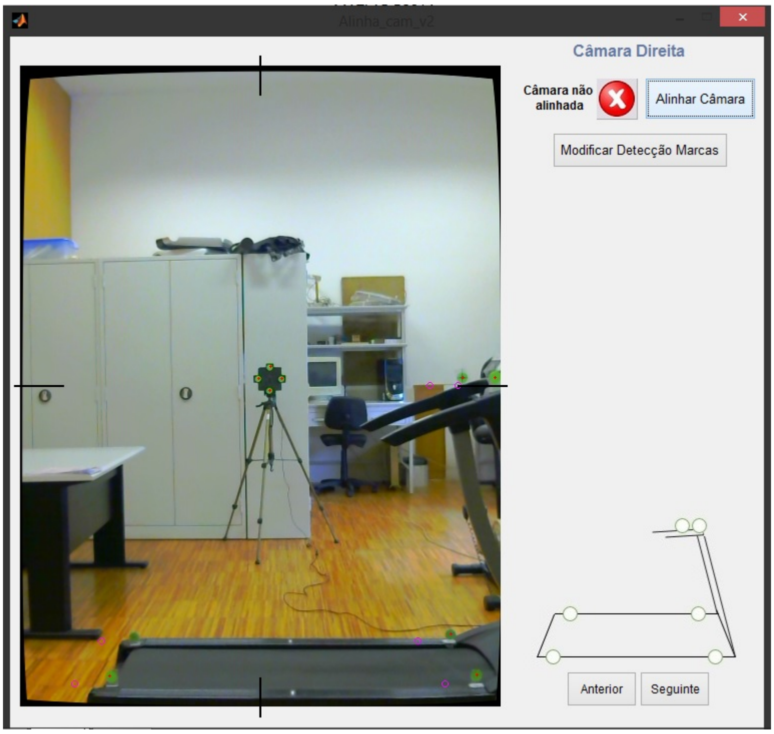


Figure 3.5: Right side camera alignment module window

in the camera model specifications and a computer application called Rhinoceros 5, the corresponding marker projections on the image plane were obtained. They are called target points and are plotted over the images in the preview box (purple circumference). As the name suggests, they act as targets for each detected point (red dots). The user tries to match both target and detected point pairs by moving the camera around. A close up of Figure 3.5 shows in higher detail both types of points (Fig 3.6)

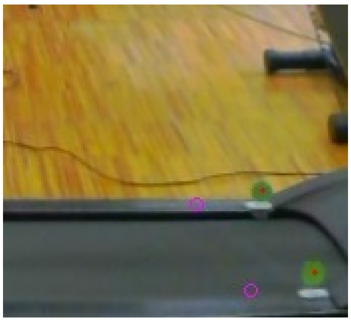


Figure 3.6: Target point (purple circumference) and detected point (red dot)

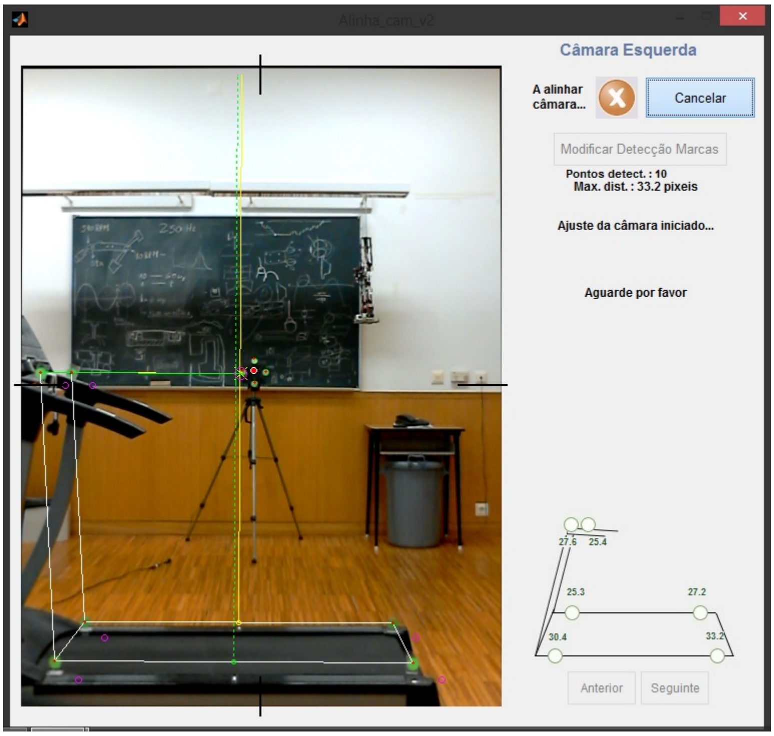
Trying to match a set of 3D points by looking at their 2D projections by continuously re-adjusting the camera (6 Degrees Of Freedom (DOF)) is not only an hard task but also counter intuitive, time consuming and sometimes frustrating. For that reason some extra visual aid is also present. First of all the user starts to set the camera's position and later the orientation.

The point corresponding to the image centre has its horizontal and vertical components marked on each side of the preview frame. It allows the user to set the camera height by aligning markers one and four with the vertical component of the image centre and the cameras lateral offset by aligning the horizontal component of the image centre with the mean points between markers two and three, and five and six, also present on the treadmill (painted in white). It is also evaluated if the detected points are closer to the image centre than the target points. This last step allows an approximate evaluation of the camera's distance to the treadmill.

For the camera's orientation two lines are drawn from markers two and three, and five and six. Starting at the mean point of each segment created and perpendicular to them, another pair of lines is drawn (one of them is a pointed line so it can be discerned). If the camera is in the desired position, and the treadmill's floor is parallel to the ground, both vertical lines and the superior mark that divides the image frame width in half should be collinear.If they aren't the angle formed between them is measured and presented to the user as a rotation around the camera's optical axis. If the vertical lines aren't collinear between them it might indicate that the camera position and/or orientation around its vertical axis needs to be further adjusted. The same principle is applied to evaluate the camera's horizontal axis. Markers one and four should be at the same height of the camera, meaning the vertical component from their image coordinates should be around the same value as the marks that divide the image height in half. Nevertheless a new pair of lines is drawn from both marker one and four to the image centre and once again they should be collinear and horizontal. If they aren't the camera needs to readjusted until the desired conditions are met, if not they might indicate an offset in the camera's height and/or a rotation around the horizontal axis.

If all the above conditions are met, within given margins, the distance between the corresponding target and detected points is measured, in pixels, and if they are under a certain threshold (4 pixels) the camera is considered aligned. There is also the requirement of being able to detect all the ten markers (Six from the treadmill and four from the opposite camera) to ensure both cameras are in place during the process and to ease the positioning of the last camera considering its optical centre will be at the centre of the first camera's image after its alignment and therefore a initial position for it can be found saving some time. If the markers are not being identified correctly the color detection parameters must be reconfigured pressing the button "Modificar Detecção Marcas".

Figure 3.7 shows de described process. In the bottom right corner can be seen a representation of the treadmill's markers with a value that indicates the distance calculated



Câmara Esquerda

A alinhar câmara...

Pontos detect. : 10
Max. dist. : 33.2 pixeis

Ajuste da câmara iniciado...

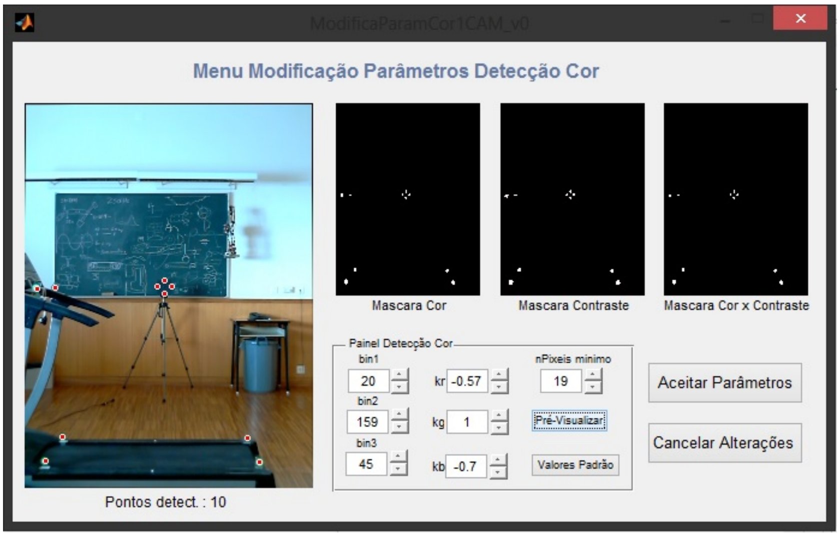
Aguarde por favor

Figure 3.7: Left camera during the alignment process

between target and detected points. To note that those distances are only a mere indication of how close the real camera pose is relative to the ideal one. It does not represent a quantitative measurement of the camera's real location and orientation nor the implications of the introduced error by having the desired data gathered with these differences present. The alignment process is also very ambiguous since a perceived correction can in fact worsen the desired goals. e.g.: moving a camera to the left and rotating it around its vertical axis might get a similar result as moving it to the right and rotating it around its vertical axis in the opposite direction, however the data acquired might not be the same for both cases.

3.4.4 Marker Detection and Adjustment

As previously discussed, each marker represents a point of interest that should be easily detectable by the cameras. To accomplish that objective, they are physically coloured in a way that contrasts against the background and other objects present. It is usually used a vivid red or green colour, however depending on the lighting conditions of the room and the background elements present, the cameras may have some difficulties distinguishing the markers from other items. When that happens the colour detection parameters can be modified so they can be adapted to the conditions present at the time by stopping the alignment process and pressing the button named "Modificar Detecção Marcas", all within the camera alignment module. A new window appears, similar to the one shown in Figure 3.8 where some parameters can be readjusted to improve marker detection.



bin1: 20
bin2: 159
bin3: 45
kr: -0.57
kg: 1
kb: -0.7
nPixeis minimo: 19
Pontos detect. : 10

Figure 3.8: Colour detection window

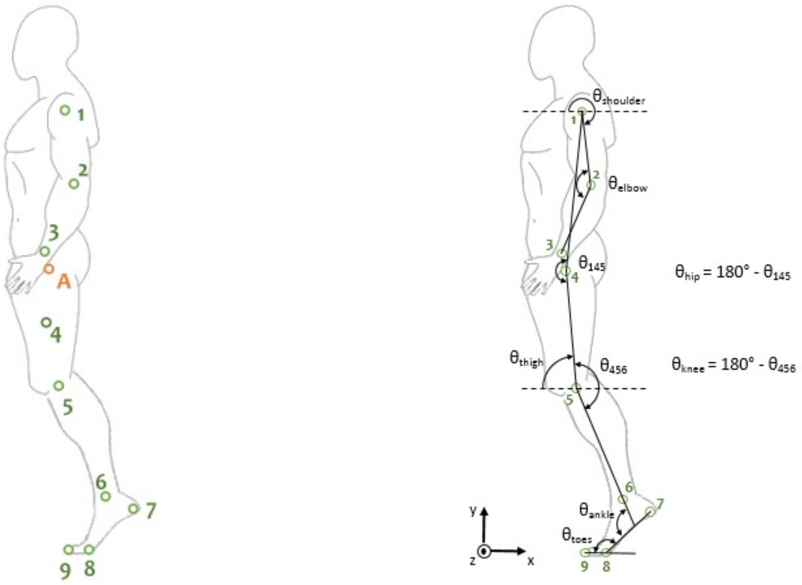
Since each camera captures a Red, Green and Blue (RGB) image each colour component can be given a weight based on the properties of the markers. e.g.: when using green markers, the green colour has an heavier weight than red or blue. By adding each resulting colour component a colour mask can be created to filter the RGB image. Each weight value can be adjusted in the menu, being kr, kg and kb the corresponding weights for red, green and blue, respectively. A contrast mask is also created using Prewitt's operator and can be tweaked using the value present on $bin2$: the minimum contrast value. $bin3$ is the threshold value to apply to the colour mask, that is, only the pixels with values above $bin3$ are considered. The colour and contrast masks are also multiplied with each other's and another mask is created as a result. The last parameters are related to blob (an interconnected region of interest) detection. $bin1$ is the minimum number of adjacent pixels in a group to be considered as a blob and its value is applied individually to all the masks presented above. $nPixeis\ minimo$ is the minimum number of pixels that will consider a blob as a valid marker.

