Master in Artificial Intelligence (UPC-URV-UB)

Master of Science Thesis

HUMAN POSE RECOVERY AND BEHAVIOR ANALYSIS FROM RGB AND DEPTH MAPS

Miguel Reyes Estany

Advisor: Dr. Sergio Escalera

5/9/2011

Abstract

Robust human pose recovery and automatic behavior analysis has applications including gaming, human-computer interaction, security, telepresence, and health-care, just to mention a few. In this work, we present a generic framework for human posture analysis and gesture recognition using RGB-D representation. It encompasses the process we have undertaken to analyze human postural configurations reliability and robustness. The work ranges from the process of image acquisition, beginning in geometric models of image representation, in order to understand the power and the use of RGB-D spaces. Having described this technology, the main focus of this work is based on the location and description of human models which can represent and describe the human pose with high accuracy. We defined an accurate pose descriptor, and defined a generic framework for automatic multi-class behavior analysis. Several applications of the proposed methodology are also presented and discussed.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Sergio Escalera for his support and dedication for reviewing my master thesis. I also like to thank him for the opportunity he gave me to dive into the world of research, without him it would not have been possible. His unvaluable advices have meant a lot to me.

I would like to thank Mr. Jose Ramírez and Instituto Global Mezieres inc. for introducing me to the world of rehabilitation analysis and offered me the opportunity and the resources to develop a real application related to my research, allowing me to make this work more robust.

I would like to thank Dr. Josep Moya for entering knowledge in the AHDH disorder and have wagered on this methodology for their clinical studies.

Finally I would like thank all the support I have received by Dr. Petia Radeva and my future doctoral colleagues. Without their contributions this work would not have obtained the same results. So thank AI research team at the University of Barcelona and Computer Vision Centre at the University Autonoma of Barcelona, and especially my teammates Antonio Hernandez, Miguel Angel Bautista, Alexander Marinov, Nadezhda Zlateva and Gabriel Dominguez, who have lived very closely the evolution of this work.

Table of contents

Master in Artificial Intelligence (UPC-URV-UB) 1

Master of Science Thesis 1

Introduction 6

1.1. Motivation 7

1.2 Outline 7

State of the art 9

2.1 Technical background in Computer Vision 9

2.2 Technical background in Multimodal Computer Vision 11

Acquiring and processing depth data 13

3.1 Introduction 13

3.2 Formation of an image 13

3.3 Pin-hole model 14

Distortions and calibration 17

4.1 Lens distortion 17

4.1.1 Radial distortion 18

4.1.2 Tangential or off-center distortion of the image: 19

4.1.2 Prism Distortion: 19

4.2 Calibration 20

4.2.1 Calibration methods: 20

4.2.1 Calibration template 2D based 23

4.2.2 Computation of homographies 23

4.2.3 Calibration process with 2D templates 25

4.2.2 Full automatic process of calibration 26

RGB-D, Visual advantage 28

5.1 RGB-D technology 28

5.2 RGB-D alignment procedure 29

5.2.1 RGB-D Mapping 30

5.2.1 Loop closure detection 32

5.2.2 Surfel representation 33

Modeling Human Pose 34

6.1 Getting the model articulated by landmarks 34

6.1.1 Color learning 35

6.1.1.1 Experimental results........ 37
6.1.1.2 System enhancements........ 37
6.1.2 Constitution of the skeletal model........ 38
6.1.2.1 Shape Context........ 39
6.1.2.2 Active Shape Models........ 40
6.1.3 Experiments for the reconstruction of the skeleton........ 46
6.2 Getting the model articulated using depth maps........ 47
6.2.1 Framework description........ 47
6.2.2 Random Forest........ 48
6.2.3 Graph-Cuts segmentation........ 49
6.2.3.1 Unary potential........ 51
6.2.3.2 Boundary potential........ 51
6.2.4 Experiments and results........ 52
6.2.4.1 Random forest results........ 53
6.2.4.2 Graphs-cut results........ 55
Gesture Recognition........ 57
7.1 Gesture recognition with an improved DTW........ 57
7.1.1 Begin-end of gesture detection........ 59
7.1.2 Feature Weighting in DTW........ 60
7.1.3 Experiments and results........ 62
Applications developed........ 64
8.1 ADIBAS Posture........ 64
8.1.1 Motivation........ 64
8.1.1 State of the art........ 65
8.1.2 Methodology........ 65
8.1.3 Pre-process: Calibration........ 67
8.2 ADHD disorder analysis........ 69
8.2.1 Motivation........ 70
8.2.2 Goal........ 70
8.2.3 Implementation........ 71
Annex I Contributions........ 75
Annex II Teaching........ 76
Bibliography........ 77

Chapter 1

Introduction

Human motion capture is an essential acquisition technology with many applications in computer vision. However, detecting humans in images or videos is a challenging problem due the high variety of possible configurations of the scenario, such as changes in the point of view, different illumination, and background complexity. There are many recent methodologies from an extensive research on this topic [1, 2, 3, 4]. Most of these works focus on the extraction and analysis of visual features. These methods have made a breakthrough in the treatment of human motion capture, achieving high performance despite having to deal with the similarities between the foreground and the background in case of possible changes in light or view. In order to treat human pose in uncontrolled scenarios, there is recent work using range image for object recognition or modeling [5]. This new approach introduced a solution to the problem of intensity and view changes in RGB images through the representation of 3D structures. At its inception, development and advancement of these new methods came slowly since data acquisition devices were expensive and bulky, with cumbersome communication interfaces when conducting experiments. However, Microsoft has recently launched the Kinect, a cheap multisensor device based on structured light technology, capable of capturing visual depth information (RGBD technology, from Red, Green, Blue, and Depth, respectively). The device is so compact and portable that can be easily installed in any environment to analyze scenarios where humans are present. In recent years, researchers have also used different methodologies and techniques for constructing 3D structures, such as stereoscopic images [9,10]. However, in this case the problems of different lighting conditions and calibration still exist. Some of the research has focused on the use of time-of-flight range cameras (TOF) to use in human parts detection and pose estimation [8,9, 10], combining depth and RGB data [11].

Following the high popularity of Kinect and its depth capturing abilities, there exist a research interest for improving the current methods for human pose and hand gesture recognition. While this could be achieved by interframe feature tracking and matching against predefined gesture models, there are scenarios where a robust segmentation of the hand and arm regions are needed, e.g. for observing upper limb anomalies or distinguishing between finger configurations while performing a gesture. In that respect, depth information appears quite handy by reducing ambiguities due to illumination, color and texture diversity. Many researchers have obtained their first results in the field of human motion capture using this technology. In particular, Shotton et al. [12] present one of the greatest advances in the extraction of the human body pose from depth images that also form the core of the Kinect human recognition framework. These major advances have been the reference and starting point of this work.

1.1. Motivation

In the Computer Vision field, Human Action/Gesture recognition is a challenging area of research that deals with the problem of recognizing people in images, detecting and describing body parts, inferring their spatial configuration, and performing action/gesture recognition from still images or image sequences. Because of huge space of human configurations, body pose recovery is a difficult problem that involves dealing with several distortions: illumination changes, partial occlusions, changes in the point of view, rigid and elastic deformations, or high inter and intra-class variability, just to mention a few. An example of application is the World Challenge in Human Layout analysis of the PASCAL VOC challenge. In 2010, the best groups of research in the area, in a set of near 400 people images of uncontrolled environments achieved accuracy about 70% for face detection, 10% in hand detection, and 2% in foot detection. Even with the high difficulty of the problem, modern Computer Vision techniques and new tendencies deserve further attention, and promising results are expected in the next years. Moreover, several subareas have been recently defined, such as Affective Computing, Social Signal Processing, Human Behavior Analysis, or Social Robotics. The effort involved in this area of research will be compensated by its potential applications: TV production, home entertainment (multimedia content analysis), education purposes, sociology research, surveillance and security, improved quality live by means of monitoring or automatic artificial assistance, etc.

1.2 Outline

This dissertation deals with the vision-based analysis of scenes involving humans. The general approach has been to make extensive use of prior knowledge, in terms of generic 3-D human models, in order recover 3-D shape and pose information from a the RGB-D space.

The dissertation is divided in eight chapters. The strategy of explanation of our methodology is an evolutionary order. Chapter 2 relates about the state of the art. Chapter 3 begins by describing the model as an image acquisition by a regular camera, describing the pin-hole model. This chapter is essentially theoretical, aimed to give the author the basics such as the procurement process for the images to have knowledge of how images are represented, and as world coordinates are adapted to be represented and coded in the systems digital for further processing. In Chapter 4, we delve into the representation of images through the pin-hole model. In this chapter discusses the various distortions in standard cameras and calibration processes that allow us to do translations between camera coordinates and world coordinates. This chapter is vital to provide robustness and reliability of the data that will be discussed throughout the thesis, will be essential especially in the developed application Adibas posture, where reliability and accuracy of the data is very important for the purpose of the application. In Chapter 5 we enter the world of an existing technology a decade ago, but now exploding due to the entry of inexpensive and accurate devices available to all RGB-D technology. This chapter will detail the complex process of aligning the RGB space with their respective depth efficiently. Chapter 6 refers to the skeletal models make up the different postural configurations that a subject can perform and the system must be able to describe. The chapter is mainly divided into two methodologies. A first approach is based on the placement of physical markers on which the system must know reconstruct the skeletal model consistently and automatically. The second part of the chapter describes a technique for obtaining a skeletal model consists of 15 joints automatically, i.e. without placing markers only by analyzing the RGB-D space. The purpose of this chapter is to provide a vector of characteristics that indicate the body pose or configuration of a subject at a certain instant. From this array of features developed, in Chapter 7 we describe a novel methodology to estimate a gesture or movement by an actor. Finally in Chapter 8 describes different applications on which we have tested and validated the methodologies that are cited in this work.

Chapter 2

State of the art

2.1 Technical background in Computer Vision

From a Computer Vision point of view, Human Behavior analysis for Human Action/Gesture recognition can be split in two main stages: 1) Pose recovery and 2) Action/gesture recognition. Although plenty of literature can be found in both lines of research, next, we join them in a common taxonomy and briefly describe their goals.

1) Pose recovery: There exist several studies for pose recovery. Following the standard Pattern Recognition pipeline, most of them perform the following tasks: a) Image measurements and b) Body parts learning.

    a. Image measurements: Image measurements contain the steps of image pre-processing (i.e. illumination normalization or noise filtering), feature detection (i.e. detection of salient points), and feature description. The main challenges of this step are focused on the feature detection and description methodologies. Depending if the action/recognition is computed from single images of video sequences, different feature detection and description methods can be applied. Feature detection if often performed as the detection of image salient points or by detection of prior landmarks, which are inferred by means of template matching or template learning. A detailed list of key point detectors can be found in [13] for the case of still images. Since one of the main reasons that reduce repeatability in key point detection is due to changes in appearance, new techniques tend to deal with the problem of feature inter/intra-class variability. For instance, in the work of [14], GroupLets are introduced as general features that allow for feature variability tolerance by inferring logical operators among feature relation measurements.