All of these concepts may seem hard to understand, from the user point of view, nonetheless since all the impact made by each change in the values can be pre-visualized by the push of a button, or even made default, the process becomes quite simple in practice. When all the markers are being detected the new parameters can be accepted and used during the course of the remaining data acquisition. To note each camera has to be adjusted individually.

When all the markers are detected, they are still not related to any specific point of the object (i.e.: treadmill, opposite camera), they are just several groups of pixels of interest laying around the image, therefore there is the need to find their location and assign them to their corresponding object point. To do that each centre of mass is found and their location relative to each others is found (i.e.: which point is the most on the left, on top and so on). It is then compared to the relative location of the expected image points for the given configuration and finally their identification is assigned.

3.4.5 Passive Marker Placement on Patient

To be possible to acquire the patient's data, similar markers must be placed over the patient during the acquisition process. Their placement is related to the patient's joint locations that are shown in Figure 3.9a



(a) Markers Location on Patient

(b) Joint Angles

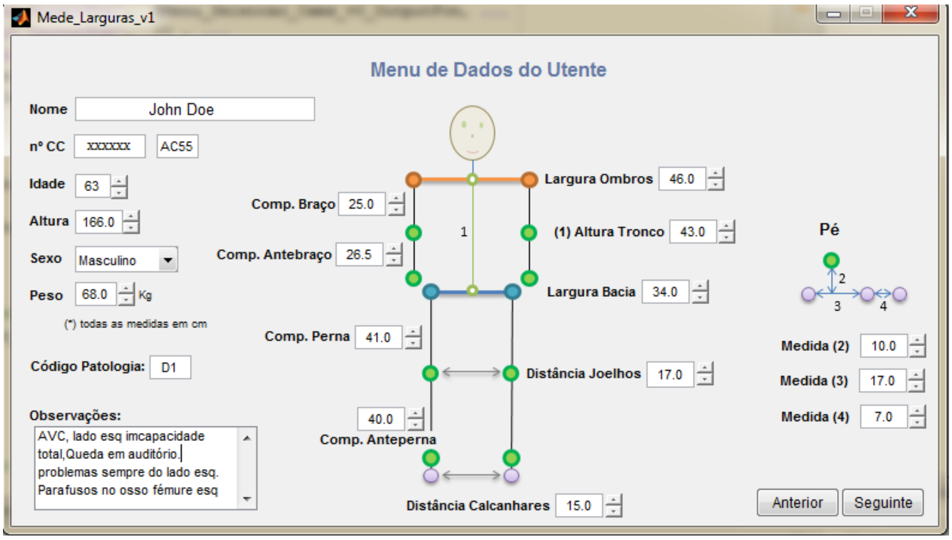
Figure 3.9: Patient markers and joint angles [10]

The data to be acquired are the patient's joint positions and the angles formed by them as a function of time while he performs several walking tests. They are strapped onto him using elastic bands. They are put over the shoulder, elbow, wrist, pelvis, knee, heel, ankle and finally the point where the foot toes fold and their tip, on both sides. Each camera captures the markers on each side of the patient. This concludes the set-up phase since all the equipment is on its intended location.

3.4.6 Patient's Data and Hip Calibration

With all the equipment being on its correct location, the second phase is initiated: the data acquisition. It can be run independently assuming all the previous steps were completed.

A window is presented to the user where he can introduce the patient's data which includes his/her name, age, citizen card number, gender, weight and several other physical characteristics that will be used to ascertain the correct location of each marker in a 3D coordinate system. To note that each camera only captures one side of the patient and therefore is considered a mono vision system. For that reason there is no depth perception and by measuring patient's physical characteristics such as shoulder, pelvis, and knee width and height and assuming he/her walks on the centre of the treadmill the depth information needed can be found. Members length is also measured and used in a similar way for the same purpose. It is also introduced the pathology code and some observations if desired. The pathology code is formed by a letter and a number associated with a set of pre-defined pathologies being $A1$ the code that represents an healthy patient. The window described above is shown in Figure



Mede_Larguras_v1

Menu de Dados do Utente

Nome: John Doe

nº CC: xxxxxx AC55

Idade: 63

Altura: 166.0

Sexo: Masculino

Peso: 68.0 Kg

(*) todas as medidas em cm

Código Patologia: D1

Observações: AVC, lado esq imcapacidade total,Queda em auditório. problemas sempre do lado esq. Parafusos no osso fémure esq

Comp. Braço: 25.0

Comp. Antebraço: 26.5

Comp. Perna: 41.0

Comp. Anteperna: 40.0

Largura Ombros: 46.0

(1) Altura Tronco: 43.0

Largura Bacia: 34.0

Distância Joelhos: 17.0

Distância Calcanhares: 15.0

Medida (2): 10.0

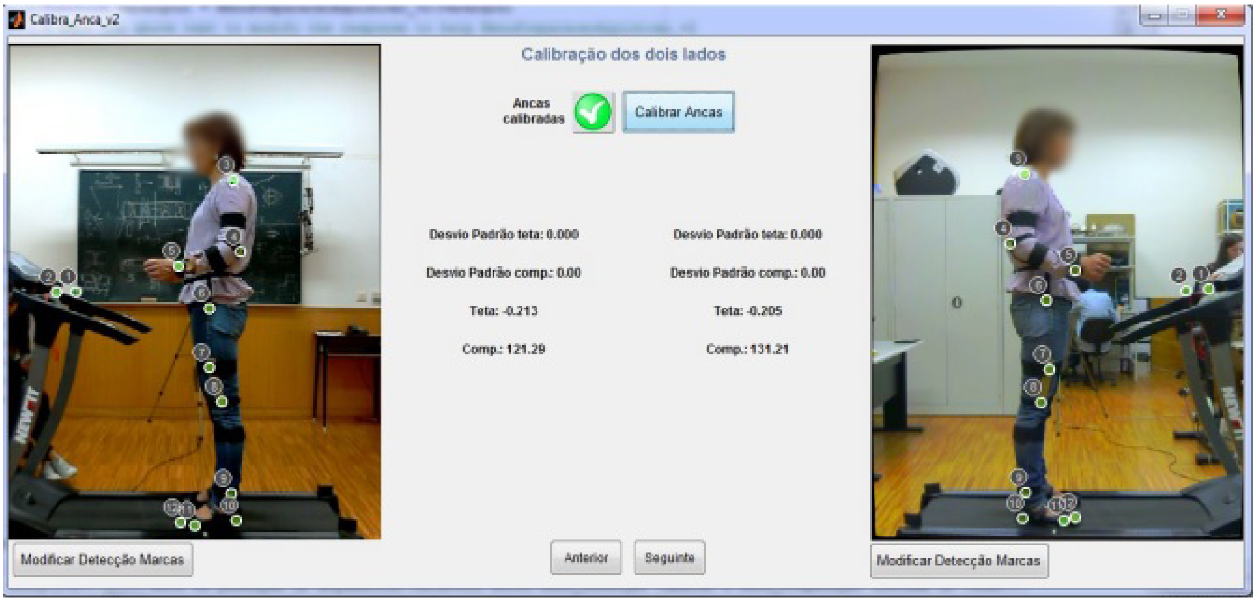
Medida (3): 17.0

Medida (4): 7.0

Figure 3.10: Patient's Data window example [10]

Due to the nature of the human gait, when a person walks the arms move forth and back. This leads to an issue since they can get in front of some markers resulting in a problem of occlusion that happens mainly in the pelvis marker. For that reason an additional marker is used and a pelvis calibration is done. The extra marker is represented in Figure 3.9a by the number 4 while the pelvis has the letter A.

The patient stands in the centre of the treadmill angling his/her forearms above the pelvis


Calibra_Anca_v2
Calibração dos dois lados
Ancas calibradas
Calibrar Ancas
Modificar Detecção Marcas
Anterior
Seguinte
Modificar Detecção Marcas

Desvio Padrão teta: 0.000

Desvio Padrão comp.: 0.00

Teta: -0.213

Comp.: 121.29

Desvio Padrão teta: 0.000

Desvio Padrão comp.: 0.00

Teta: -0.205

Comp.: 131.21

Figure 3.11: Pelvis calibration Window [10]

as in Figure 3.11 while the calibration process occurs. Marker 4 acts like an intermediate point and in conjunction with marker 5 their relation is determined and used to infer the pelvis position. After the calibration process is done the marker A is removed and no longer used. It is also relevant to mention that from now on the treadmill markers that were placed on its base (numbers 2, 3, 5 and 6 from Figure 3.3) are also removed due to patients being constantly kicking them during the walking tests. The numbering presented in Figure 3.11 refers to the detected markers and shall not be confused with the numbering from Figure 3.9a. If the marker are not detected the colour detection can be re-adjusted in a similar way as described in 3.4.4. When the calibration process is finished for both cameras the tests can start.

3.4.7 Data Acquisition Module

The Data Acquisition Module is responsible to record the videos of the patient walking on the treadmill and associate them with the patient's info as well as the information of tests performed. They can be done with or without force sensors provided by the instrumented shoes and several velocities can be chosen. For a typical healthy person, the tests are set to run on 5 distinct velocities, starting with the slowest and between sets of two minutes to ensure it is gathered enough data to generate a gait pattern. There is an option to do the increments automatically or manually, since not all the patients are able to walk at the highest velocities due to their pathologies. Before the patient starts the tests he walks on the treadmill for six or more minutes so he can be accustomed to it[22].

To ensure synchronization between both cameras a pair of white leds are present on the front of the treadmill and are activated during this stage, blinking at a constant frequency. When all the tests are done all the information is gathered in a file and associated with the patient's videos and information.

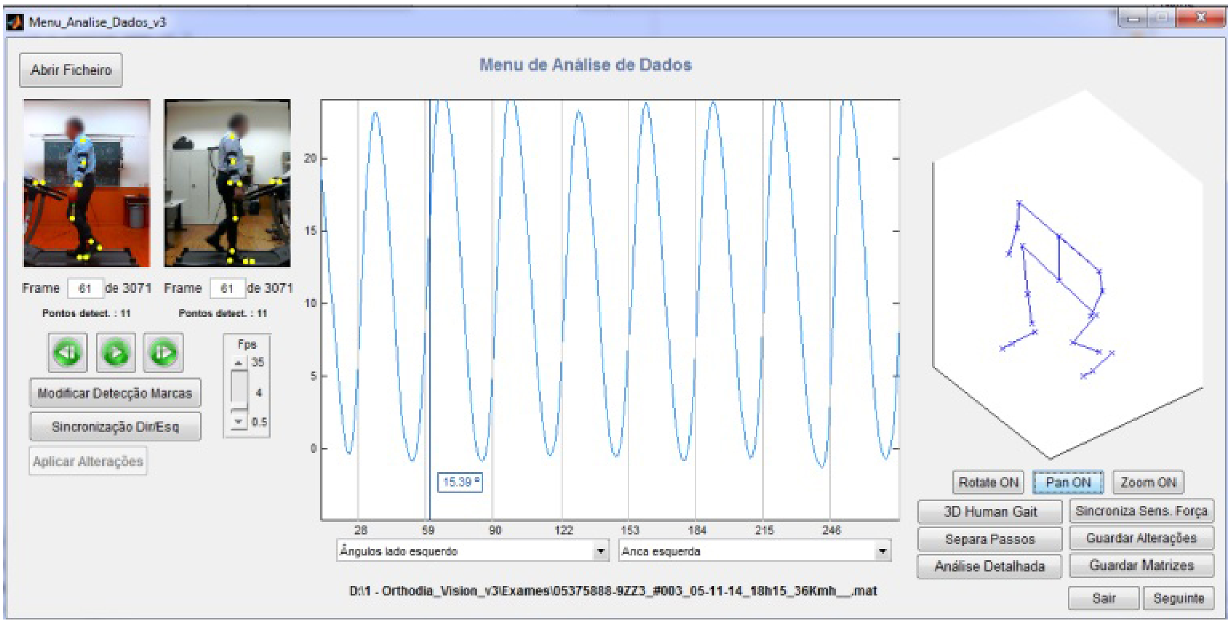
3.4.8 Data Analysis Module

The Data Analysis Module gets the information form the file generated by the Data Acquisition Module and its videos and loads them so they can be processed. With this, begins the third phase: the data analysis. A new window as shown in Figure 3.12 is presented where the data can be adjusted and pre-visualized before being processed. Loading the data file may take up to about fifteen minutes, depending on the hardware used. During the loading process the videos are analysed so the markers can be detected using a similar process as described in 3.4.4 and associated with the human joints. Their position is determined using all the information from the cameras and patient's physical data and the angles are extracted. It is important to keep in mind this really is a very simplistic description of what is done during the whole process. Figure 3.9b shows how each joint angle is measured.

In the Data Analysis Window, the frames from both cameras as well the joint angle are plotted and can be seen as a function of time. Each joint can be selected from a drop down menu and the respective information is shown. They are also represented in 3D by a figure that mimics the human body form and has its joint angles set by the data acquired by the tests, also represented as a function of time.

Some adjustments need to be made to ensure data accuracy, with one of them being the video synchronization. In a perfect world both cameras should start the capture process at the same time and at a constant frame rate, however in a real world that does not happen, meaning that the first frame from the left camera does not have the same time mark as the first frame from the right camera. In fact they can be desynchronized by a few seconds which is not desirable at all. Using the leds as a reference both video frames can be synchronized in a separate window by pressing the button "Sincronização Dir/Esq". The synchronization window has several adjustable parameters as well as both video frames that can be used to validate they are being corrected by visual confirmation. After the required modifications are made, they are applied and ready to be processed to create the Patient's Gait Pattern.

Since not all the markers are placed on the same plane and due to the projective geometry, the angle formed between them might not be perceived as the real angle. Therefore some


Menu_Analise_Dados_v3

Menu de Análise de Dados

Frame 61 de 3071 Frame 61 de 3071

Pontos detect. : 11 Pontos detect. : 11

Ângulos lado esquerdo

Anca esquerda

D:\1 - Orthodia_Vision_v3\Exames\05375888-9ZZ3_#003_05-11-14_18h15_36Kmh__.mat

Figure 3.12: Data Analysis window [10]

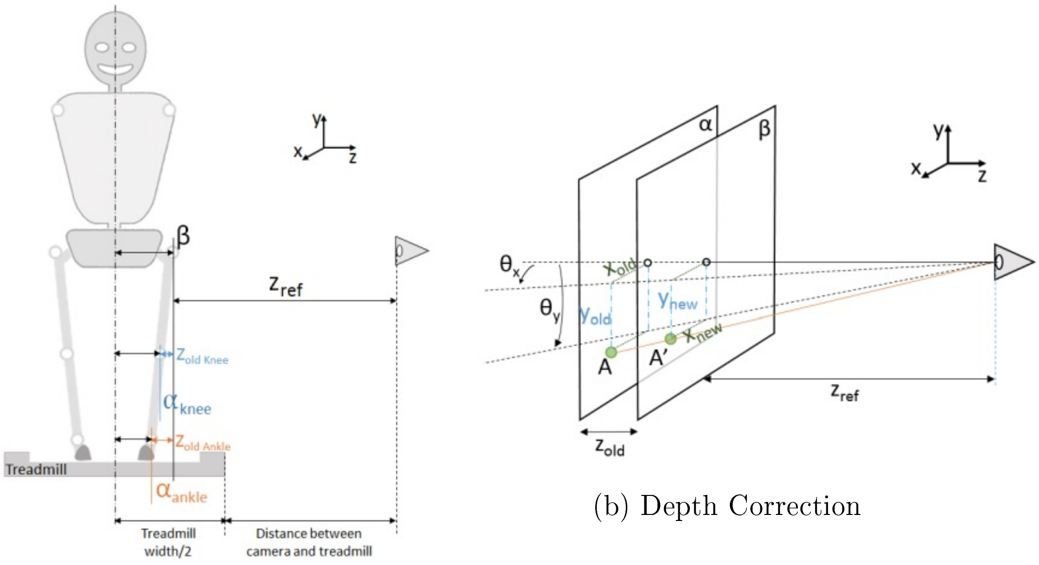
depth corrections are also made [14] that projects every point of interest on the same plane and only after, the angles formed between them are measured. Figure 3.13a illustrates a representation of the several planes in which the user markers are contained and the reference plane β where they will be considered to be for the angle measurement. The new coordinate points are calculated using Equation 3.1 and a graphical illustration is presented in Figure 3.13b where the old coordinates from point A contained plane α are transformed into point A' contained on the reference plane β. The angles between all the markers are calculated only after these depth corrections are applied using equation 3.1.

$$\begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix} = \frac{z_{ref}}{z_{ref} + z_{old}} \begin{bmatrix} x_{old} \\ y_{old} \end{bmatrix} \tag{3.1}$$

The Data Analysis Window also presents some extra options like synchronization with force sensors, data exportation to use with other applications as well some buttons that allow to save and process the newly adjusted data. When all the desired work is done the user can go to the next step, the Pattern Processing Module.

3.4.9 Gait Pattern Processing and Comparison Modules

The Gait Pattern Processing Module is responsible for analysing the data obtained from the Data Analysis Module and generate the patient's gait patterns. A new window is presented, similar to Figure 3.14 which shows a graphical representation of each joint angle data, more specifically the mean joint angle profile and their respective standard deviation (green) after

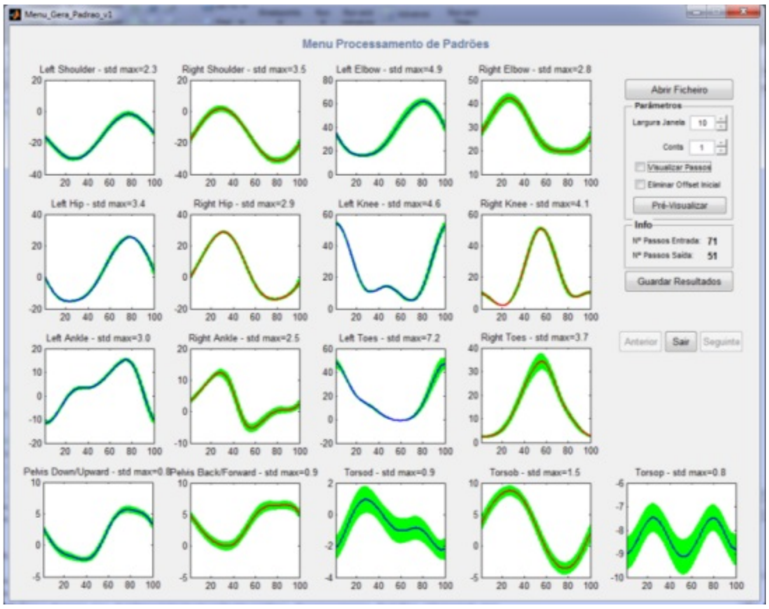


(a) Reference Plane

(b) Depth Correction

Figure 3.13: Depth Correction illustrations [7]

having all the stride grouped. This window can also be started independently and can load any data file previously saved.



Parâmetros

Largura Janela 10

Conta 1

□ Visualizar Passos

□ Eliminar Offset Inicial

Info

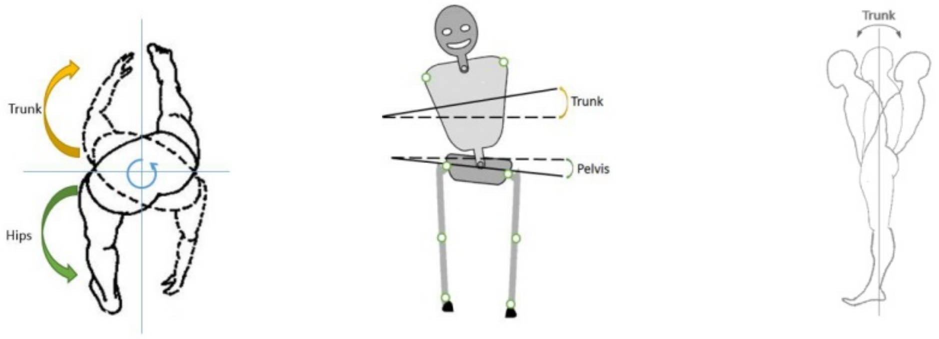
Nº Passos Entrada: 71

Nº Passos Saída: 51

Figure 3.14: Gait Pattern Generator Window [10]

By combining the markers positions from each camera is also possible to evaluate some other angles, namely the backward/forward, posterior/anterior and downward/upward pelvis and trunk angles as shown in Figure 3.15a, 3.15b and 3.15c

The last module developed is the Gait Pattern Comparison Module where a maximum total of 10 patient tests can be loaded and compared with each other. The data is grouped



(a) Backward/Forward (b) Downward/Upward (c) Posterior/Anterior

Figure 3.15: Angles formed by crossing information from both cameras [7]

and can be chosen by the user using drop menus. It is very useful and not only allows an overview of each patient's gait pattern properties but also allows for a quick assessment over the possible gait pathologies present just based on the patient's data and its comparison against patterns from other patients. Figure 3.16 shows an example window of the referred module.

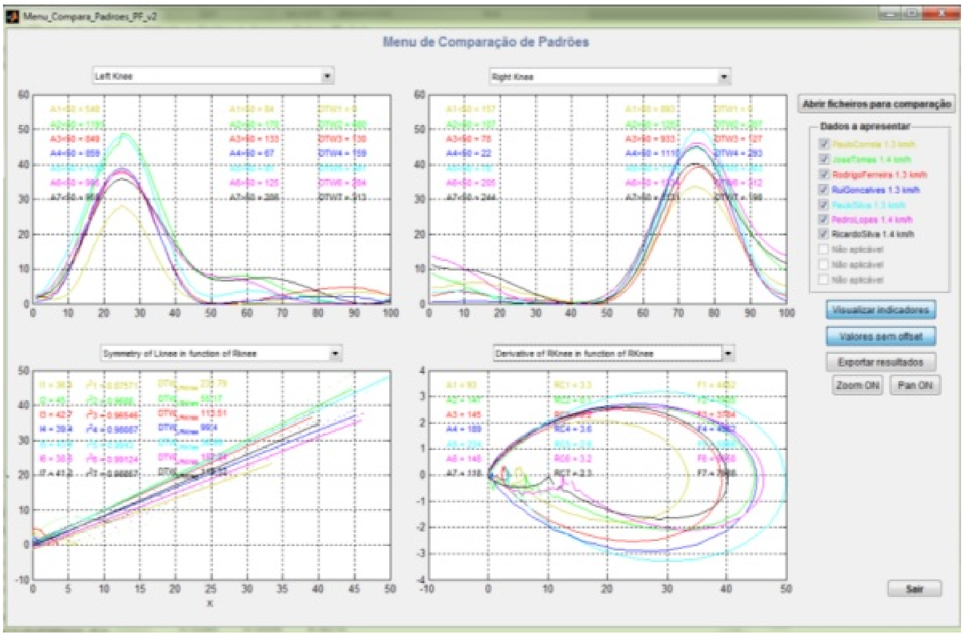


Figure 3.16: Gait Comparison Module Window [10]

Chapter 4

Simulation

One of the main objectives of the present thesis is to improve the Camera Alignment Module, a process that requires a considerable amount of time and may introduce some error in the desired results due to the misalignment of the cameras. The alignment algorithm itself relied on several verification conditions that were very prone to induce errors on the alignment process as well in the patient's gait data acquired later on, e.g.: if one of the markers 2 or 3 of the treadmill are just slightly higher or lower in relation to each other, the line that is drawn between them will not be parallel to the ground and the perpendicular line created from its mean point will not be a straight vertical line, thus giving the user the instruction that a rotation should be applied to the camera around its optical axis. If we combine all the possible factors that may result in erroneous instructions for the user and the fact there is no way to really confirm the true pose of each camera in relation to the treadmill since the main verification condition is based on pixel distances between target and detected points, a new method should be applied that not only eases the camera alignment process but also gives some measurable information that can be used to relate their poses to the treadmill.