       Regarding feature description, several approaches for still images have been proposed based on color representation, contour/shape analysis, or pixel orientation distribution, just to mention a few (see [15] for more details). In the case of image sequences, similar approaches have been proposed, some of them taking benefit from the temporal dimension. For instance, the description approach of [16] combines the Histogram of Oriented gradients description with an Histogram of Oriented Flow vectors descriptors, looking for both spatial and temporal coherence of 3D region descriptions. In order to achieve higher tolerance to 3D key point distortions, the authors of [17] include an extra description to 3D patches so that codify trajectory coherence within the video sequence.

b. Body parts learning: Once we have defined a feature detection/description procedure, in order to perform pose recovery, body parts use to be detected. This step is particularly challenging because of the huge inter/intra-limb feature variability in both still images and image sequences. Because of this reason, most of the state-of-the-art approaches for pose recovery are based on limb detection from a pre-defined set of possible human views, applied to a particular set of applications, and with a reduced number of poses. In order to reduce the confusion and false positive detections that body limbs introduce, spatial coherence of body parts is also taken into account. For instance, in the approach of [18], PoseLets are defined as a two level pose inference procedure. At the first stage, HOG descriptors of body parts are learned using SVM classifiers. At a second procedure, the classifier answers are ranked, and a second classifier codifying a weighted spatial coherence is trained [19-20]. This approach has shown robust results, reducing dramatically the performance when introducing new poses with high changes in the body point of view. Other works are used on performing inference of Graphical Models including appearance and/or spatial relations of body parts and performing inference in order save probabilities for those more likely limb relations given a particular training set [21-22]. For the cases where the space of poses is large and it is difficult to perform inference for a unique statistical models, the authors of [19-20] propose the AND-OR Graph. In this case, each possible path from the root to a leave of the structure represents a pose, and each internal node can have logical operators in order to allow for different configurations of body parts.

2) Action/gesture recognition: This step of the process is usually defined as an association of the inferred body parts with a prior action/gesture knowledge (i.e. coming from deformable models, 2D & 3D models, or motion priors). For summary purposes, we divide the state-of-the-art methods in four groups:

Temporal templates: a patch is used to find correspondences (i.e. using normalized cross-correlation) in the whole image/sequence (see Figure 1(a)). These methods are simple and fast but are very sensitive to segmentation errors. A common procedure in this case is to define a Bag-of-Visual-Words model, where the frequency of appearance of each body part is computed in an histogram, which can come from single or a sequence of images. After template matching, different learning approaches can be performed to perform action/gesture recognition.

Active Shape models: Allows for shape regularization, but are sensitive to initialization and tracking failures (see Figure 2.1(b)). One of the benefits of these methods is that they involve a matching cost which can be directly used as an action/gesture classification threshold [25].

Tracking with motion priors: These methods take benefit of the pose information linked to the tracking process to perform simultaneous action recognition. However, these methods are also sensitive to initialization and tracking failures (see Figure 2.1(c)).

Motion-based recognition: These methods are useful for generic descriptors and less dependent on appearance. On the other hand, they are sensitive to localization and tracking errors (see Figure 2.1(d)).

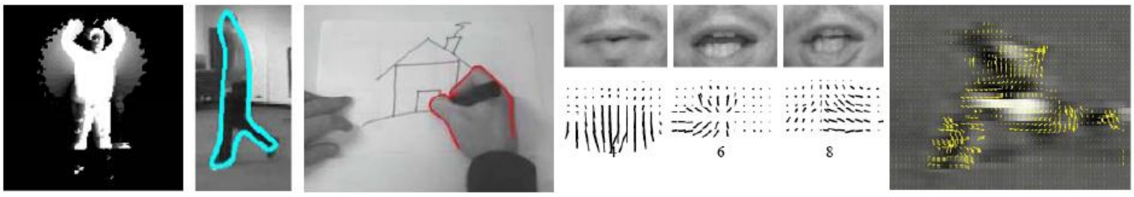


Figure 2.1 Examples of (a) Temporal templates, (b) Active Shape models, (c) Tracking with motion priors, and (d) Motion-based recognition.

Independently of the family a particular action/gesture strategy belongs to, once pose vector descriptors are obtained for a particular image sequence, general Pattern Recognition and Machine Learning Strategies that involves spatial or temporal relations can be applied to perform final action/gesture recognition. Examples of common strategies are Neural Networks, Stacked Sequential Learning, of different kind of Bayesian Networks, such as Conditional Random Fields or Hidden Markov Models. Another common strategy used to match temporal series for gesture recognition is Dynamic Time Warping. The method comes from Dynamic Programming discipline of algorithmic, and has been extended to allow temporal and spatial deformation of gestures to include invariance to gesture speed and person physical characteristics.

2.2 Technical background in Multimodal Computer Vision

Previous Computer Vision approaches do not require working alone. The different visual descriptors can be combined with other discriminative features coming from different sensors. In the same way, most previous approaches for final inference can be applied independently of the considered feature space. For instance, some actions/gestures can have similar visual representation but they can be split using audio features. In other cases, only visual information is discriminative enough but classification strategies do not generalize. In some of these cases, the combination with proper data features from different sensors can increase generalization capability of the system.

Recently, with the arrival of the Kinect hardware/software to the market, the Depth map information has obtained much attention. The multi-sensor Kinect combines accelerometer, video information, with a Depth map, allowing for a new RGBD representation. The Depth map is computed from the Kinect infrared sensor. The infrared sensor displays a set of points though the environment. Then, each depth pixel is computed by sampling the derivative of the higher resolution infrared image taken by the infrared camera. This value would be inversely proportional to the radius of each Gaussian dot, which is linearly proportional to the actual depth. The authors of the main Kinect software have recently published their work for action/gesture recognition using Depth maps [12]. Basically, from a huge set of depth maps coming from real and virtual people, a supervised method describe different limb labels using simple depth relations from each described point in a 3D space and its neighborhood applying random offsets. These features are then learnt using a probabilistic Random Forest approach. Though the method produces noisy labeling segmentation, main limb densities are real time captured only applying pixel-wise classification, allowing for the computation of a 15-joint skeleton representation of the human pose. The main failures of this approach are produced when different people become closer in the 3D space or when objects appear next to the person cluster, since depth relations between person and background are altered.

Chapter 3

Acquiring and processing depth data

3.1 Introduction

The use of video cameras in activities metrics has grown considerably in recent years due to the flexibility in the collection of images. It facilitates the acquisition process, since it is not necessary to control the shots, the time exposure, etc. In addition, scenes to be effectively used in processing can be reviewed and selected in the cabinet.

This chapter describes the geometric model associated with the formation process a picture with the camera. The model is characterized by a number of parameters which characterize the intrinsic properties of the camera and its position in the world. From a geometric point of view the situation of a point in the image is the result of linear transformations and perspectives applied to the coordinates of the world. The geometric model is a projection matrix, which contains all these operations. If you do not change the intrinsic properties and the position of the camera is the same possible to know, using this projection matrix, the position of a point in the image knowing the same position in the world. Also, based on the camera model is possible to locate the camera in an environment from images obtained from the same and obtain the intrinsic characteristics of the camera.

3.2 Formation of an image

The process of forming an image consists of two parts to consider. The first part, mainly geometric, the position in the plane of the image, from the projection of a point in the scene. The second is the nature of light which determines the brightness of a point in the plane of the image based on properties surface and lighting.

The most elementary form an image of a 3D scene on a surface 2D is a fully enclosed box. In this case using two sides faced which simulate two screens located in parallel. The first screen is a small very small hole through which only have to spend a photon of light. This hole allows the rays of light emitted or reflected by an object in the scene break through the first screen and form an inverted image in the second as shown in Figure 3.1. To collect the image is simply necessary to place a photosensitive element the second screen. Since in practice the small hole does not let enough light to excite the photosensitive element, is placed in the opening lenses to focus the ray bundle reaching a point in the scene to the corresponding point in the plane of the image.

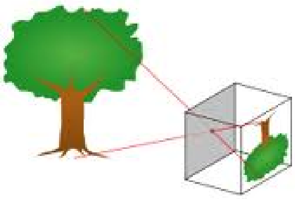


Figure 3.1 Example formation an image in a camera

The effect of the lenses is based on the principle of light refraction. A beam of light refracted when it encounters an obstacle transparent, and as a result, it undergoes a change in his career. This change of direction is determined by the angle of refraction, which depends on the angle of incidence and the wavelength of the light beam. Using the angle refraction is possible with a lens all the rays of light from a same point in the scene intersect at a single point behind the lens. This cutoff depends on the angle of incidence on the lens of the light rays. To get a clear picture of the scene, it is necessary that the image plane is located right in the distance court. This is called focal length of the vision system. The focal length of a lens is the separation between the lens and the cutoff of light rays from infinity. This is not has to coincide with the focal length of the vision system as the latter is set function of distance from the scene is to portray.

3.3 Pin-hole model

This is the simplest model which is used as the basis for all other models. It represents an ideal distortion-free camera. You can obtain the geometrical model of the camera from Figure 3.2. ***I*** is the image plane. ***F*** is the focal plane in which all points have $z_c = 0$. The point ***o'*** is the optical center or center of projection, which is located focal length ***f*** of the origin of the coordinate system of the camera ***c*** (focal length vision system). The optical center is used to form the image of a point ***p*** in the plane image ***I***. The image point ***p*** called a point ***q*** is obtained by the intersection of ***o'p*** straight to the image plane ***I***. The optical axis is the line that passes through the optical center ***o'*** and is perpendicular to the plane ***I***. Point ***c*** is the intersection of the optical axis with the plane of the image, also called the main point. For the pin-hole model of camera is to take this point as the origin of the image measures. The focal plane ***F*** is parallel to the plane of image and passes through the optical center ***o'***. The points in the focal plane with no image the image plane ***I*** and that the line is parallel to the plane ***o'p*** image ***I***.

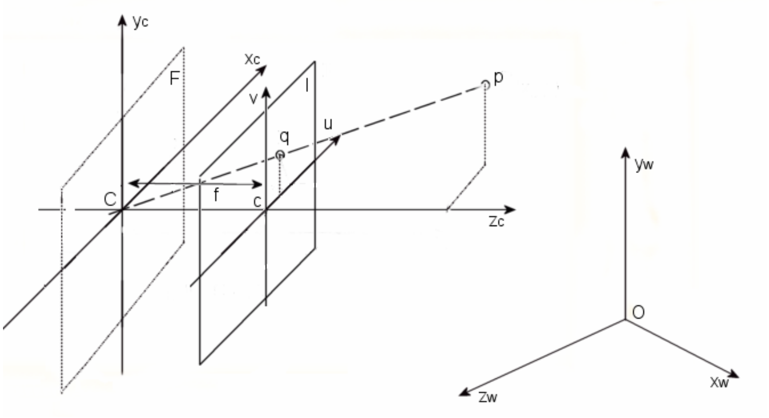


Figure 3.2 Scheme of the pin-hole model

From the viewpoint of projective geometry, this line intersects the plane of the image at infinity. It sets a 3D coordinate system for the stage (o {x_w, y_w, z_w}), another 3D coordinate system for the camera (o', {x_c, y_c, z_c}) and a 2D image plane (c, {u, v}). The optical axis is aligned with the axis **Z** of the reference system of the camera such as shown in Figure 3.2. The coordinates of the points of interest in space are referring to the stage coordinate system (or {x_w, y_w, z_w}) and their corresponding image positions are referred to the system of coordinates of the same (c, {u, v}).

To characterize the ideal camera model must take into account two transformations that perform the same. With these transformations we obtain the coordinates points in the image from their positions on the reference system stage. First, there is a transformation of the reference metric scenario, the metric reference system associated with the camera. Secondly, there is a perspective projection that transforms a point of interest on the Stage at a point 2D image. Given the coordinates of a point ***p*** on the reference system stage p_w = (x_w, y_w, z_w), the coordinates of this point ***p*** on the system camera reference $\mathbf{p_c}$ = (x_c, y_c, z_c) are:

$$p_c = R \,.\, p_w - \ t$$

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11}x_w & r_{12}y_w & r_{13}z_w \\ r_{21}x_w & r_{22}y_w & r_{23}z_w \\ r_{31}x_w & r_{32}y_w & r_{33}z_w \end{bmatrix}$$

Equation 3.1

R is a 3x3 orthonormal rotation matrix that relates both sets of reference ***t*** = (t_x, t_y, t_z) are the coordinates of the center of projection ***o'*** to the origin of coordinates stage. The expression (3.1) represents the relationship between the coordinates of same point as the reference system of the stage or the reference system the camera. The matrix ***R*** and the vector ***t*** represent the correlation between the two reference systems. To obtain the coordinates of ***q*** with respect to the coordinate system image $\boldsymbol{q_c}$ = (u_c, v_c) is performed perspective projection coordinates $\boldsymbol{p_c}$ = (x_c, y_c, z_c) respect to the coordinate system of the camera:

$$\frac{f}{z_c} = \frac{u_c}{x_c} = \frac{v_c}{y_c}$$

Equation 3.2

$$q_c = \begin{bmatrix} u_c \\ v_c \end{bmatrix} = f \begin{bmatrix} \frac{x_c}{z_c} \\ \frac{x_c}{z_c} \end{bmatrix}$$

Equation 3.3

Being ***f*** the focal length of the vision system. In projective space, if $\boldsymbol{z_c}$ = 0 indicates the point is in the focal plane of the camera. In this case the coordinates of the $\boldsymbol{q_c}$ image = (u_c, v_c) are not defined and correspond to a point at infinity.