To make sure that every piece of equipment is in its correct place a very simplistic simulation of the scenario was created using MATLAB. It was used to test not only the new method applied for the alignment module but also to ascertain the outcome of not having a camera with the desired pose by creating and placing several elements over the treadmill that simulated human limbs, called objects. Their projections were calculated onto the 2D image plane and the angles measured and compared with their respective true angle. There were considered several positions and orientations both for the cameras and the objects created.

4.1 Simulation Setup

To recreate the conditions needed to run the simulation a base referential frame which will serve as the room where all the equipment will be placed as described in Section 3.4 was created and will be referred as the World Frame. It basically consists of a 3D space whose dimensions are 8 × 8 × 3 meters with its origin placed onto a corner. The referential frame of the treadmill is located at the centre of the room and the point coordinates of its markers are referenced to it based on the real known measures presented in Figure 3.3. The camera positions were also set accordingly to Figure 3.3. The 3D room representation created in MATLAB is presented on Figure 4.1.

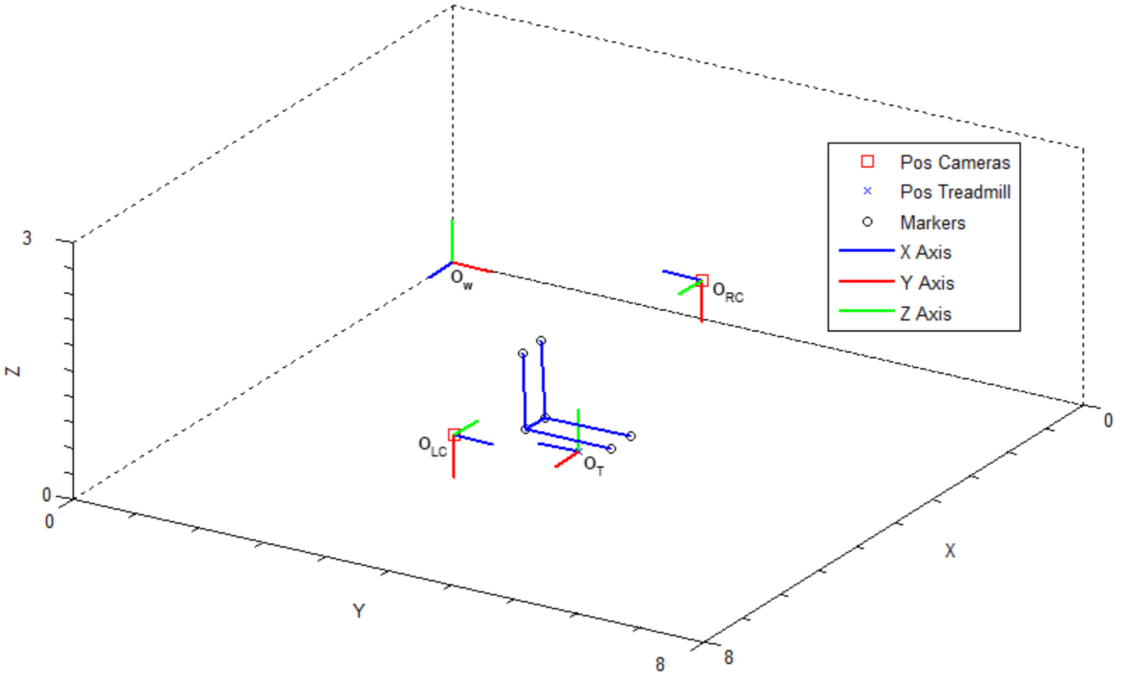


Figure 4.1: 3D Simulation using Matlab

Some markers of the treadmill are united so the user can have a sense of its orientation. Both camera frame's origins are denoted by the square mark, although they are independent from each others. The orientation of each object is denoted by each coloured axis, being blue, red and green for x, y and z respectively. Notice that all the referential frames are perpendicular to each others. All the object's coordinates present in this simulation can be mapped onto another referential frame, depending on the task requirements to be simulated. The 3D representation also act as a visual confirmation method that can be used to validate some of the results.

The camera axes, by convention, have the optical axis (z) pointing in the direction of the object, away from the camera itself, while the x and y axes don't have any requirement apart from being perpendicular amongst them and the z axis [25]. For this simulation was adopted the model suggested by MATLAB [20] that uses the x axis pointing to the left side of the camera and the y axis pointing down completing the system (Figure 4.2)

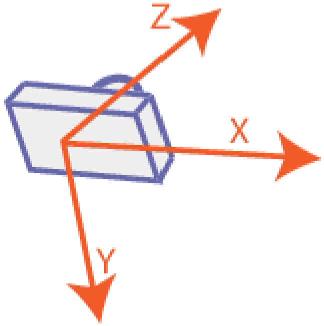


Figure 4.2: Camera Coordinate System [20]

4.2 Ideal Camera Intrinsics Matrix

As described in Section 3.4.3, the target points were calculated by an application that used the camera focal distance (4 mm) provided by its manufacturer. This obviously needs to be improved by using the focal distance obtained from the camera calibration. The intrinsics matrices for the left and right cameras are presented in (4.1), (4.2) respectively.

$$\begin{bmatrix} 805.2059 & -2.2178 & 247.2042 \\ 0 & 808.7742 & 300.8397 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

$$\begin{bmatrix} 819,2952 & -0,0351 & 232,5143 \\ 0 & 818,4780 & 332,3348 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

In Section 2.2.1 we saw that $f_{x,y} = \frac{F}{S_{x,y}}$, where F is the camera's focal distance in m, $S_{x,y}$ is the sensor pixel size also in m and $f_{x,y}$ is the focal distance in pixels, or in other words, the value presented in the Intrinsics Matrix. Ideally the cameras should have square pixels, however considering the camera's manufacturer only provided the sensor's pixel size the following calculations were made:

$$f_x = \frac{F}{S_x} = 805.2059 \Leftrightarrow F \approx 0.0023 \tag{4.3}$$

$$f_y = \frac{F}{S_y} = 808.7742 \Leftrightarrow F \approx 0.0023 \tag{4.4}$$

which gives approximately 2.3 mm for the left camera's focal distance. Now, assuming that f is unknown and solving for it using the values we just obtained.

$$f_x = \frac{F}{S_x} = \frac{0.0023}{2.8 \times 10^{-6}} \approx 821,4286 \tag{4.5}$$

Applying the same calculations to the right camera's intrinsics matrix it is also obtained the value of 2.3 mm for its focal distance F therefore concluding that the value obtained in Equation 4.5 is more appropriated to use in future calculations not only because it was obtained using parameters from the camera calibration but also because now f_x and f_y have the same value which is desired and expected on an ideal camera. Having in consideration than an ideal camera also does not show any type of lens distortion and that it has its sensor centred the camera Intrinsics Matrix becomes:

$$\begin{bmatrix} 821,4286 & 0 & 240 \\ 0 & 821,4286 & 320 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

From now on, both cameras will be considered as being physically identical, that is, they are considered to be ideal. It's also worth mentioning that the objective of this simulation is not to recreate exactly all the conditions seen in a real world scenario but rather having the possibility of using a simple platform that allows to guarantee certain conditions that can't be assured in a real world scenario such as knowing the exact location of an object or even the exact value of an angle formed by a set of points with unknown 3D coordinates, as seen by a camera.

4.3 3D Treadmill's Marker Projections onto 2D Image Plane

Using rigid body transformations (see Section 2.2) each marker from the treadmill is mapped onto each camera's reference frame. All referential frames have a known position and orientation and the desired mapping can be achieved by several ways. In order to have an independent frame from all the objects that are part of the equipment (cameras and treadmill) and for debug reasons, the world frame was used to map all the objects first and only then, further transformations were applied. To obtain the coordinates of the markers of the treadmill in the World's reference frame Equation 2.9 was used becoming:

$$\begin{bmatrix} {}^{W}P \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} {}^{W}_{T}R & {}^{W}O_{T} \\ 0^{T} & 1 \end{bmatrix}}_{{}^{W}_{T}T} \begin{bmatrix} {}^{T}P \\ 1 \end{bmatrix} \tag{4.7}$$

where ${}^{T}P$ is a point from the treadmill's reference frame, ${}^{W}P$ is a point in World's coordinate frame and ${}^{W}_{T}R$ is:

$${}^{W}_{T}R = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

Each camera position and orientation in relation to the World frame is also known and applying Equation A each marker can be mapped from the World frame into each camera frame using the inverse transformation matrix ${}^{C}_{W}T =^{W}_{C} T^{-1}$:

$$\begin{bmatrix} {}^{C}P \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} ({}^{W}_{C}R)^{T} & -{}^{W}_{C}R^{-1} \cdot^{W} O_{C} \\ 0^{T} & 1 \end{bmatrix}}_{{}^{W}_{C}T^{-1}} \begin{bmatrix} {}^{W}P \\ 1 \end{bmatrix} \tag{4.9}$$

where ${}^{W}_{C}R^{T}$ is presented in (4.10) and (4.11) for the left and right cameras, respectively.

$${}^{W}_{LC}R^{T} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 1 \end{bmatrix} \tag{4.10}$$

$${}^{W}_{RC}R^{T} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.11}$$

Now that each marker is mapped into each camera frame the 2D image points can be found by pre-multiplying each point by each camera's intrinsics Matrix as previously showed in Equation 2.23. The resulting projection points are shown in Figure 4.3. The top markers are displayed in red to avoid any possible confusion regarding the orientation and the star shaped mark represents the image centre. Both new projected points (circles) and those that were being previously used on the project (black boxes) are also shown on the image from the left camera. A noticeable difference can be seen and even considering that the simulated projection is not using the exact intrinsic camera matrix it is an observation to be taken into account.

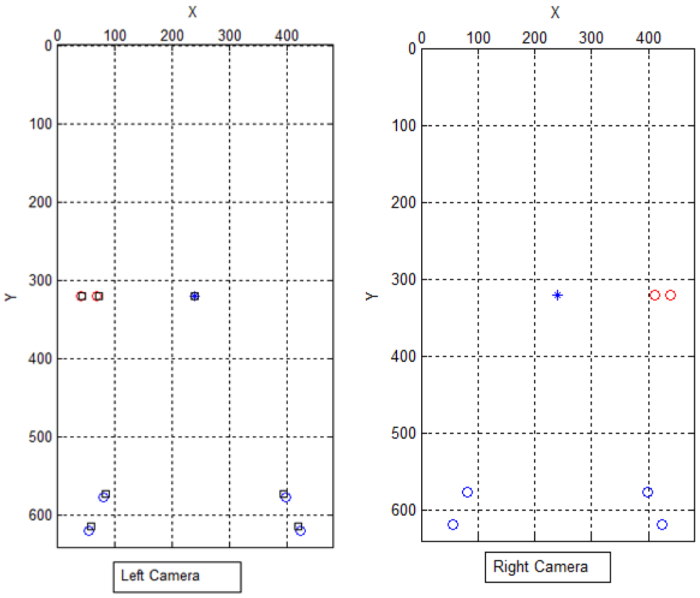


Figure 4.3: Treadmill's Markers projections from each camera and previously used projections (squares on left camera).

From now on, every result presented from the simulation will be shown from the left camera point of view only for the reason that both images are symmetrical and there is no need to keep presenting similar results.

4.4 Camera's Pose Estimation

In order to improve the camera alignment process a new method needs to be used that not only must be faster but also more reliable. To accomplish those goals a new approach can be used that allows to estimate the position and orientation of the cameras based on the known 3D object points, their respective image projections and the camera's intrinsic parameters. This point correspondence problem is known as the Perspective-n-Point (PnP) problem [16] and can be solved using various approaches such as the Gauss-Newton [12][36] or the Levenberg-Marquardt methods [12][17][19].

For this project it was used a function available in the OpenCV library called "SolvePnP" which implements an algorithm based on the Levenberg-Marquardt optimization method that finds such a pose that minimizes reprojection error [27]. The function inputs are the 3D object points, their corresponding 2D image points, the intrinsics camera matrix and lens distortion coefficients. The object points are the 3D coordinates of each marker of the treadmill given in the treadmill's reference frame, the image points are their corresponding projections calculated as shown in Section 4.3 and the intrinsics camera matrix is the one presented in (4.6). The lens distortion coefficients are assumed to be zero since the simulation considers the camera to be ideal. As optional inputs the function also admits an estimated guess for the position and orientation of the object in relation to the camera coordinate frame, which are used as initial values in the optimization algorithm. The estimated guess is therefore the rotation and translation vectors from the transformation that relates the treadmill to the ideal camera coordinate frame (${}^{C_I}_{T}T$). The outputs returned are the rotation and translation vectors from the transformation that relates the treadmill and the estimated camera coordinate frame (${}^{C_E}_{T}T$). Given that both ${}^{C_E}_{T}T$ and ${}^{C_I}_{T}T$ share the treadmill's referential frame and are known, the transformation between the estimated and ideal camera can be calculated by Equation 4.12

$$ {}^{C_I}_{C_E}T = {}^{C_I}_{T}T \cdot ({}^{C_E}_{T}T)^{-1} \tag{4.12} $$

The rotation matrix and the translation vector from ${}^{C_I}_{C_E}T$ can be used to extract the information needed to move and rotate the camera from which the estimation was computed. The Euler angles can also be obtained from the rotation matrix and given the physical setup of the equipment, degenerate solutions will never occur [31]. The extracted Euler angles and the translation vector will be used to display to the user how and where he should move and rotate each camera in order to obtain the desired pose.

4.5 Simulation Tests and Results

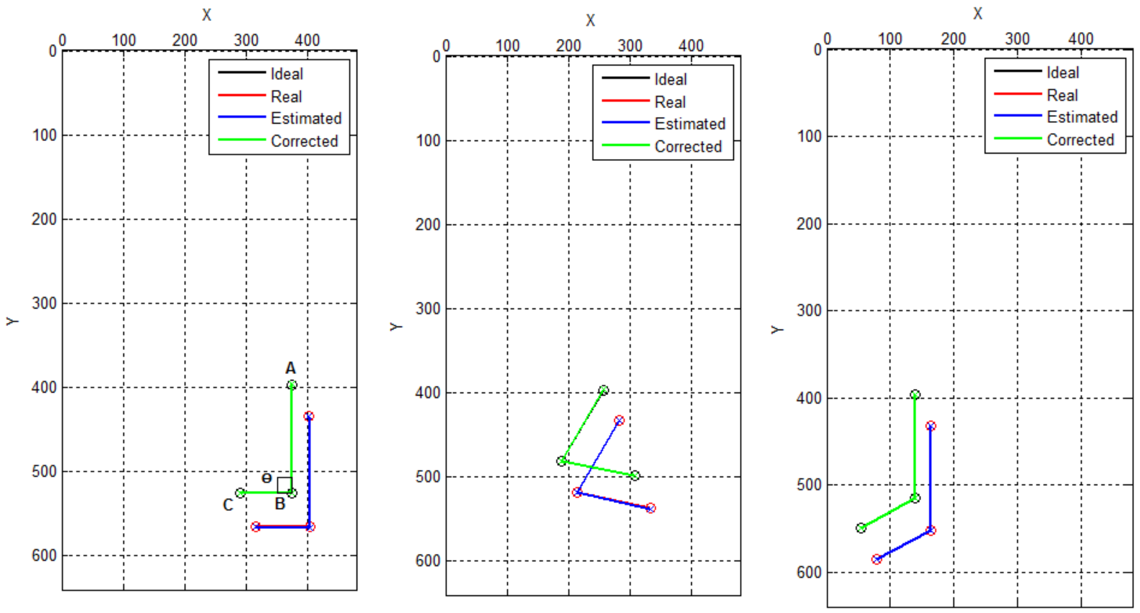
The new alignment method based on the PnP problem should improve drastically the time needed for the alignment process for the reason that it gives precise indications to the user on how to align each camera. Considering that in a real world scenario it is impossible to do a perfect alignment, even after the camera is considered by the system as being aligned, a small error on its pose is still present. This present section intends to evaluate the perception of an angle formed by a set of known 3D points, by the camera on various poses, in order to determine if a misalignment within the margins accepted by the system can have negative impact on the results. A brief assessment on the pose estimation results is also done by comparing them to the real camera pose which is known as it was set in this simulation.

As already mentioned, the camera in an ideal pose is considered to be on the desired location and orientation in respect to the world frame as described in section 3.1. For simplicity reasons, the ideal camera reference frame will be referred as the ideal camera. Following the same nomenclature, the real camera frame is created from the ideal camera by translating it by a given distance and rotating it around its axes by a set of Euler angles. The resulting pose is analogous to the real world camera with the difference that in this simulation its pose is known since it was defined by the user while the real world one is not. The markers from the treadmill were then projected in the real camera image plane and then used to compute an estimated pose for the real camera as if its pose was not known. From the results obtained, is created another referential frame that will be referred as the estimated camera. Finally by using the matrix transformation that relates the estimated camera and the ideal one, any known 3D world coordinates can be mapped from the estimated camera into the ideal one, however since the estimation process is not perfect, the resulting pose will not be the same as the ideal one by a small margin and for that reason, this will be called the corrected camera frame and can be used to evaluate the pose estimation results.

In order to perform the desired tests various set of points were created over the treadmill area. Their 3D coordinates are known and they act as an object of interest to be observed, as a limb from a patient. All the points from an object are contained in a plane parallel to the sagittal plane, therefore being coplanar and avoiding possible errors due to their positioning. Each object has three points that form two segments and an articulation whose projected angle is measured and compared against the real 3D angle and all the other projections. Let's consider the following example where an object created from three points A, B an C form an angle θ measured from the known 3D point coordinates as seen in Figure 4.4a.

Calculating the projection points of an object from the 3D world coordinates to the ideal camera frame results in the projection displayed in black in Figure 4.4a. Applying a translation of 30 mm to the x axis of the ideal camera position and a rotation of $+2.5°$ and -2.5° around the x and y axes [1], respectively, a new pose is created that simulates a misaligned camera, that is, the real camera. The respective projections of the 3D object points are displayed in red where an obvious shift on the position over the image plane can be seen. The resulting computation for the estimated pose, created from the projected points of the markers of the treadmill and their 3D location as seen by the real camera is also found giving origin to the estimated camera. Once again, projecting the points A, B and C into the reference frame of the estimated camera we get the projection shown in blue. To note that if the estimation provides a good approximation for the real camera, both projections should be identical. Finally using the pose from the estimated camera the corrected projection can

[1] values arbitrarily chosen



(a) Object 1: 90.000°

(b) Object 2: 59.470°

(c) Object 3: 111.801°

Figure 4.4: Projections from simulated objects

be found. It is shown in green and as expected it also overlaps the ideal camera projections.

Table 4.1: Absolute values of the angle differences between Figure 4.4 projections and the actual known angle

Obj.	**3D Angle**	**|Ideal|**	**|Real|**	**|Corrected|**	**|Real-Corrected|**
1	90.000°	0°	1.127°	0.010°	1.118°
2	59.470°	0°	0.674°	0.006°	0.668°
3	111.801°	0°	0.005°	0.001°	0.004°

In Table 4.1 are shown the real angles θ from Figures 4.4a, 4.4b and 4.4c and the absolute values of the differences between the known angle from each 3D object and the respective angles measured from the image projections of each camera frame being considered.

The presented example does not have sufficient data to get any definite conclusions, however when correcting the real camera pose using the computed estimation, the differences of the projected angles between the corrected and ideal camera frames are minimal, which means the estimated pose obtained is a good approximation to the real camera pose. Table 4.2 shows the absolute values of the differences between the estimated and real camera poses. It is also obvious that if a camera could be perfectly aligned there would be no perceived error. Another curious observation can be noticed when comparing the angle differences between the real and ideal cameras since they suggest that for a single pose, those differences depend

Table 4.2: Absolute values of rotation and translation differences between Real and Estimated cameras axes

$\lvert R_x \rvert$	$\lvert R_y \rvert$	$\lvert R_z \rvert$	$\lvert t_x \rvert$	$\lvert t_y \rvert$	$\lvert t_z \rvert$
0.024°	0.022°	0.115°	0.000 mm	0.000 mm	0.000 mm

not only on the camera rotations applied but also on the object location and shape.

In order to assess on how the camera pose might affect the perceived angle in a generalized way, a new approach was employed: using the same objects shown in the previous example and having the camera initially set on the ideal pose, the projections for each object were computed and the camera was then moved along each of its axis, individually, by a fixed distance, so at each new iteration, the projection of each object could be calculated and compared. The boundaries were set in such a way that didn't allow the markers from the treadmill to get out of the image plane. All the angles resulting from the projections had the same value for each object, meaning that moving around the camera in such a way that allows the image plane and the object plane to be parallel, does not have any impact on the angle perceived by the real camera.

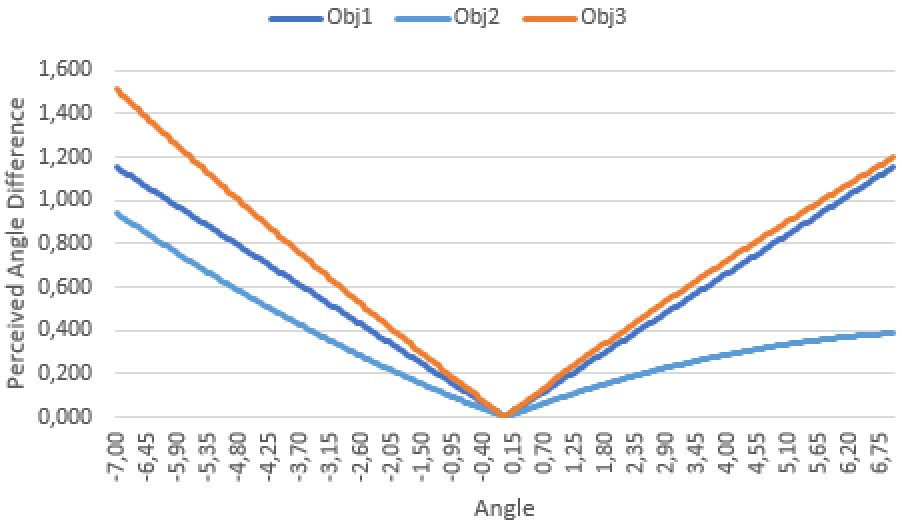


Figure 4.5: X axis rotation: Absolute values of the difference between the actual angle and projected angle on the rotated camera, for each object (in degrees)

The same experiment was done to evaluate the impact of having each camera axis rotated individually whilst having it placed on the ideal position. The rotation boundaries were calculated so the markers from the treadmill weren't allowed to get out of the image plane. The boundaries found are: $-1.310° < x < 21.290°$; $-2.700° < y < 3.600°$ and $-7.390° < z < 7.390°$ and correspond to the expected limits in a real world scenario. Nevertheless, this

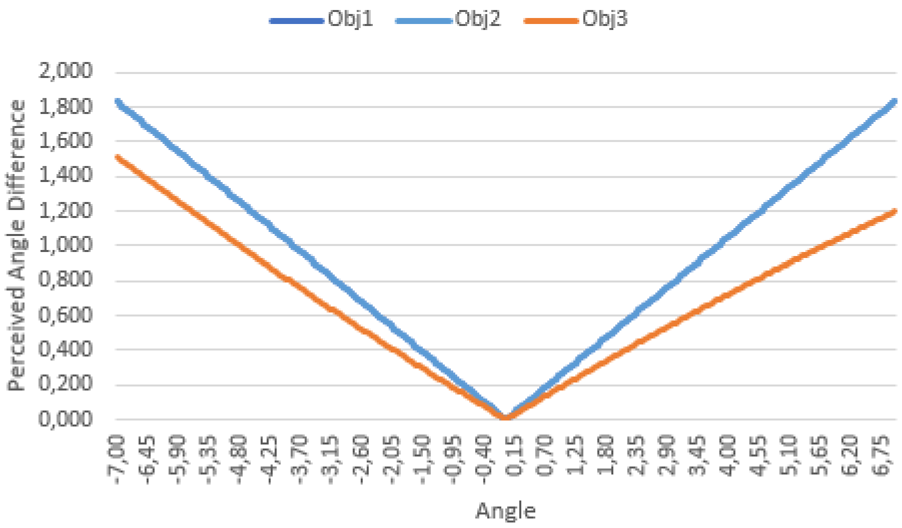


Figure 4.6: Y axis rotation: Absolute values of the difference between the actual angle and projected angle on the rotated camera, for each object (in degrees)

simulation can compute the desired projection outside those boundaries as if the image had an infinite width and height, covering all the image plane. Separated rotations were applied around each camera axis from $-8°$ to $8°$ and the projected angles from objects measured.