There are other parameters, these relate the coordinate system of the stage and the system camera coordinates determine the position and orientation of the camera on stage. These are called extrinsic parameters. The camera features a focal length determined and is independent of the position of the same on stage. We will not deal in depth with these parameters, because our work is based on obtaining the world coordinate, without going into the analysis of the position of the camera and position of the stage.

To study the behavior of the camera and be able to treat it as a box black performs a transformation of the coordinates of the projective space P^2 to P^3 space different models which fit more or less the actual behavior of the same. The basic model is the so-called pin-hole from which others are built models. This basic model only transforms the coordinates of the points in the scene. This transformation gives the relation between the reference frame of the camera and the stage. Does not take into account the transformations that occur in the formation of the image. A model adjusted to the actual behavior of the camera adds to the pin-hole model linear transformations suffered by the coordinates of the points in the image. These are the scaling that allows the change of pixel units to millimeters, the location of the origin of measurements in the upper left corner of the image and the non-orthogonality of the axes of measures due to imperfections in the construction process.

Interested in this project is to examine the world coordinate, for it will proceed with the reverse process we have developed in the pin-hole model (equation 3.3).

$$x_w = \frac{(u - x_c) * z_w}{f_x} \; ; \; x_y = \frac{(v - y_c) * z_w}{f_y}$$

Equation 3.4

To obtain the world coordinates are necessary the focal length and the principal point. Therefore we perform a calibration method with which we can obtain the intrinsic values of the camera.

Chapter 4

Distortions and calibration

The calibration is necessary to adjust the model behavior of the camera which takes into account the distortions that occur in the vision system constructive imperfections of the lenses. These imperfections cause deviations in the path of the beam is assumed a priori that straight. In this chapter we are going to analyze how to reduce these imperfections in order to obtain very realistic image, and techniques about calibration, which it allows us to obtain quantitative data from the images.

4.1 Lens distortion

Imperfections in the shape of the lens causing lateral deviation of the beam light that passes through. The result is a point position in the image observed different from its actual position reflecting a point in space. The position deviation is respect to a radial center of distortion. In the case of imperfections [26] due to assembly lenses, are generated both radial and tangential distortions of the positions of points in the image. The consequences are geometric displacement of the points in the image. The visual effect is shown in Figure 4.1.

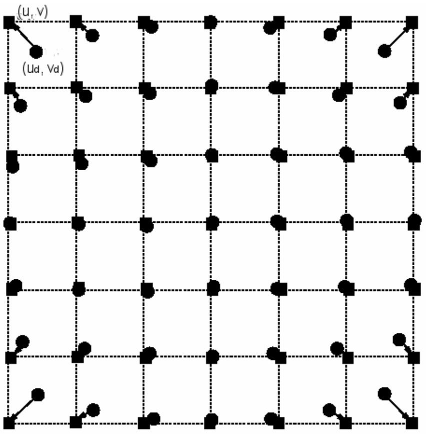


Figure 4.1 Example of distortion by the lens

The correct position in pixels of the point ***q*** in the image ***q_p*** is related to observed in the same position ***q_d***. This relationship is expressed by the following expressions:

$$u_d = u_p - \delta_u(u_p, v_p)$$

$$v_d = v_p - \delta_v(u_p, v_p)$$

Equation 4.1

According to the equation 4.1 the amount of error in each coordinate geometry point ***q*** depends of the position within the image itself.

4.1.1 Radial distortion

The radial distortion produces a shift in the position of the point along the line connecting the point with the center of radial distortion of the image. Normally, this is not coincides with the center of the image. This distortion is caused by defects that exist in curvature of the lens. A negative radial point of the image is also called barrel distortion. The effect produced is that the points in the edges of the image and scale approaching decline. If offset is positive is also known as pincushion distortion. In this case the edge points away from image and scale increases. This distortion [27] is symmetric about the optical axis of the camera. If assumes that the center of distortion is in the center of the image, the expression mathematics that represents this distortion is the next (Equation 4.2):

$$\delta_{ur}\left(u_p, v_p\right) = \Delta u \cdot (k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6 + \cdots)$$

$$\delta_{vr}\left(u_p, v_p\right) = \Delta v \cdot (k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6 + \cdots)$$

Equation 4.2

$\boldsymbol{r}$ is the distance of the pixel to the main point of the picture, $c = (u_o, v_o), \Delta u = u_p - u_o, \Delta v = v_p - u_o$. The radial distortion coefficients are k_1, k_2, k_3,…. The distance $\boldsymbol{r}$ is calculated as follows:

$$r^2 = \Delta u^2 + \Delta v^2$$

Equation 4.3

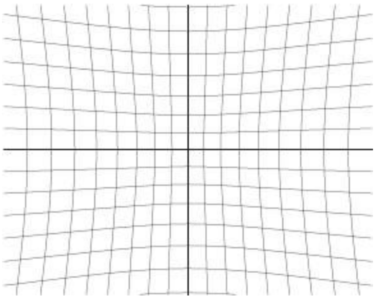


Figure 4.2 Example image of radial distortion

4.1.2 Tangential or off-center distortion of the image:

The optical systems are subject to various degrees of offset because the optical centers of the different lenses are not located in the same line. The defect results call offset a distortion of the image. This distortion has both components, radial and tangential, which can be described by the following expression [27]:

$$\delta_{ud}\left(u_p, v_p\right) = p_1(r^2 + \ 2 \cdot \Delta u^2) + 2p_2 \cdot \Delta_u \cdot \Delta_v + \cdots$$

$$\delta_{vd}\left(u_p, v_p\right) = 2p_1 \cdot \Delta_u \cdot \Delta_v + p_2(r^2 + 2 \cdot \Delta v^2) + \cdots$$

Equation 4.4

$p_1, p_2, p_3, \ldots$ are the coefficients which model the tangential distortion, r is the distance from the principal point to the pixel image.

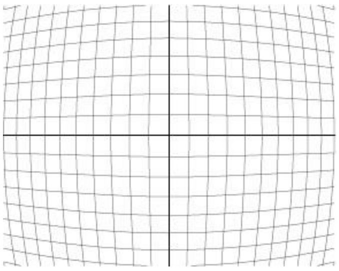


Figure 4.3 Example image of tangential distortion

4.1.2 Prism Distortion:

The prism is distorted [28] due to imperfections in the lens design, in manufacture and assembly. Mainly it is slight shift of some lenses resulting in a lack of perpendicularity to the optical axis of the camera. This distortion can be modeled by adding a small prism to the camera's optical system which causes a radial and tangential greater distortion. In this case the distortion is modeled as follows:

$$\delta_{up}\left(u_p, v_p\right) = s_1 \cdot r^2 + s_2 \cdot r^4 + s_3 \cdot r^6 + \cdots$$

$$\delta_{vp}\left(u_p, v_p\right) = s_1 \cdot r^2 + s_2 \cdot r^4 + s_3 \cdot r^6 + \cdots$$

Equation 4.5

r is the distance to the main point of the pixel of the image and s_1, s_2, s_3, ... are the coefficients to model this type of distortion.

4.2 Calibration

The process of calibrating a camera is a necessary step to extract 3D information from 2D images. Much work has been performed in this field which is to estimate the intrinsic and extrinsic parameters of the same from one or more images obtained from a template. This chapter presents the state of the art both of the different calibration techniques, and the different aspects that affect the process of calibrating a camera. First named the currently existing methods for calibrating a camera, establishing different classifications according to techniques, results and elements necessary for calibration.

Based on all existing methods have chosen a calibration process according to various studies by authors [29], [30], [31] is more interesting for calibration of a camera under certain conditions. The goal is to define the most comprehensive calibration method that allows solving the largest number of possible scenarios based on existing calibration techniques to date. Once the method of calibration intervals will be calculated on the results and optimize it using statistical techniques.

4.2.1 Calibration methods:

The calibration of a camera is the first step to solve applications where it is necessary to obtain quantitative data of the image. Although it is possible to obtain scene information from images taken with uncalibrated cameras [32], the calibration process is essential when trying to obtain measurements of the same. The accurate calibration of the camera allows for distances in the real world from images taken from the same [33]. For example, from a standpoint of location of objects, you can place them in the real world when you have a picture of them. This location can be absolute with respect to an origin of world coordinates or relative to other objects. This makes it possible to solve industrial parts assembly [34] or avoid obstacles in the navigation of a robot [39]. If instead we focus on 3D reconstruction of objects, each image point determines an optical beam passing through the optical center of camera at the scene. The management of multiple images of a scene in which there is no movement allows connecting the two optical beams for the 3D position of points in the scene [36], [37]. In this case it is necessary to solve the step of matching an object in different images [38]. To take several images of the scene can use a single moving camera, multiple cameras mounted on a stereo system or a source of structured light. Once you have able to perform 3D reconstruction of the object, it can be compared with a stored model to determine the result of imperfections in the same manufacturing process.

The visual inspection is a useful tool for quality control, which lets you browse all the products automatically and comprehensively, which means a significant improvement over human inspection which requires statistical tools for its realization. Part of camera calibration is to estimate the intrinsic parameters of the model itself which the camera's internal geometry and optical characteristics the sensor. These parameters determine the coordinates of a point in the image from the point position in the scene with respect to the coordinate system of the camera. It is also necessary to take into account the parameters that measure the distortion of the image produced by the constructive imperfections of the camera. These parameters are used to correct the position of the points in the image getting behavior of the camera ideal for the pin-hole model. Estimating the geometric relationship between the camera and the scene or between different cameras is also important in the calibration process.

The extrinsic parameters measure the position and orientation of the camera about the coordinate system established for the world. These give the relation with respect to the coordinate system of the user instead of the coordinate system of the camera.

Currently there are several methods for calibrating a camera. These methods can be classified according to different criteria. For example, considering the resolution method can be classified into linear versus nonlinear or iterative. The methods used methods of solving linear systems of equations based on least squares. These methods get a transformation matrix which relates 3D points in the world their 2D projections in the image. In this case no parameters are calculated to model the distortion of the camera so the results are quite approximate, yet are easy to implement and very fast to run [35]. If, however, requires a camera model more complex, which include the distortions produced by the camera is necessary to minimize nonlinear index iteratively. Minimize the index usually include the distance between measured points on the image and the projected points obtained with the model of the camera. The advantage of these iterative methods is that any model can also be calculated and the accuracy of it increases with the number of iterations until it converges. However, they are much slower and need from a good approximation of the parameters to ensure the convergence. It is for this reason we use the results obtained by linear methods to start the search nonlinear parameters. With linear methods are calculated part of the set of parameters and then using iterative methods are improved these parameters and estimate the rest. This calibration performed in two steps can greatly reduce the number of iterations also ensuring the convergence of the iterative search of the parameters.

Another classification of calibration methods can be based on the outcome of it. An explicit calibration parameters obtained directly from the camera model [39], while an implicit transformation matrices are obtained that contain the set of all parameters. Although no one knows the exact value of any of the parameters, the results can be used to perform measurements and generate 3D coordinates in the image. Implicit methods are not suitable for modeling the camera and the parameters obtained do not correspond with the actual camera. Based on the parameters that form the model of camera calibration methods can also be classified into intrinsic and extrinsic. The calibration methods only obtain intrinsic parameters physical and optical of the camera. By contrast, the calculated position and extrinsic orientation of the camera in the scene.