Figures 4.5 and 4.6 display the angle differences between the true angle of each object and their respective projections that result from rotating the camera around the x and y axis. It can be seen that their difference increases as a function of the rotation applied, in either directions. It is obvious that the contribution from a rotation does cause a noticeable impact on the angle as seen from the real camera point of view and when compared to a translation. Using the estimated pose to apply a transformation on the real camera correcting its orientation into a corrected pose and registering the module of the differences from the resulting angles from both camera projections for each object gives origin to Figures 4.7 and 4.8, and Tables 4.3 and 4.4.

No rotation around the z axis (optical axis) is shown since the angle differences between the rotated camera and the ideal one are always zero, or in other words, a rotation around the camera optical axis does not have any impact on the perceived angle by the real camera.

Table 4.3: X axis rotation: Mean ($\bar{x}$) and standard deviation (σ) of the absolute value of the difference between the actual and corrected camera angles, for each object (in degrees)

	Obj. 1	**Obj. 2**	**Obj. 3**
$\bar{x}$	0.002	0.001	0.003
σ	0.001	0.001	0.002

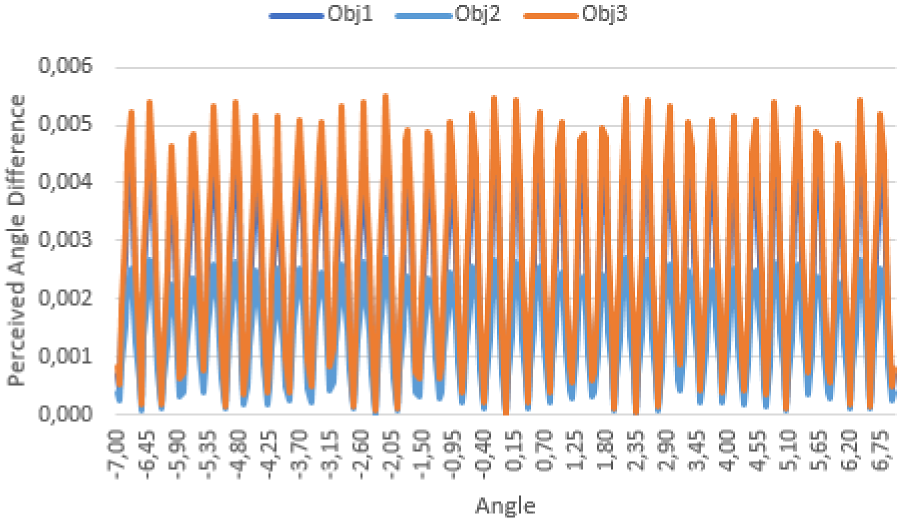


Figure 4.7: X axis rotation: Absolute values of the difference between the actual angle and projected angle on the corrected camera, for each object (in degrees)

Table 4.4: Y axis rotation: Mean ($\bar{y}$) and standard deviation (σ) of the absolute value of the difference between the actual and corrected camera angles, for each object (in degrees)

	Obj. 1	**Obj. 2**	**Obj. 3**
$\bar{y}$	0.002	0.004	0.002
σ	0.001	0.002	0.002

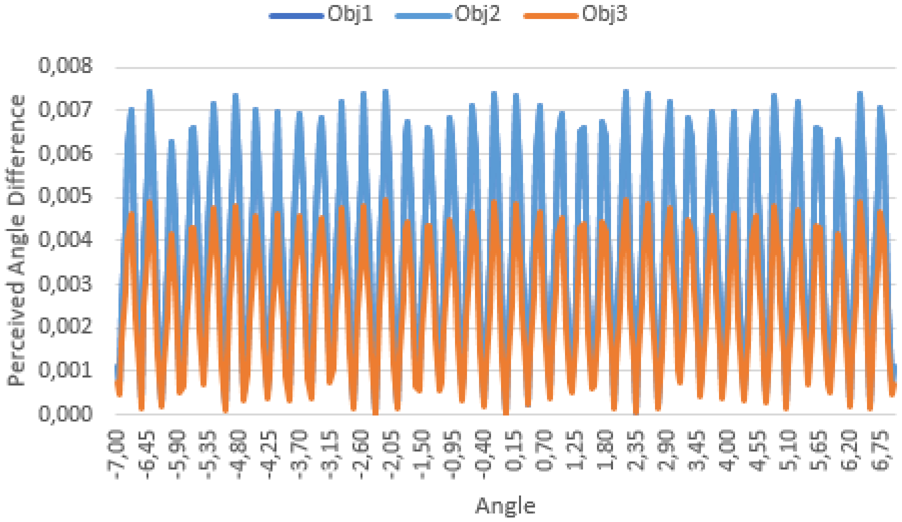


Figure 4.8: Y axis rotation: Absolute values of the difference between the actual angle and projected angle of the corrected camera, for each object (in degrees)

Considering that all the object are placed in different locations along a given plane and are composed of several distinct angles, the values from Tables 4.3 and 4.4 indicate that the corrections applied from the estimated camera pose provide a very small error value when compared to the values perceived by the real camera that does not have any kind of correction.

Another factor to have in consideration is that during the alignment phase the user will place the real camera as close as he can to the position of the ideal camera which means that if it might close enough so the angles perceived will be very similar to those obtained after the corrections are applied using the pose estimation method already introduced.

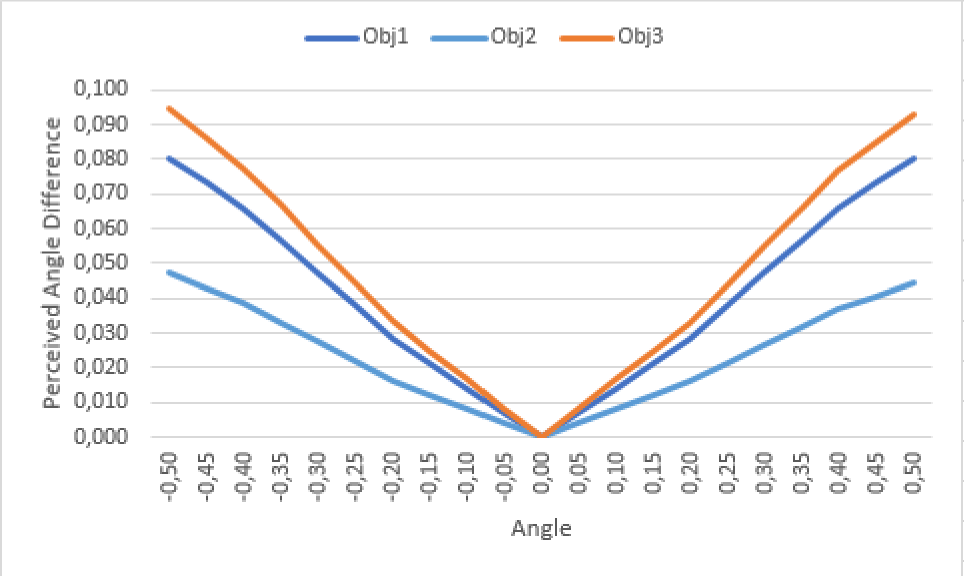


Figure 4.9: X axis rotation: Difference between the angle as seen by the real and corrected camera for each object (in degrees)

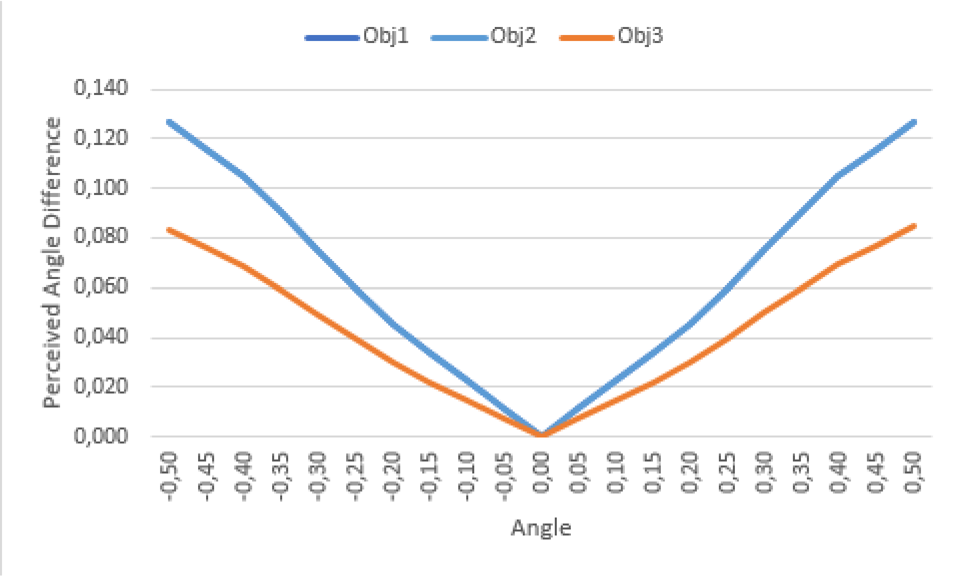


Figure 4.10: Y axis rotation: Difference between the angle as seen by the real and corrected camera for each object (in degrees). To note that Obj1 and Obj2 are superimposed.

Figures 4.9 and 4.10 shows the mentioned case. The angle difference from the real and corrected cameras is very low when the real and ideal camera poses are close to each other. Depending on the requirements of the alignment module, the real camera might have to be aligned in such a way that the resulting axis angles are so close to the desired pose that any correction that could be applied results in an angle value very close to the one that the real camera gets without any correction. e.g.: from Figure 4.9, if the real camera is misaligned by $-0.12°$ around the x axis, the angle difference between the camera with and without the correction applied would be only 0.010° for the object 2.

On a final note, there were only considered individual translations and rotations for each camera axis. In order to be able to obtain more precise conclusions more combinations needed to be studied. It would imply to mix not only individual rotations and translations but also various objects in several positions, orientations and forms which results in an infinite number of possible combinations. The main goal of this section was to determine what should be prioritized during the camera alignment. It can be concluded that an offset in the camera orientation introduces more error on the perceived angle than an individual translation, which in this simulation did not have any influence on the results. Such fact can be explained by imagining how the image plane moves when a rotation is applied to the camera. When the camera starts rotating the image plane is no longer parallel with the object's plane and their points start having different depths in relation to the image plane, depending on their original position. A depth correction needs to be applied in order to obtain the correct angle formed by those points. In the case of pure translations both camera and object planes stay parallel, having no impact on the perceived angles.