Finally, considering the characteristics of the template used for calibration, there are methods that use templates 3D, 2D, 1D or do not use template. Methods using reference templates based calibration of the camera to establish a relationship between the known coordinates of the points in the template and the coordinates of these points in the image. For templates with a single 3D image of it is possible to calibrate. In this case the template is two or three orthogonal planes between them. As the points where planes take avoiding measurement errors of the coordinates of points on the template as it is assumed the same for all points on the same plane. On the contrary, this type of calibration requires expensive processing. If you use templates 2D is necessary to take several pictures of it from various positions or change the position and orientation of the template. It is not necessary to know the positions from which images are taken [40]. This method is more versatile as the development of the template is easily accomplished. The calibration methods are based on templates 1D very useful in the case of calibrating multiple camera systems. In the case of template-based methods use 3D or 2D, as it is necessary that they all see different points of the calibration template at the same time, it is difficult to establish a position for it except that the template is transparent. It is for this reason that the calibration method based on a template 1D attractive when calibrating a system with multiple cameras [41]. In some cases, template-based methods need to know the relationship between the plans to set more restrictions on the calibration process and to achieve results more accurate. Also the use of geometric properties from the scene of such lines to infinity, or elements such as straight lines or circles within the same, allow calibration without take measurements of the coordinates of points on the template. On the other hand, techniques that do not use any calibration object can be considered as template 0D and you only need to relate to a point in different images. Only by moving the camera in a static scene, the rigidity of the scene causes in general two constraints within the camera intrinsic parameters [42], [43]. So with several images of the same scene taken with the same intrinsic parameters, correspondences between three images are sufficient to calculate both intrinsic and extrinsic parameters. This rule allows your time to perform 3D reconstruction of a structure from several images of the same. In these cases, although a template is not necessary, it is necessary to calculate a large number of parameters, resulting in a rather complex mathematical problem. Due to the difficulty to start searching for self-calibration methods tend to be unstable [10], [26]. Finally, the summary of the different methods that exist for the calibration of cameras, it is also necessary to consider the family of algorithms that calculate the parameters that model lens distortion in the image without the use of calibration objects and thus without knowing any 3D structure. These methods are based on a projection that in ideal perspective, the camera turns straight lines in 3D space in straight lines within the 2D space corresponding to the image. Thereby strengthening the linearity of the parts of the image are curved due to the distortion of camera lens, we can estimate the distortion it produces [44]. There are methods that use epipolar and trilinear constraints among pairs and triplets of images respectively to estimate the radial distortion.

Based on the current state of calibration processes described so far, it is difficult to choose an efficient method to calibrate the camera in any situation. Tsai's method [45], is a classic calibration process based on measures the coordinates of points on a 3D template with respect to a fixed reference point. This method has been widely used in the last century. In comparison calibration methods developed between 1982 and 1998 by Salvi [46], Tsai's method shows better results in spite of that to get good results is an important qualification necessary data entry. By contrast, the method of Zhang [41], which is not included in the comparison of Salvi, represents a new era in the process of calibrating the camera. This method uses the coordinates of the points within a 2D flat template taking different pictures of it from different positions and orientations. This will combine the benefits of calibration methods based on measures of the coordinates of the template with the advantages of self-calibration which is not necessary to use template. This calibration mode is very flexible from the standpoint of both the camera and the template can be moved freely and it can take as many pictures as you want without having to perform measurements on the template. Sun [47] compared the method of Tsai to the method of Zhang. On the one hand the method of Tsai gets an accurate estimation of the parameters of the camera if the input data are contaminated with some noise. Given that it takes at least a hundred points in the template and the coordinates are to be referred to a fixed origin of coordinates is essential an appropriate template design calibration and a precise measurement of the coordinates of the points. Nevertheless, the possibility of errors in the measurements is high as confirmed by the experiments performed by Sun. By contrast, Zhang's calibration method based on 2D template that requires no special design of the template, nor as accurate measurement of the points of it. Sun performs experiments with a template made by hand and gets better results than the method of Tsai. Furthermore, the sensitivity of the calibration algorithm to errors in measurements can be improved by increasing the number of points in the template, simply printing a chessboard with more corners. The comparison results show the flexibility and adaptability of the calibration method of Zhang and can be performed at any scenario.

4.2.1 Calibration template 2D based

Zhang [105] proposed a calibration technique based on the observation of a flat template from various positions. This method is more versatile as the development of the template is easily accomplished. The camera can be shifted manually as it is not necessary to know the positions of the camera where the pictures were taken from the template. This makes it a very flexible technique as the previous method requires a more complicated design template in addition to knowing the exact positions of points within it. To calibrate the camera with this method is necessary to estimate the homographies of each of the images taken from the template. With estimadasse homographies calculated parameters of the camera. Before describing the method of camera calibration process is detailed estimate of the homographies.

4.2.2 Computation of homographies

In the previous section we discussed the estimation of the projection matrix from a staff of various levels so that contains points with coordinates in three dimensions. In case you want to estimate a homography from a 2-dimensional template, the reasoning is similar to that performed so far. The coordinates of a point $q_i = (u_i, v_i)$ in the image is calculated from the corresponding position in the scene $p_i = (x_i, y_i)$ and the elements that form the homography using the following expressions:

$$u_i = \frac{h_{11}x_i + \ h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + \ h_{33}}$$

$$v_i = \frac{h_{21}x_i + \ h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + \ h_{33}}$$

Equation 4.6

Rearranging these two equations for each known position of a point on the template $p_i = (x_i, y_i)$ and its corresponding in image $q_i = (u_i, v_i)$, is possible to obtain two equations with 9 unknowns, which correspond to the elements the homography:

$$u_i(h_{31}x_i + h_{32}y_i + h_{33}) = h_{11}x_i + h_{12}y_i + h_{13}$$

$$v_i(h_{31}x_i + h_{32}y_i + h_{33}) = h_{21}x_i + h_{22}y_i + h_{23}$$

Equation 4.7

If ***n*** points are available, we can obtain a linear system of ***2 ° N*** equations with 9 unknowns as follows:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -u_1 \cdot x_1 & -u_1 \cdot y_1 & -u_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -v_1 \cdot x_1 & -v_1 \cdot y_1 & -v_1 \\ .. & .. & .. & .. & .. & .. & .. & .. & .. \\ x_n & y_n & 1 & 0 & 0 & 0 & -u_n \cdot x_n & -u_n \cdot y_n & -u_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -v_n \cdot x_n & -v_n \cdot y_n & -v_n \end{bmatrix} \cdot \begin{bmatrix} h_{11} \\ h_{12} \\ .. \\ h_{32} \\ h_{33} \end{bmatrix} = 0$$

Equation 4.8

The system of linear equations can be expressed in matrix form as:

$$A \cdot h = 0$$

Equation 4.9

In this case, the matrix of dimension (2 ° n) x 9 contains the coordinates of the points in the template and its corresponding 3D positions of the projections in the image, all known. ***h*** is a vector containing all the elements of the homography ***H*** placed in the vector:

$$h = [h_{11} \quad h_{12} \quad h_{13} \quad h_{21} \quad h_{22} \quad h_{23} \quad h_{31} \quad h_{32} \quad h_{33}]^T$$

Equation 4.10

For the elements of the homography ***h***, since the solution ***h = 0*** has no interest, imposes the constraint **| h | = 1**. A vector h is solution of the equation 4.10, and any other vector **k • h** are equally valid, since ***h***, is defined with a scale factor.

Therefore, since no exact solution for solution for **A • h = 0**, we can minimize the norm **|A • h|** subject to the constraint **| h | = 1**. The solution h is the unit eigenvector matrix **A^T·A** associated with smaller eigenvalue. This vector could be obtained from the decomposition into eigenvalues of positive definite symmetric matrix $\mathbf{A}^T\cdot\mathbf{A}$.

4.2.3 Calibration process with 2D templates

The linear model of the camera to be estimated is described in the previous chapter, in which obtaining the coordinates of the image may be obtained by:

$$q_i = \lambda \cdot K \cdot [\mathrm{R} \quad \mathrm{t}] \cdot p = \lambda \cdot K \cdot [r_1 \quad r_2 \quad r_3 \quad t] \cdot p_i$$

$$\begin{bmatrix} w_i \cdot u_i \\ w_i \cdot v_i \\ w_i \end{bmatrix} = \lambda \cdot \begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

Equation 4.11

λ is a scale factor and the points are expressed in homogeneous coordinates. ***R*** and ***t*** represent the extrinsic camera parameters being ***r****ᵢ* the columns of the rotation matrix ***R***. ***K*** contains the intrinsic parameters, being α_u and α_v scale factors each of the axes of the image, u_0 and v_0 are the coordinates of main point of the image and γ is the parameter that represents the loss of orthogonality of the coordinate axes in the image.

If part of the calibration template is flat, we can assume that the points of the template are arranged so that its coordinate $z_w = 0$. In this case the model is reduced by the following expression:

$$q_i = \lambda \cdot K \cdot [R \quad t] \cdot p = \lambda \cdot K \cdot [r_1 \quad r_2 \quad r_3 \quad t] \cdot p_i$$

Equation 4.12

Now the initial model is transformed into a homography ***H*** relating the coordinates of the template flat stage with their counterparts in the image:

$$q_i = \lambda \cdot K \cdot [r_1 \quad r_2 \quad t] \cdot p_i = H \cdot p_i$$

Equation 4.13

This homography can be calculated by the method described in the previous section. If you separate the columns of the homography is obtained that:

$$[h_1 \quad h_2 \quad h_3] = \lambda \cdot K \cdot [r_1 \quad r_2 \quad t]$$

Equation 4.14

Translation vector ***t*** is from world coordinate system zero point to optical center vector $\boldsymbol{r_1}$,$\boldsymbol{r_2}$ is the image plane two coordinate axes in the world coordinate system's direction vector, obviusly ***t*** will not be located at the $\boldsymbol{r_1},\boldsymbol{r_2}$ plane, as a result of $\boldsymbol{r_1},\boldsymbol{r_2}$ orthogonal, therefore $\det([r_1 \quad r_2 \quad t]) \neq 0$, also $\det([A]) \neq 0$, therefore $\det([H]) \neq 0$. The computation of ***H*** causes between the actual image coordinate $m_i = [u \quad v]^T$ and the image coordinate $\widehat{m_i}'$ according to type equation 4.13, calculates the diverse smallest process. The objective function is:

$$\min \sum_i \|m_i - \widehat{m_i}\|^2$$

Equation 4.15

Figure 4.5 shows the iterative procedure to obtain the extrinsic and intrinsic parameters using a two-dimensional template.

- Place the template on a horizontal solid wall;
- Moves plane or camera to shot some template images from different angle;
- Detects characteristic point of the image;
- Obtains each image the unitary matrix ***H***;
- Computes camera's internal parameter by using matrix ***H*** extracted in the premise of the distortion factor being zero;
- Obtains a group of precision higher camera's internal parameter, simultaneously calculating each distortion factor.

Figure 4.5. Algorithm to obtain intrinsic parameters

4.2.2 Full automatic process of calibration

In this section we discuss a widespread methodology to locate points of interest on a template calibration through image processing methods. Checkerboards with black-and-white squares are most widely used because the easy sub-pixel detection algorithm for X-corners with high precision [Luc00a]. Traditional algorithm for detecting X-corners first estimates their pixel locations by standard corner detectors (such as Harris [Har01a] and Noble [Nob00a]), then the sub-pixel positions can be determined by fitting quadratic functions to the local intensity profile around the corners and computing their extremal points. The main shortcoming of this algorithm is that the fitting of local intensity surface complicate the detection process. Lucchese [] proposed a new simplified algorithm finding the extremal points by a morphological shrinking operation on the local intensity profile. This algorithm requires a preliminary interpolation of intensity over the 2 × 2 -pixel neighborhood of the detected corners, which means that shortening the interpolation interval will improve the precision of detection.

The more extended algorithm for detecting X-corners first finds their pixel positions by Harris detector based on a Hessian matrix looking for the auto-correlation matrix:

$$M = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 \otimes w & \left(\frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y}\right) \otimes w \\ \left(\frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y}\right) \otimes w & \left(\frac{\partial I}{\partial y}\right)^2 \otimes w \end{bmatrix}$$

Equation 4.16

where ***w*** is a Gauss smoothing operator. Harris corner detector is expressed as:

$$R = \det(M) - \lambda(trace(M))^2$$

Equation 4.17

The X-corner is just the local peak point of **R** . Considering the local intensity around one X-corner in the image which has been smoothed by a Gauss low pass filter, the 3D shape of the intensity profile is just like a saddle. The saddle point of this surface is just the X-corner to be detected. For each X-corner, a quadratic fitting of the local intensity profile can be obtained. The function can be expressed as:

$$F(x, y) = ax^2 + bxy + cy^2 + dx + ey + f$$

Equation 4.18

This function turns out to be a hyperbolic, so the position of the saddle point can be determined by calculating the intersection of the two lines as follows:

$$\begin{cases} 2ax + by + d = 0 \\ bx + 2cy + e = 0 \end{cases}$$

Equation 4.19

In general, this traditional algorithm allows for accuracies of the order of a few hundredths of a pixel [Luc00a], which can satisfy most applications of 3D machine vision. But the preliminary interpolation of intensity and the latter surface fitting aggravate the computation load of this algorithm, although this is not problem for us because the calibration process it is an off-line process, once time we have obtained a reliable set of intrinsic values this process is not used again.

Chapter 5

RGB-D, Visual advantage

RGB-D cameras are novel sensing systems that capture RGB images along with per-pixel depth information. In this chapter we introduce how such cameras can for our proposal, in the context building dense 3D maps, in order to rebuild human pose in indoor uncontrolled environments with actors. This novel technology has applications in robot navigation, manipulation, semantic mapping, and telepresence. We present RGB-D Mapping, a full 3D mapping system that utilizes a novel joint optimization algorithm combining visual features and shape-based alignment. Visual and depth information are also combined for view-based loop closure detection, followed by pose optimization to achieve globally consistent maps. One of the main problems in the field of computer vision is the variability of view to analyze any object. By introducing the depth values can reconstruct a three-dimensional map in which we can recognize the position of that object, and thus eliminate the possible error due to perspective. In this chapter we focus on the alignment procedure between the values of depth and the image captured by a camera RGB. Extended to analyze the methodology as well as their possible applications. In the next chapter will go into the section as a model the pose of a person based on the data set obtained through RGB-D.

5.1 RGB-D technology

Building rich 3D maps of environments is an important task for mobile robotics, with applications in navigation, manipulation, semantic mapping, and telepresence. Most 3D mapping systems contain three main components: first, the spatial alignment of consecutive data frames; second, the detection of loop closures; third, the globally consistent alignment of the complete data sequence. While 3D point clouds are extremely well suited for frame-to-frame alignment and for dense 3D reconstruction, they ignore valuable information contained in images. Color cameras, on the other hand, capture rich visual information and are becoming more and more the sensor of choice for loop closure detection [48, 49, 50]. However, it is extremely hard to extract dense depth from camera data alone, especially in indoor environments with very dark or sparsely textured areas.

RGB-D cameras are sensing systems that capture RGB images along with perpixel depth information. RGB-D cameras rely on either active stereo [51, 52] or time-of-flight sensing [53, 54] to generate depth estimates at a large number of pixels. While sensor systems with these capabilities have been custom-built for years, only now are they being packaged in form factors that make them attractive for research outside specialized computer vision groups. In fact, the key drivers for the most recent RGB-D camera systems are computer gaming and home entertainment applications.

RGB-D cameras allow the capture of reasonably accurate mid-resolution depth and appearance information at high data rates. In our work we use a camera developed by PrimeSense, which captures 640x480 registered image and depth points at 30 frames per second. This camera is equivalent to the visual sensors in the recently available Microsoft Kinect [55]. Fig. 5.1 shows an example frame observed with this RGB-D camera. As can be seen, the sensor provides dense depth estimates. However, RGB-D cameras have some important drawbacks with respect to 3D mapping: they provide depth only up to a limited distance (typically less than 5m), their depth estimates are very noisy and their field of view (≈60°) is far more constrained than that of the specialized cameras and laser scanners commonly used for 3D mapping (≈80°).

Figure 5.1 Kinect Device

RGB-D Mapping exploits the integration of shape and appearance information provided by these systems. Alignment between frames is computed by jointly optimizing over both appearance and shape matching.

5.2 RGB-D alignment procedure

The solution to the frame alignment problem strongly depends on the data being used. For 3D laser data, the iterated closest point (ICP) algorithm and variants thereof are popular techniques [56,57]. The ICP algorithm iterates between associating each point in one time frame to the closest point in the other frame and computing the rigid transformation that minimizes distance between the point pairs. The robustness of ICP in 3D has been improved by, e.g., incorporating point-toplane associations or point reflectance values [58].

Passive stereo systems can extract depth information for only a subset of feature points in each stereo pair. These feature points can then be aligned over consecutive frames using an optimization similar to a single iteration of ICP, with the additional advantage that appearance information can be used to solve the data association problem more robustly, typically via RANSAC [59]. Monocular SLAM and mapping based on unsorted image sets are similar to stereo SLAM in that sparse features are extracted from images to solve the correspondence problem. Projective geometry is used to define the spatial relationship between features [60], a much harder problem to solve than correspondence in ICP.

For the loop closure problem, most recent approaches to 3D mapping rely on fast image matching techniques [50]. Once a loop closure is detected, the new correspondence between data frames can be used as an additional constraint in the graph describing the spatial relationship between frames. Optimization of this pose graph results in a globally aligned set of frames .

While RGB-D Mapping follows the overall structure of recent 3D mapping techniques, it differs from existing approaches in the way it performs frame-to-frame matching. While pure laser-based ICP is extremely robust for the 3D point clouds collected by 3D laser scanning systems such as panning SICK scanners or 3D Velodyne scanners [61], RGB-D cameras provide depth and color information for a small field of view (60° in contrast to 180°) and with less depth precision (≈3cm at 3m depth). The limited field of view can cause problems due to a lack of spatial structure needed to constrain ICP alignments. There has been relatively little attention devoted to the problem of combining shape and visual information for scan alignment. While this approach provides excellent results, it is computationally expensive and does not scale to large 3D clouds.

A common addition to ICP is to augment each point in the two point clouds with additional attributes. The correspondence selection step acts in this higherdimensional space. This approach has been applied to point color, geometric descriptors, and point-wise reflectance values. In comparison, our algorithm uses rich visual features along with RANSAC verification to add fixed data associations into the ICP optimization. Additionally, the RANSAC associations act as an initialization for ICP, which is a local optimizer.

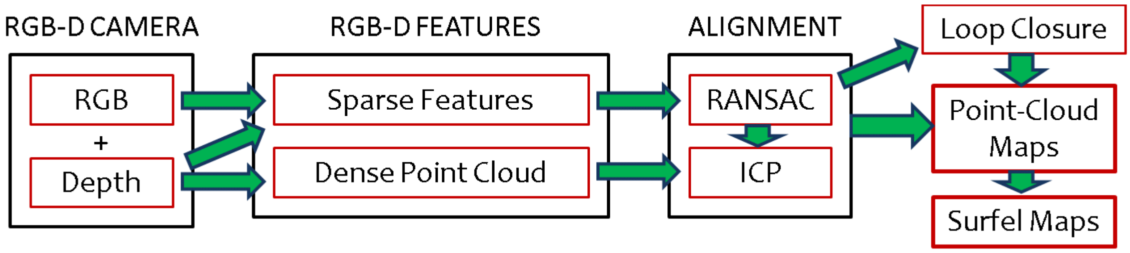


Figure 5.2 Block diagram of the alignment procedure

5.2.1 RGB-D Mapping

To align the current frame to the previous frame, the alignment step uses RGBDICP, our enhanced ICP algorithm that takes advantage of the combination of RGB and depth information. After this alignment step, the new frame is added to the dense 3D model. This step also updates the surfels used for visualization and occlusion reasoning. A parallel loop closure detection thread uses the sparse feature points to match the current frame against previous observations, taking spatial constraints into account. If a loop closure is detected, a constraint is added to the pose graph and a global alignment process is triggered.

In the Iterative Closest Point (ICP) algorithm [2], points in a source cloud $\boldsymbol{P_s}$ is matched with their nearest neighboring points in a target cloud P_t and a rigid transformation is found by minimizing the n-D error between associated points. This transformation may change the nearest neighbors for points in $\boldsymbol{P_s}$, so the two steps of association and optimization are alternated until convergence. ICP has been shown to be effective when the two clouds are already nearly aligned. Otherwise, the unknown data association between $\boldsymbol{P_s}$ and $\boldsymbol{P_t}$ can lead to convergence at an incorrect local minimum. Alignment of images, by contrast, is typically done using sparse feature-point matching. A key advantage of visual features is that they can provide alignments without requiring initialization. One widely used feature detector and descriptor is the Scale Invariant Feature Transform (SIFT) [62]. Though feature descriptors are very distinctive, they must be matched heuristically and there can be false matches selected. The RANSAC algorithm is often used to determine a subset of feature pairs corresponding to a consistent rigid transformation. However, in 2D this problem is not fully constrained due to the scale indeterminacy.

```
RGBD-ICP ($P_s$,$P_t$):

1. $F_{source}$=Extract_RGB_features($P_s$)
2. $F_{target}$=Extract_RGB_features($P_t$)
3. (t*,$A_f$) = Perform_RANSAC_Alignment($F_{source}$, $F_{target}$)
4. repeat
5.   $A_d$ =Compute_Closest_Point(t*,$P_s$,$P_t$)
6. until (Error_Change(t*)≤θ) or (maxIter reached)
7. return t*
```

Figure 5.3 Algorithm of align RGB-D with ICP

Since we have RGB-D frames, we can fuse these two approaches to exploit the advantages of each. It takes as input a source RGB-D frame, $\boldsymbol{P_s}$, and a target frame, $\boldsymbol{P_t}$. Steps 1 and 2 extract sparse visual features from the two frames and associate them with their corresponding depth values to generate feature points in 3D. These steps can be implemented with arbitrary visual features. Step 3 uses RANSAC to find the best rigid transformation, **t***, between these feature sets. Perform RANSAC Alignment does this by first finding matching features between the two frames. It then repeatedly samples three pairs of feature points, determines the optimal transformation for this sample, and counts the number of inliers among the remaining 3D feature points. The function also returns a set of associations $\boldsymbol{A_f}$ containing the feature pairs that generated the best transformation.

Steps 4 through 6 perform the main ICP loop. Step 5 determines the associations $\boldsymbol{A_d}$ between the points in the dense point cloud. This is done by transforming the 3D points in the source cloud, $\boldsymbol{P_s}$, using the current transformation ***t***. In the first iteration, ***t*** is initialized by the visual RANSAC transformation, which allows RGBD-ICP to match frames without any knowledge of their relative pose (if enough visual features are present). For each point in $\boldsymbol{P_s}$, Step 5 then determines the nearest point in the target cloud $\boldsymbol{P_t}$. While it is possible to compute associations between points based on a combination of Euclidean distance, color difference, and shape difference, we found Euclidean distance along with a fast kd-tree search to be sufficient in most cases. It minimizes the alignment error of both the visual feature associations and the dense point associations. The first part of the error function measures average distances for the visually associated feature points, and the second part compute a similar error term for the dense point associations.