Chapter 5

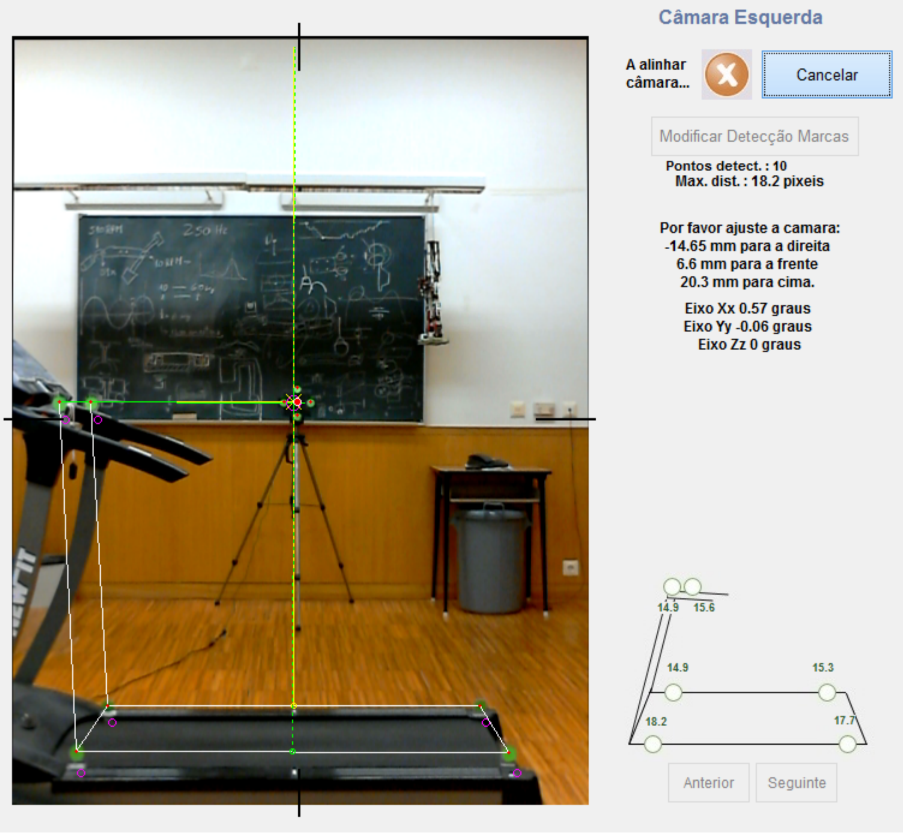
Experimental Results

This chapter describes the implementation of a new method for the camera alignment module of the gait acquisition and analysis system, based on the pose estimation technique for the cameras, employed in the simulation (chapter 4), as well as the tests conducted to evaluate its performance. It also concludes, based on the results from the estimated poses for the aligned cameras, if potential corrections to be applied on the angles perceived by the cameras are justified. All the images obtained by the cameras have their distortion corrected and the intrinsics camera parameters used are those obtained from the camera calibrations. The target points are now calculated using each camera's parameters.

5.1 Real World Implementation

As discussed in Section 3.4.3, the camera alignment module designed, relied mainly on a trial and error basis since any camera adjustments made by the user were mostly based on his perception. Any change applied could in fact end up achieving the opposite result. In addition of not being very intuitive, the indications provided by the software were based on assumptions that may vary considerably in a real world scenario, such as using lines to measure some of the offset angles, that were created relying heavily on a pair of markers present in the treadmill, whose location is not always perfect for the reason that some of them are detachable and their centres of mass also vary on every image obtained. The new camera alignment method, being an iterative process, does not rely on conditions that may be heavily dependent on a couple of marker locations but instead, in all of them as a group. It also provides quantitative information that is displayed to the user in conjunction with precise instructions on how to move and rotate the camera to align it, as Figure 5.1 shows.

The camera's pose estimation implementation was done in a similar way as described in



Pontos detect. : 10
Max. dist. : 18.2 pixeis

Por favor ajuste a camara:
-14.65 mm para a direita
6.6 mm para a frente
20.3 mm para cima.

Eixo Xx 0.57 graus
Eixo Yy -0.06 graus
Eixo Zz 0 graus

Figure 5.1: Improved Camera Alignment Module window (not aligned)

section 4.4, where the Euler angles are extracted from the rotation matrix and presented to the user in addition with the translation vector. Nevertheless, some minor adjustments were made to compensate for the fluctuation of the detected blob centroids that represent each marker by collecting ten consecutive image frames with all the relevant markers detected and averaging their coordinates. This approach yielded in a series of consistent values that not only depicts a better solution but also does not confuse the user.

Table 5.1: Standard deviation (σ_x, σ_y) values of the centroids for the left and right cameras. (in pixels)

	Left Camera	**Right Camera**
σ_x	0.133	0.175
σ_y	0.120	0.183

Table 5.1 shows, for both left and right cameras, the standard deviation (σ_x, σ_y) values of the centroids [13]. Judging from both camera values, it can be concluded that pixel detection can be quite precise since most of the values that are obtained show deviations that are less than a half a pixel. As for the pose estimation, these results shows that successive adjustment

indications will produce identical result given the camera does not move. Tables 5.2 and 5.3 shows the mean ($\bar{x}$) and standard deviation (σ) values for a sequence of ten consecutive pose adjustments given by the system for both misaligned cameras, without moving them.

Table 5.2: Mean ($\bar{x}$) and Standard Deviation (σ_x) values of 10 consecutive pose adjustments given by the system for the left camera.

	R_x	R_y	R_z	t_x	t_y	t_z
$\bar{x}$	0.424	-0.85	0.054	13.188	5.621	26.052
σ	0.058	0.026	0.019	1.237	0.736	2.032

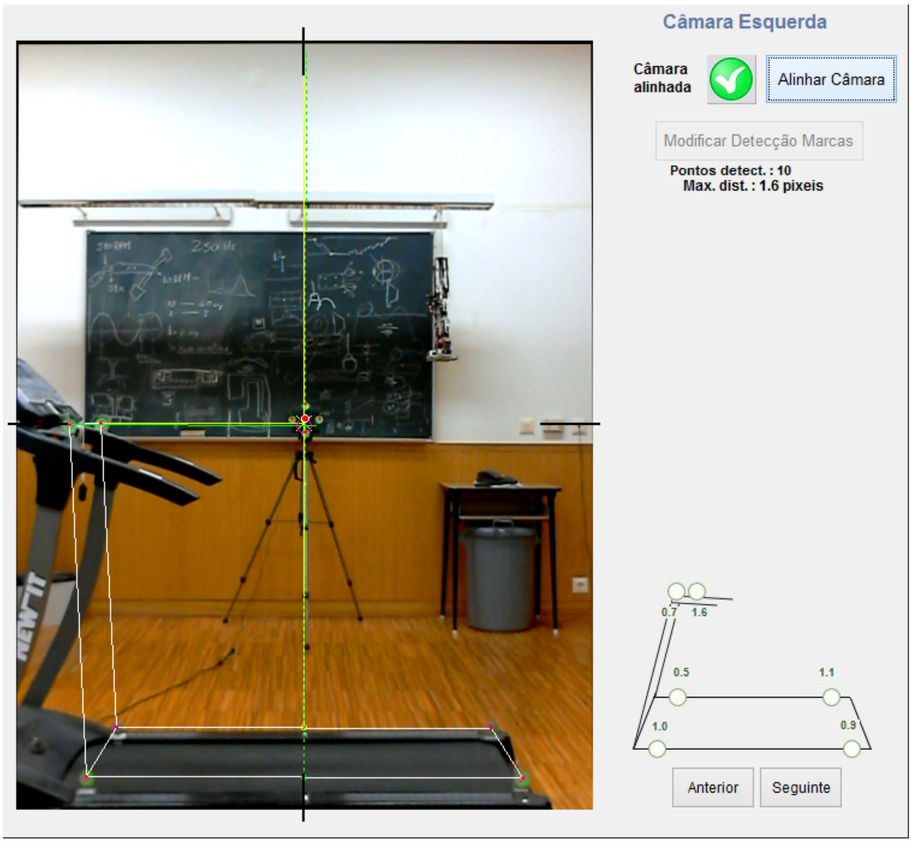
Table 5.3: Mean ($\bar{x}$) and Standard Deviation (σ_x) values of 10 consecutive pose adjustments given by the system for the right camera.

	R_x	R_y	R_z	t_x	t_y	t_z
$\bar{x}$	0.194	-0.346	-0.144	44.178	51.047	-1.645
σ	0.031	0.019	0.198	0.0644	1.145	1.099

Needless to say, each adjustment instruction given is independent from the previous one, meaning that even when successive values are slightly off from one another it can still lead to the intended pose. However, by looking at both tables 5.2 and 5.3 an idea on how those values change without moving the camera, can be perceived and can be concluded that, from the user perspective, a successive set of instructions won't produce inconsistent values that might get him confused on how to operate.

Following the instructions given allow the user to quickly align each camera. Variations on the lighting conditions present on the room will always play an important role over the marker detection process which will in turn contribute to the time it takes to complete the alignment when some markers can't be detected. However, considering that they always are, the alignment can be quickly concluded by following the instructions given, without even looking at the image being captured by the camera. Having in consideration that in the real world scenario there is no way to verify if the corrected camera pose achieved by following the new method corresponds to the ideal pose, or very close to it, measuring tape was used, which confirmed that the correct distances were achieved. As for the orientation, there is no way to measure them accurately.

In Figure 5.2 is shown a completed alignment that results in a maximum pixel difference between target and detected points of 1.6 pixels. Only two of the six points have a difference



Câmara Esquerda

Câmara alinhada ✓

Pontos detect. : 10
Max. dist. : 1.6 pixeis

Figure 5.2: Improved Camera Alignment Module (camera aligned)

superior to one pixel. This value differences could also be achieved using the previous method but at a cost of a considerable amount of time. It is also worth to mention that the only factor that, without any doubt, greatly increases the amount of time it takes to complete the alignment process, are the tripods used. They do not allow precise rotations to be made without affecting the other axes.

Regarding the corrections that could potentially be applied, judging by the instructions given by the new method when a camera is already within acceptable conditions to be considered aligned (less than 0.2° in all axes) and comparing those values with the ones obtained from the simulations (Figures 4.9 and 4.10) it is concluded that the differences between the angles before and after potential corrections are minimal, thus not being required. As a final note, when comparing the error introduced by each camera's pose and the mean error of the system for simple angles (2°), reported in [7], it becomes even more evident that the error introduced by each camera's pose, when they are within the system's acceptable values, can be considered negligible.

Chapter 6

Conclusions and Future Work

The pose estimation method for the cameras, applied during this thesis work, revealed to have improved considerably the process of setting up the equipment. It not only gives quantitative information that is used to display instructions to the user on how to move the cameras, but also allows for a precise estimation on each camera's pose that previously could not be found. The alignment process also became more robust considering it now treats the markers from the treadmill as a whole, meaning that it does not rely heavily on the information provided from only a few of the markers anymore. The instructions given to the user are consistent and not confusing. The coordinates of the centroids of each marker can be detected with great accuracy, specially when taking in consideration the webcams are working at a resolution of only 0,3 MP.

The Euler angles obtained from the estimation method, when the camera is within the acceptable parameters to be considered as being aligned, in conjunction with the results given by the simulations, indicate that the differences between the real and the perceived angles by the camera are very small, thus not having a negative impact over the final results.

As mentioned in chapter 5 the Gait Acquisition and Analysis System has a mean error of 2° for the simple angles [7], which is very good since the human gait presents deviations above 6° relatively to mean values [5]. Nevertheless, the data obtained from the present work, suggest that the error introduced by the poses of the cameras that are accepted by the system is even lower and can be neglected.

As future work, by analysing the present Gait Acquisition and Analysis System, it is noticed that it already presents excellent results when compared to available commercial alternatives such as Vicon Bonita or Optotrak Certus since it offers the best quality/price ratio. Even so, there is some available margin to improve it even more. The current system's depth estimation can be improved not only by using more cameras [12] but also by using different types of cameras or both. That way, the equipment itself could be used to enhance the depth estimation without even requiring the physical characteristics of the patients, such as the distance between shoulders, to acquire data that can be even more precise. The equipment and configuration chosen should take into account the costs involved since it is desired to keep the current system within a low price tag. For that reason and having in consideration that at the moment, no specified maximum value for the project is known, no recommendations will be made in this work.