The loop exits after the error no longer decreases significantly or a maximum number of iterations is reached. Otherwise, the dense data associations are recomputed using the most recent transformation. Note that feature point data associations are not recomputed after the RANSAC procedure. This avoids that the dense ICP components might cause the point clouds to drift apart, which can happen in underconstrained cases such as large flat walls.

5.2.1 Loop closure detection

Alignment between successive frames is a good method for tracking the camera position over moderate distances. However, errors in alignment between a particular pair of frames, and noise and quantization in depth values, cause the estimation of camera position to drift over time, leading to inaccuracies in the map. This is most noticeable when the camera follows a long path, eventually returning to a location previously visited. The cumulative error in frame alignment results in a map that has two representations of the same region in different locations. This is known as the loop closure problem, and our solution to it has two parts. First, loop closure detection is needed to recognize when the camera has returned to a previously visited location. Second, the map must be corrected to merge duplicate regions. Our overall strategy is to represent constraints between frames with a graph structure, with edges between frames corresponding to geometric constraints. The relative transformations from the alignment of sequential frames give us some constraints, so without any loop closure, the graph consists of a linear chain. Loop closures are represented as constraints between frames that are not temporally adjacent.

To keep the graph relatively sparse we define keyframes, which are a subset of the aligned frames. We determine keyframes based on visual overlap, adapting the density of keyframes to camera motion and local appearance. After we align a frame F, we reuse the SIFT features to find a rigid transformation with the most recent keyframe, using the same RANSAC procedure defined for frame-to-frame align [63].

Each time we create a new keyframe we attempt to detect a loop closure with each previous keyframe. A closure is detected if enough geometrically consistent 3D feature point matches are recovered by RANSAC, and if so, we add an edge to the graph representing this newly discovered constraint. For this stage we modify RGBD-ICP slightly to return no-matches if no RANSAC match is found with sufficient inliers, so that the same algorithm that performs frame-to-frame matching also performs loop closure detection and initializes pose-graph edges. We only perform the RANSAC check with keyframes that are within a small distance of our current position estimate.

5.2.2 Surfel representation

Considering that each frame from the RGB-D camera gives us roughly 250,000 points, it is necessary to create a more concise representation of the map. One option is to downsample the clouds. However, it is more appealing to incorporate all the information from each frame into a concise representation for visualization. One method for doing this is surfels [64,65]. A surfel consists of a location, a surface orientation, a patch size and a color. Surfels store a measure of confidence, which is increased through being seen from multiple angles over time. Surfels with low confidence are removed from the representation. Because surfels have a notion of size (obtained initially from the depth of the original point in the RGB-D frame), we can reason about occlusion, so if an existing surfel is seen through too often, it can be removed. Based on the estimated normals within each RGB-D frame, the surfel normal directions can be updated as well. We can also wait to add surfels until their normal is pointed (within some angle) towards the camera position, which leads to a more accurate recovery of the surfel size. The color of a surfel is determined from the RGB-D frame most aligned with the normal direction.

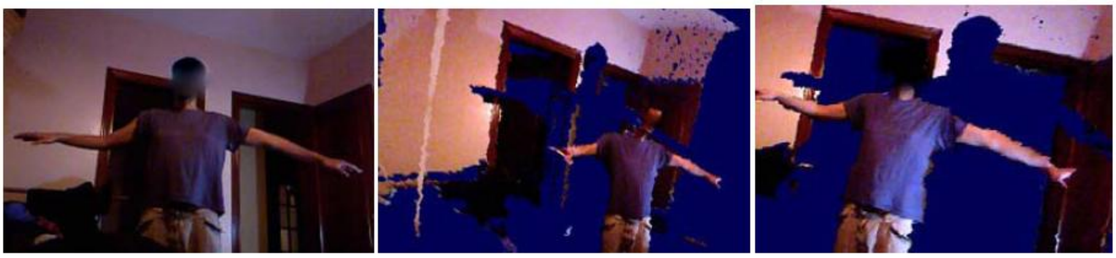


Figure 5.4 Visualization of a RGB-D alignment and its 3D mapping

Chapter 6

Modeling Human Pose

In this chapter we will try to model the human pose consistently for at a later stage, performing recognition procedures. Visual analysis of human motion is currently one of the most active research topics in Computer Vision. Several segmentation techniques for body pose recovery have been recently presented, allowing for better generalization of gesture recognition systems. The evaluation of human behavior patterns in different environments has been a problem studied in social and cognitive sciences, but now it is raised as a challenging approach to computer science due to the complexity of data extraction and its analysis. From the point of view of data acquisition, many methodologies treat images captured by visible light cameras. Computer Vision is then used to detect, describe, and learn visual features [62]. The main difficulties of visual descriptors on RGB data is the discrimination of shapes, textures, background objects, changing in lighting conditions and viewpoint. On the other hand, depth information is invariant to color, texture and lighting objects, making it easier to differentiate between the background and the foreground object. The first systems for depth estimation were expensive and difficult to manage in practice. Earlier research used stereo cameras to estimate human poses or perform human tracking. In the past few years, some research has focused on the use of time-of-flight range cameras (TOF).

Our proposal to create a representative model of the body pose is based on two aspects. Our first proposal is based on a traditional methodology to get the articulated model which represents the pose based on the manual placement of markers. The markers are arranged in a visual features where our system will be able to distinguish them and interpret the static pose that reflects the actor. The second proposal is based on creating a skeletal model based on 15 joints, which are performed automatically through obtaining approximate silhouette poses learned by examining the depth map. In this second proposal, the process is completely automatic and non-invasive and should not be placing any kind of marker to the actor.

6.1 Getting the model articulated by landmarks

In this part, is presented a fully-automatic system that is able to segment the human body as well as the markers distributed among the human body. The system is composed by two main modules. The first module requires the user to make and show an image with the color model of the markers and to calibrate the sensors in a simple way. Once the system is adapted, in the second module the user only needs to put the markers to the subject and make a photo. The system then automatically computes a depth map, segments human body and markers, and processes the data so that it is automatically formatted and displayed to the physician in order to give objective support for pose analysis.

The architecture of the system is shown in Figure 6.1. From a hardware point of view, the architecture uses a RBG video camera and infrared laser. The details of the sensors are described in the results section. From a software and usability point of view, the system can be split into two main modules: system adaptation and automatic analysis. Next, we describe each module in detail.

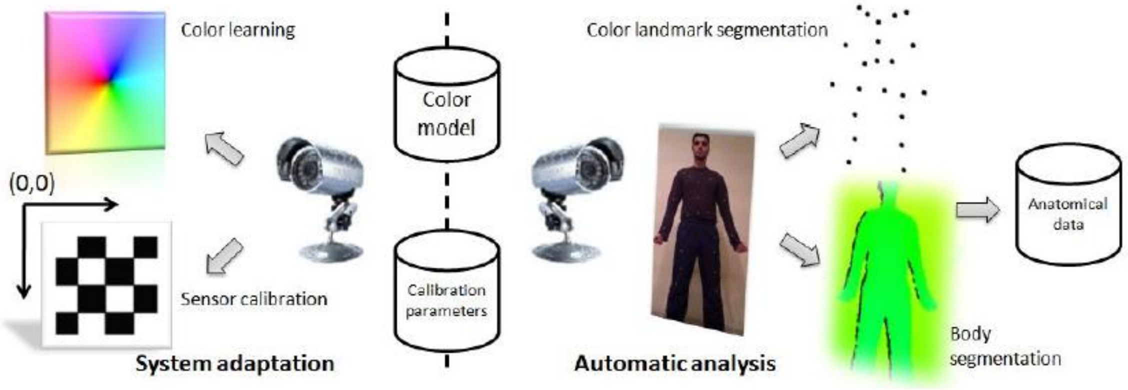


Figure 6.1 System architecture

In order to obtain a fully-automatic system able to segment the human body and detect markers, color learning and sensor calibration are first required (left part of Figure 6.1). The explanation of the calibration will be omitted, because we use the calibration template methodology explained in chapter two-dimensional calibration.

6.1.1 Color learning

In order to learn a color model, an interactive interface has been designed so that the user can load an image where a model of the target markers has been previously captured. The user can then click on a pixel i of the image with the desired color. Then, a region growing procedure [5] starting at $\mathcal{V} = i$ is applied in the following way:

$$\mathcal{V} = \mathcal{V} \cup \forall j | j \epsilon N_V, d(x_j, x_i) < \theta_c$$

Equation 6.1

where N_V is the set containing all the neighbor pixels to pixels contained in V, function $d(x_j, x_i)$ measures the color difference in CIELAB color representation x_j and x_i of pixels j and i, respectively, and θ_c is a sensitivity threshold parameter. Previous equation is iteratively performed until no updated is obtained over $\mathcal{V}$. The sensitivity parameter is changed online by the user by means of an interactive toolbar. In this way, the user can observe the set of selected pixels $\mathcal{V}$ and include them to the color model, avoiding the inclusion of non-representative color pixels. The CIELAB color model of pixels $\mathcal{V}$ is then clustered into ***k*** clusters using *k*-means algorithm. The clustering is performed in the following way:

$$S_i^t = \{j : \|x_j - m_i^t\| \leq \|x_j - m_{i*}^t\| \forall i * \epsilon[1,..,j]\}$$

Equation 6.2

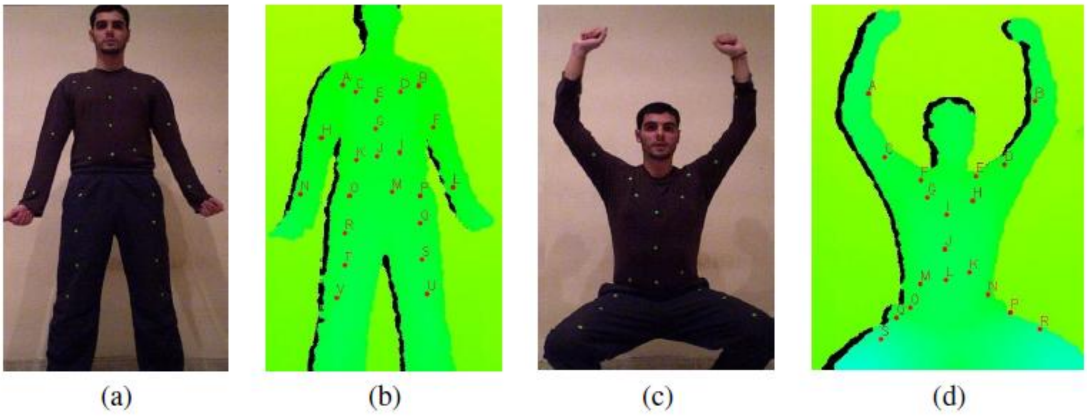
where S_i^t contains the pixels for cluster *i* at iteration *t* and m_i^t is the mean value of cluster *i*. After applying previous equations, new means (centroid) **m** are estimated as follows,

$$m_i^{t+1} = \frac{1}{|S_i^t|} \sum_{j \epsilon S_i^t} x_j$$

Equation 6.3

and the procedure is performed until no updated is obtained on *S*. The *k* centroid **m** obtained by running *k*-means correspond to our learned color model.

Given the region of the image corresponding to the automatic body segmentation, the color model **m** is used within this region is order to look for color markers. This task is performed automatically by comparing each pixel value in CIELAB representation with the centroid obtained by *k*-means clustering using a sensitivity threshold θ_k. Since the color segmentation is only performed within the body region, possible false positive detections are avoided. Moreover, we performed a postfiltering by means of mathematical morphology in order to remove noise pixels and join mark pixels [9]. From the estimated final blobs, its center of coordinates in then computed. Examples of automatically labeling of markers are shown in Figure 6.2(b) and (d) for input images (a) and (c), respectively.



(a) (b) (c) (d)

Figure 6.2 Example of marker recovery

If a form is described by **n** points on a dimension **d**, represent the way to **n$^{\mathbf{d}}$** vector formed by the concatenation of the individual position of landmarks. For example, on a 2-D image we can represent **n** landmarks on the feature vector **x** as:

$$x = (x_1, \dots, x_n, y_1, \dots, y_n)^T$$

Equation 6.4

6.1.1.1 Experimental results

In order to test the reliability of the first prototype of the application, we captured a set of data corresponding to the following specifications: four different corporal configurations and ten different markers configurations. Since we used five subjects for the experiment, it represents a total of 200 analysis. Some samples are shown in Figure 3(a) and (c), respectively. For these experiments we used green markers of 8*mm* of diameter, and fixed the system sensors at a 2*m* of distance from the subject. From this initial configuration, we captured one image and used the calibration interface in order to obtain the sensor parameters and perform color region growing to learn the color model. Then, automatic analysis on the commented data is performed. In order to see the scalability of the application in different environments, we also slightly changed the illumination conditions of the inner environment where we performed the experiments. The output of the system is a formatted excel file where the angle among all possible triplets of markers and the distance among all pairs of markers are shown.

The video data uses 8 bits VGA resolution at 15Hz, and we capture frames at 1280x1024 pixels. The infrared laser in combined with a monochrome CMOS sensor that captures the 3D data from the environment. We tested the laser sensitivity in a range between 1,2 and 3,5 meters, with successful results. The angular scope of the sensor is 57 and 43 degrees in horizontal and vertical, respectively. The software is implemented in C and uses OpenCV library [66].

Number of markers	mm	Degree
76,3%	2,3±1,2	3,7±2,4

Table 6.1 Results of experiments measured in percentage of detected markers, deviation to real distance, and deviation to real angle including confidence interval, respectively

The percentage of successfully detected markers, mean deviation in *mm* and degrees as well as their confidence interval at 95% are shown in Table 6.1. One can see that though the high distance among the subject and the sensor and the low diameter of the markers, the system is able to automatically detect almost all the markers, and obtain very accurate estimations of angles and distance compared to the ground truth data.

6.1.1.2 System enhancements

Due to low system performance regarding the detection of markers, has decided to change the marker type. This poor performance is due to different lighting conditions, color and uniformity of the markers caused by shadows and lighting. The new proposal each marker consists of a rechargeable battery button IR2032 20 mm in diameter with a corresponding holder, a microswitch and a 3mm LED cool white, 6000-8000mcd, 3V and 20mW. This assembly enables high brightness lasts up to 2 hours. In order to avoid having to put up the stage lighting is incorporated into the multi-sensor system, externally, a digital infrared 850nm filter. This configuration allows the capture of LED markers instantly, because infrared light is highly distinguishable from the rest of stage. These system elements are shown in Figure 6.3. In the Figure 6.4 we can see a real test of the system with the improvement.

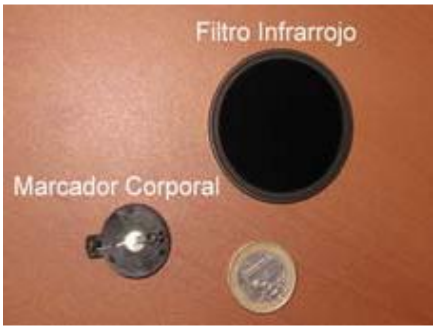


Figure 6.3 Picture with LED markers and cut-off IR filter

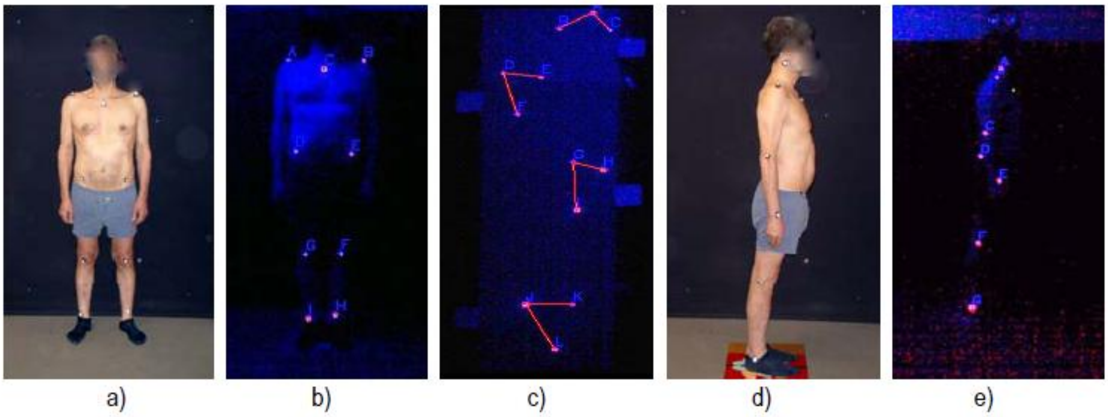


Figure 6.4 Example of LED marker recovery

This section will not be commenting on the experiments and results with the improvement, because part of the chapter on applications, specifically the implementation ADIBAS posture. In this section details the experiments and results.

6.1.2 Constitution of the skeletal model

Once obtained the set of markers, we characterize these as a vector of characteristics and then to treat them and distinguish them. To do this we must develop and manage such markers to obtain a consistent set capable of representing any articulated model, ie the human pose. There are numerous configurations to characterize positional skeletal model, we will try to render the model shown in Figure 6.5.

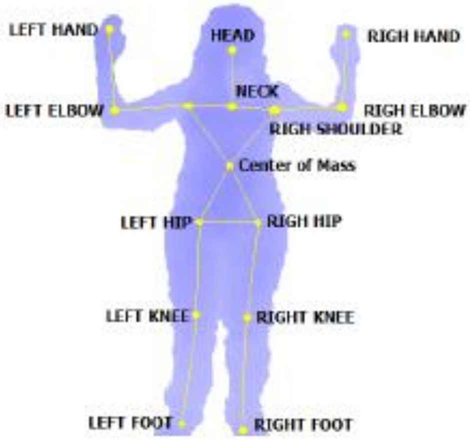


Figure 6.5. Skeletal model formed by 15 joints

The system described should make a reliable reconstruction of the vector of features obtained by the vector into the skeletal model. The system described should make a reliable reconstruction of the vector of features obtained by the vector into the skeletal model. This reconstruction of the feature vector formed by the position of the markers will be able to represent multiple combinations on which you can configure the human body pose. This ordering and indexing the vector of features will support the process of classifying the pose. For this purpose we have examined two algorithms capable of reconstructing the shape formed by the markers, they are Shape Context and Active Shape Models.

6.1.2.1 Shape Context

Shape context [68] is the term given by Serge Belongie and Jitendra Malik to the feature descriptor they first proposed in their paper "Matching with Shape Contexts" in 2000". The algorithm is mainly based on the following premises:

- Solve the correspondence problem between the two shapes. In our context, the source shape is a vector obtained by the position of the feature vector regarding of the markers, and how destiny is a predetermined form in our database. In practical purposes we use a simple initial form, as a calibration phase. Once we have rebuilt our vector of features in a skeletal model, the detection of the markers is still running shot by shot, keeping the ratio of the skeletal model.
- Use the correspondences to estimate an aligning transform.
- Compute the distance between the two shapes as a sum of matching errors between corresponding points together with a term measuring the magnitude of the aligning transformation.

The shape context is intended to be a way of describing shapes that allows for measuring shape similarity and the recovering of point correspondences. The basic idea is to pick ***n*** points on the joints of a shape. For each point $\boldsymbol{p_i}$ on the shape, consider the ***n* − 1** vectors obtained by connecting p_i to all other points. The set of all these vectors is a rich description of the shape localized at that point but is far too detailed. The key idea is that the distribution over relative positions is a robust, compact, and highly discriminative descriptor. So, for the point $\boldsymbol{p_i}$, the coarse histogram of the relative coordinates of the remaining ***n* – 1** points is defined to be the shape context of p_i. The bins are normally taken to be uniform in log-polar space.

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \epsilon \; bin(k)\}.$$

Equation 6.5

In order for a feature descriptor to be useful, it needs to have certain invariances. In particular it needs to be invariant to translation, scale, small perturbations, and depending on application rotation. Translational invariance come naturally to shape context. Scale invariance is obtained by normalizing all radial distances by the mean distance α between all the point pairs in the shape although the median distance can also be used. Shape contexts are empirically demonstrated to be robust to deformations, noise, and outliers using synthetic point set matching experiments.

Shape Context Methodology

1. Randomly select a set of joints of a known silhouette and another set of joints on an unknown silhouette.
2. Compute the shape context of each point found in step 1.
3. Match each joint from the known silhouette to a point on an unknown silhouette. To minimize the cost of matching, first choose a transformation (e.g. affine, thin plate spline, etc) that warps the edges of the known shape to the unknown (essentially aligning the two shapes). Then select the point on the unknown shape that most closely corresponds to each warped point on the known shape.
4. Calculate the "shape distance" between each pair of points on the two shapes. Use a weighted sum of the shape context distance, the image appearance distance, and the bending energy (a measure of how much transformation is required to bring the two shapes into alignment).
5. To identify the unknown shape, use a nearest-neighbor classifier to compare its shape distance to shape distances of known objects.

Figure 6.6 Shape context procedure

6.1.2.2 Active Shape Models

Active shape models [67] (ASMs) are statistical models of the shape of objects which iteratively deform to fit to an example of the object in a new image, developed by Tim Cootes and Chris Taylor in 1995. The shapes are constrained by the PDM (point distribution model). Statistical Shape Model to vary only in ways seen in a training set of labelled examples. The shape of an object is represented by a set of points (controlled by the shape model). The ASM algorithm aims to match the model to a new image. It works by alternating the following steps:

- Look in the image around each point for a better position for that point.
- Update the model parameters to best match to these new found positions.

Given a set of s training images, generate **s** vectors $\mathbf{x}_j$ (produced by Equation 6.4). Before statistical analysis of these vectors is very important that these forms, now represented by a vector, are all positioned on a same origin of coordinates. This process is called alignment.

The technique to position all forms of the same origin of coordinates will be the Procrustes Analysis [ref 14 app_models.pdf]. This technique aligns each shape in order to minimize the amount of distance each way about how average:

$$D = \sum |x_i - \bar{x}|^2$$

Equation 6.6

Where $\bar{x}$ is the feature vector relating to the possession of origin or calibration, and x_i the target feature vector. The operations to be carried out during the alignment process of the form passed in its final distribution. Different approaches to alignment can produce different distributions of aligned forms. Our goal is to maintain a compact layout of forms, with minimal non-linearization.

So far the model that we have vector contains a set of landmarks aligned. These vectors form a distribution on the $\mathbf{n}^{\mathbf{d}}$ dimensional space which is projected. If we design new forms on this site, and are positioned similarly to all learning, we can decide if the new examples are plausible or not. To examine the constitution of the form will look for a model to parameterize the form:

$$X = M(b)$$

Equation 6.7

Where **b** is a vector with the model parameters. Different models produce different vectors, **x**. Creating a distribution of parameters in a model, **p**, would be able to limit the scope of the new objects. These projections could play a classification by examining the similarity of the training set with new items.

To simplify the problem and improve the processing of the data, we reduce the dimensionality $\mathbf{n}^{\mathbf{d}}$ coming of vectors containing the landmarks. A very effective solution is to process the algorithm Principal Component Analysis (PCA) to the form of joint learning. Applying PCA on the data form a cloud of points projected information to each form on a space of reduced dimension. The above process is detailed in the following steps:

1. Find the center of mass of the information in the learning set size **s**.

$$\bar{x} = \frac{1}{s}\sum_{i=1}^{s} x_i$$

Equation 6.8.

2. Calculating the covariance of the information in the training set.

$$S = \frac{1}{s-1}\sum_{i=1}^{s}(x_i - \bar{x})(x_i - \bar{x})^T$$

Equation 6.9

3. Obtaining the eigenvectors, **φ** and their corresponding eigenvalues **λ**$_i$. Reordering of eigenvalues in descending order, resulting in the reorganization of the eigenvectors.