Chapter 7

Bibliography

[1] J.K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428 – 440, 1999.

[2] J.K. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Nonrigid motion analysis: Articulated and elastic motion. *Computer Vision and Image Understanding*, 70(2):142 – 156, 1998.

[3] Thomas P. Andriacchi and Debra E. Hurwitz. Gait biomechanics and the evolution of total joint replacement. *Gait & Posture*, 5(3):256 – 264, 1997.

[4] CatharineD. Barclay, JamesE. Cutting, and LynnT. Kozlowski. Temporal and spatial factors in gait perception that influence gender recognition. *Perception & Psychophysics*, 23(2):145–152, 1978.

[5] Andrew A Biewener, Claire T Farley, Thomas J Roberts, and Marco Temaner. Muscle mechanical advantage of human walking and running: implications for energy cost. *Journal of Applied Physiology*, 97(6):2266–2274, 2004.

[6] Hsuan-Sheng Chen, Hua-Tsung Chen, Yi-Wen Chen, and Suh-Yin Lee. Human action recognition using star skeleton. In *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks*, VSSN '06, pages 171–178, New York, NY, USA, 2006. ACM.

[7] A. Coimbra, J. Ferreira, P. Ferreira, and M. Crisostomo. Human gait acquisition and characterization with a low cost vision system. Technical report, QREN- Quadro Referencia Estrategica Nacional, 2014.

[8] John J Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River, 2005.

[9] Dictionary.com. "workflow," in the free on-line dictionary of computing. `http://dictionary.reference.com/browse/workflow`. Last accessed: September 01, 2015.

[10] Paulo Ferreira. *User Manual - Vision System for Human Gait Characterization.* Institute of Systems and Robotics - University of Coimbra, June 2015.

[11] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach.* Always learning. Pearson, 2012.

[12] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[13] Todd C Headrick. *Statistical simulation: Power method polynomials and other transformations.* CRC Press, 2009.

[14] I.P. Howard and B.J. Rogers. *Perceiving in Depth, Volume 2: Stereoscopic Vision.* Oxford Psychology Series. Oxford University Press, USA, 2012.

[15] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, 1973.

[16] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

[17] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168, 1944.

[18] Haojie Li, Shouxun Lin, Yongdong Zhang, and Kun Tao. Automatic video-based analysis of athlete action. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 205–210, Sept 2007.

[19] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[20] mathworks.com. "coordinate systems" in mathworks matlab's documentation. `http://www.mathworks.com/help/vision/gs/coordinate-systems.html`. Last accessed: September 01, 2015.

[21] mathworks.com. "single camera calibrator app" in mathworks matlab's documentation. `http://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html`. Last accessed: September 01, 2015.

[22] Angelo Matsas, Nicholas Taylor, and Helen McBurney. Knee joint kinematics from familiarised treadmill walking can be generalised to overground walking in young unimpaired subjects. *Gait & posture*, 11(1):46–53, 2000.

[23] E.A. Maxwell. *The methods of plane projective geometry based on the use of general momogeneous coordinates.* Cambridge University Press, 1946.

[24] E.A. Maxwell. *General Homogeneous Coordinates in Space of Three Dimensions.* Cambridge University Press, 1951.

[25] P.J. Narayanan, S.K. Nayar, and H.Y. Shum. *Computer Vision - ACCV 2006: 7th Asian Conference on Computer Vision, Hyderabad, India, January 13-16, 2006, Proceedings.* Number pt. 2 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006.

[26] William M. Newman and Robert F. Sproull. *Principles of Interactive Computer Graphics (2Nd Ed.).* McGraw-Hill, Inc., New York, NY, USA, 1979.

[27] opencv.org. "solvepnp" in opencv 2.4.11.0 documentation. `http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html`. Last accessed: September 04, 2015.

[28] J. Rittscher, A. Blake, and S.J. Roberts. Towards the automatic analysis of complex human body motions. *Image and Vision Computing*, 20(12):905 – 916, 2002.

[29] D.F. Rogers and J.A. Adams. *Mathematical Elements for Computer Graphics.* Computer science series. McGraw-Hill, 1990.

[30] Sheldon R. Simon. Quantification of human motion: gait analysis-benefits and limitations to its application to clinical problems. *Journal of Biomechanics*, 37(12):1869 – 1880, 2004.

[31] Gregory G Slabaugh. Computing euler angles from a rotation matrix. *Retrieved on August*, 6(2000):39–63, 1999.

[32] Sarah V. Stevenage, Mark S. Nixon, and Kate Vince. Visual analysis of gait as a cue to identity. *Applied Cognitive Psychology*, 13(6):513–526, 1999.

[33] support.logitech.com. "logitech hd webcam c270 technical specifications". `http://support.logitech.com/en_gb/article/17556`. Last accessed: September 05, 2015.

[34] Faezeh Tafazzoli, George Bebis, Sushil Louis, and Muhammad Hussain. Improving human gait recognition using feature selection. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Ryan McMahan, Jason Jerald, Hui Zhang, StevenM. Drucker, Chandra Kambhamettu, Maha El Choubassi, Zhigang Deng, and Mark Carlson, editors, *Advances in Visual Computing*, volume 8888 of *Lecture Notes in Computer Science*, pages 830–840. Springer International Publishing, 2014.

[35] Nikolaus F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2(5):2, 2002.

[36] Emanuele Trucco and Alessandro Verri. *Introductory techniques for 3-D computer vision*, volume 201. Prentice Hall Englewood Cliffs, 1998.

[37] Michael W. Whittle. Clinical gait analysis: A review. *Human Movement Science*, 15(3):369 – 387, 1996.

Appendix A

Tables

This appendix presents the coordinate points acquired during the experimental results.

Tables A.1 and A.2 present the values for the left camera coordinates x, y of the markers for a given pose. Tables A.3 and A.4 present the values for the right camera coordinates x, y of the markers for a given pose. Tables A.5 and A.6 present the values returned from the camera pose estimation method applied in this thesis, for a given pose of the left and right cameras, respectively.

Table A.1: Left Camera: X coordinates of markers from the treadmill.

x	m1	m2	m3	m4	m5	m6
	40,911	55,095	416,025	67,398	81,385	391,263
	41,024	55,382	415,916	67,486	81,346	390,885
	40,853	55,166	416,017	67,392	81,500	390,831
	40,868	55,029	416,109	67,398	81,364	391,051
	40,992	54,814	416,045	67,330	81,392	390,500
	40,864	55,364	416,068	67,323	81,474	390,603
	40,911	55,275	416,043	67,276	81,325	390,972
	40,968	54,818	416,076	67,396	81,303	390,750
	40,853	55,191	416,043	67,390	81,280	390,824
	40,901	55,250	415,925	67,323	81,303	391,000
STD_dev	0,061	0,202	0,062	0,059	0,073	0,221
Variance	0,004	0,041	0,004	0,003	0,005	0,049
mean_var	0,018					
group_std_dev	0,133					

Table A.2: Left Camera: Y coordinates of markers from the treadmill

y	m1	m2	m3	m4	m5	m6
	305,911	597,276	598,580	305,942	557,308	557,363
	305,898	597,353	598,227	305,943	557,308	557,154
	306,000	597,353	598,598	306,069	557,378	557,182
	306,050	597,200	598,550	305,942	557,247	557,141
	305,960	596,985	598,411	306,278	557,311	557,500
	305,896	597,144	598,610	306,031	557,395	557,164
	305,911	597,176	598,647	305,888	557,364	557,268
	305,808	597,371	598,546	306,059	557,224	557,500
	306,000	597,168	598,491	305,820	557,280	557,392
	306,008	597,265	598,292	306,031	557,224	557,000
STD_dev	0,072	0,120	0,141	0,126	0,062	0,167
Variance	0,005	0,014	0,020	0,016	0,004	0,028
mean_var	0,015					
group_std_dev	0,120					

Table A.3: Right Camera: X coordinates of markers from the treadmill

x	m1	m2	m3	m4	m5	m6
	414,5	403,9245	92,56667	444,6667	428,14	70,4466
	414,2424	403,6452	92,48276	444,6746	428,1915	70,55963
	414,1364	403,5	92,81081	444,608	428,125	70,57692
	414,2857	403,3803	92,06667	444,5656	428,2526	70,39362
	414,459	403,6935	92,84375	444,64	428,2021	70,31731
	414,2857	403,5	92,5	444,6693	428,2088	70,53846
	414,2167	403,5	92,87097	444,7692	428,2088	70,30303
	414,2344	403,8448	92,97436	444,6406	428,1579	70,33663
	414,5	403,6935	93	444,608	428,0769	70,27885
	414,5	403,4531	93,24	444,6825	428,1579	70,24742
STD_dev	0,139	0,177	0,335	0,055	0,051	0,123
Variance	0,019	0,031	0,112	0,003	0,003	0,015
mean_var	0,031					
group_std_dev	0,175					

Table A.4: Right Camera: Y coordinates of markers from the treadmill

y	m1	m2	m3	m4	m5	m6
	317	568,2075	572,3333	316,4715	608,34	610,5146
	317,197	567,9839	572,3793	316,5635	608,1915	610,4404
	316,8636	568	571,8108	316,768	608,7292	610,4231
	317	567,9437	572,2	316,7377	608,6737	610,6489
	316,9344	568,0645	571,7188	316,768	608,6489	610,5769
	317	568	572,5	316,8425	608,4725	610,5385
	316,6833	567,9688	572,2903	316,7308	608,4725	610,6263
	317,0469	567,8103	571,8462	316,7188	608,1474	610,7327
	317	567,6452	572	316,768	608,7692	610,4231
	317	567,9219	572,36	316,7937	608,1474	610,6701
STD_dev	0,132	0,149	0,277	0,112	0,243	0,110
Variance	0,017	0,022	0,077	0,013	0,059	0,012
mean_var	0,033					
group_std_dev	0,183					

Table A.5: Successive poses obtained for the left camera (without moving it)

POSE	x	y	z	dx	dy	dz
esq	0,460	-0,060	-0,060	-14,670	6,250	25,380
	0,460	-0,060	-0,060	-14,750	6,220	24,720
	0,340	-0,060	-0,06	-14,23	6,77	28,76
	0,400	-0,060	-0,060	-13,66	6,33	27,43
	0,400	-0,110	-0,060	-11,59	5,22	26,53
	0,460	-0,110	-0,060	-12,87	5,09	24,3
	0,520	-0,110	0,000	-11,86	4,42	22,73
	0,460	-0,110	-0,060	-12,46	5,04	24,87
	0,340	-0,110	-0,060	-11,73	5,47	29,15
	0,400	-0,060	-0,060	-14,06	5,4	26,65
mean	0,424	-0,085	-0,054	-13,188	5,621	26,052
std_dev	0,058	0,026	0,019	1,237	0,736	2,032

Table A.6: Successive poses obtained for the right camera (without moving it)

	x	y	z	dx	dy	dz
dir	0,170	-0,340	0,230	43,27	52,84	-1,43
	0,170	-0,340	-0,230	43,72	52,25	-1,6
	0,17	-0,34	-0,23	44,6	52,06	0,54
	0,17	-0,34	-0,23	44,71	51,64	-0,79
	0,23	-0,34	-0,23	43,81	50,94	-2,09
	0,17	-0,34	-0,23	43,67	51,19	-1,89
	0,23	-0,34	0,23	44,47	50,32	-2,49
	0,23	-0,34	-0,23	43,66	49,73	-3,52
	0,23	-0,34	-0,23	44,52	49,52	-2,23
	0,17	-0,4	-0,29	45,35	49,98	-0,95
	0,194	-0,346	-0,144	44,178	51,047	-1,645
	0,031	0,019	0,198	0,644	1,145	1,099