4. Choose the number of eigenvectors to be used according to the percentage of representation of the eigenvalues. It only uses the number of eigenvectors that represent 98% of the information through the eigenvalues.

If **φ** contains t eigenvectors corresponding to largest eigenvalues, then we can approximate the projection x of a new training set to $\bar{x}$ with the following expression:

$$x \cong \bar{x} + \varphi \mathrm{b}$$

Equation 6.10

Where **b** is a vector of **t** dimension given by:

$$b = \varphi^{\mathrm{T}}(\mathrm{x} - \bar{\mathrm{x}})$$

Equation 6.11

The vector b defines a set of parameters that define a strain belonging to the model of learning. To determine whether a model is plausible when we design the shape of the new image on all forms of learning, we obtain a vector b with the parameters of the form. We will create $\boldsymbol{p}(\boldsymbol{b})$ as an estimator of the distribution of all forms of learning. Decide whether this new form is plausible if $p(b) \geq p_t$, p_t be an average value of all distributions of learning chosen arbitrarily.

Once we have projected the new image on the learned model, and therefore, we have obtained have obtained the vector b which form the parameters that correspond to the shape. Before, once the proposed new form, it appears plausible to consider the problem this way.

We'll change the vector **b** by **b'**, which will form the parameters of an image, **x'**, plausible of training set close to **x**:

$$b' = \varphi^{\mathrm{T}}(\mathrm{x}^{'} - \bar{\mathrm{x}})$$

Equation 6.12

Once we have a first approximation of the form, so plausible, we initiated the process of convergence and thus adapt to new parts of the image. An example of the trained model is described by the parameters of **b**, combined with transformations such as translation **(X_t, Y_t)**, rotation **(θ)** and scale **(s)**.

The position of points on the model image, **x**, is given by:

$$x = T_{X_tY_t,s,\theta}(\bar{x} + \varphi\mathrm{b})$$

Equation 6.13

If you wanted an example of the model converge to the point (x, y) we would:

$$T_{X_tY_t,s,\theta}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X_t \\ Y_t \end{pmatrix} + \begin{pmatrix} s\cos\theta & s\sin\theta \\ -s\ \sin\theta & s\cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

Equation 6.14

If we want to find a correspondence between such a model learned on the set of points of the new shape, **Y**, we find an expression to be solved by minimizing:

$$\left|Y - T_{X_tY_t,s,\theta}(\bar{x} + \varphi\mathrm{b})\ \right|^2$$

Equation 6.15

Diagram of the training process:

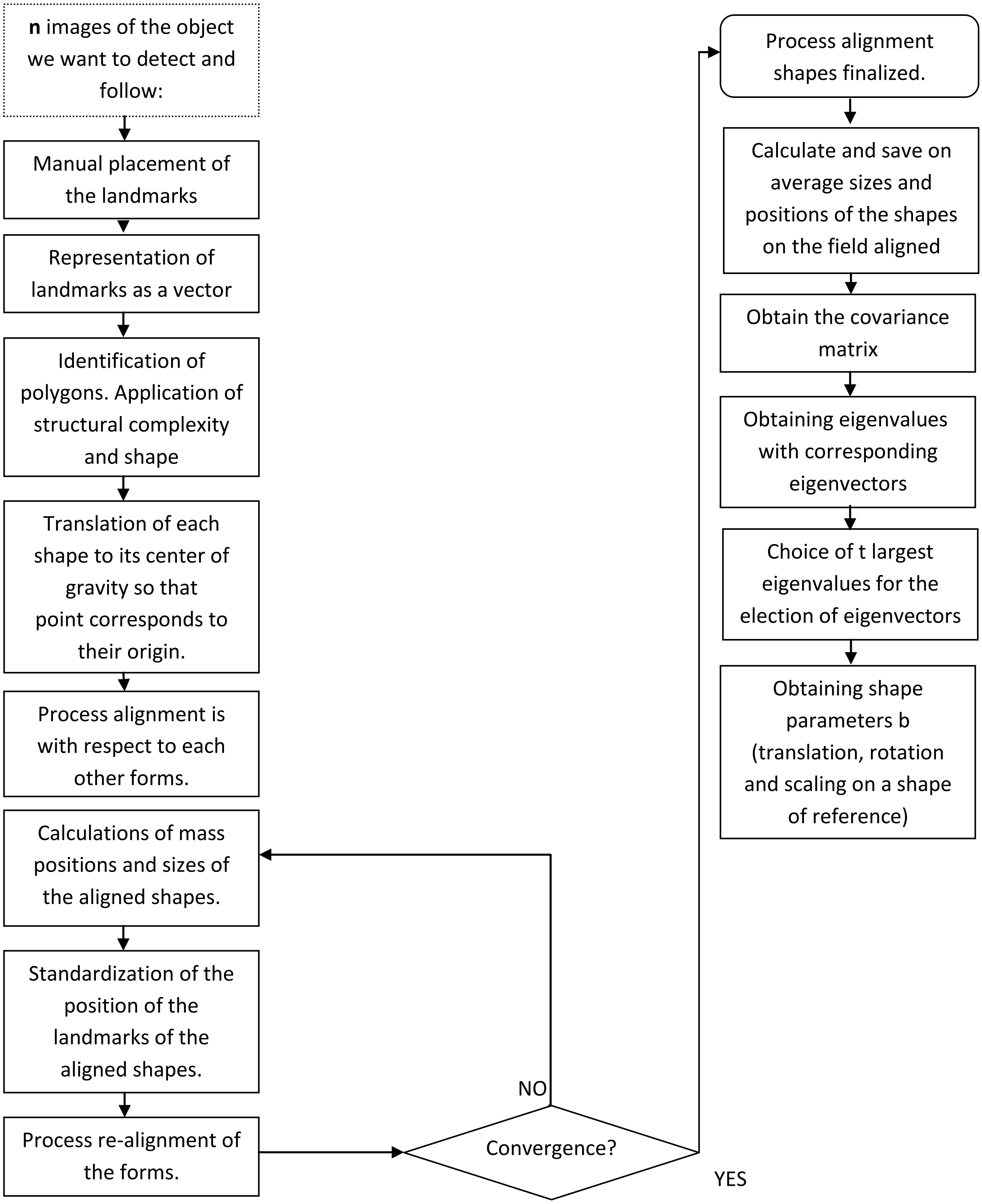
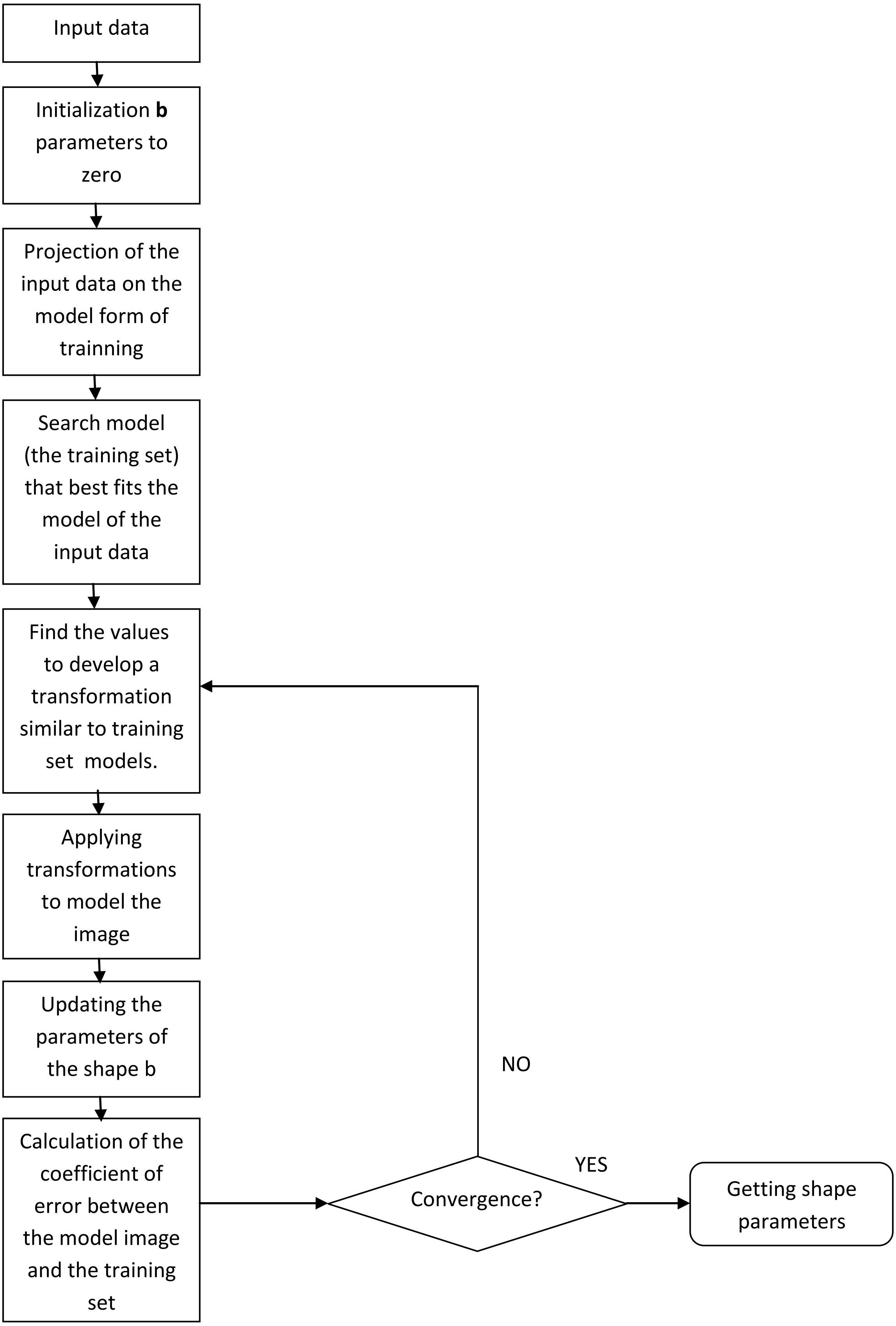

n images of the object we want to detect and follow:
Manual placement of the landmarks
Representation of landmarks as a vector
Identification of polygons. Application of structural complexity and shape
Translation of each shape to its center of gravity so that point corresponds to their origin.
Process alignment is with respect to each other forms.
Calculations of mass positions and sizes of the aligned shapes.
Standardization of the position of the landmarks of the aligned shapes.
Process re-alignment of the forms.
Convergence?
NO
YES
Process alignment shapes finalized.
Calculate and save on average sizes and positions of the shapes on the field aligned
Obtain the covariance matrix
Obtaining eigenvalues with corresponding eigenvectors
Choice of t largest eigenvalues for the election of eigenvectors
Obtaining shape parameters b (translation, rotation and scaling on a shape of reference)

Diagram of fitting


Input data
Initialization **b** parameters to zero
Projection of the input data on the model form of trainning
Search model (the training set) that best fits the model of the input data
Find the values to develop a transformation similar to training set models.
Applying transformations to model the image
Updating the parameters of the shape b
Calculation of the coefficient of error between the model image and the training set
Convergence?
NO
YES
Getting shape parameters

6.1.3 Experiments for the reconstruction of the skeleton

In this section we are going to expose a experiment in order to analyze the skeletal model obtained by both methods.

The video data uses 8 bits VGA resolution at 15Hz, and we capture frames at 640x480 pixels. The infrared laser in combined with a monochrome CMOS sensor that captures the 3D data from the environment. We tested the laser sensitivity in a range between 1,2 and 3,5 meters. The angular scope of the sensor is 57 and 43 degrees in horizontal and vertical, respectively. Had been implemented in C/C++ and uses OpenCV library [10].

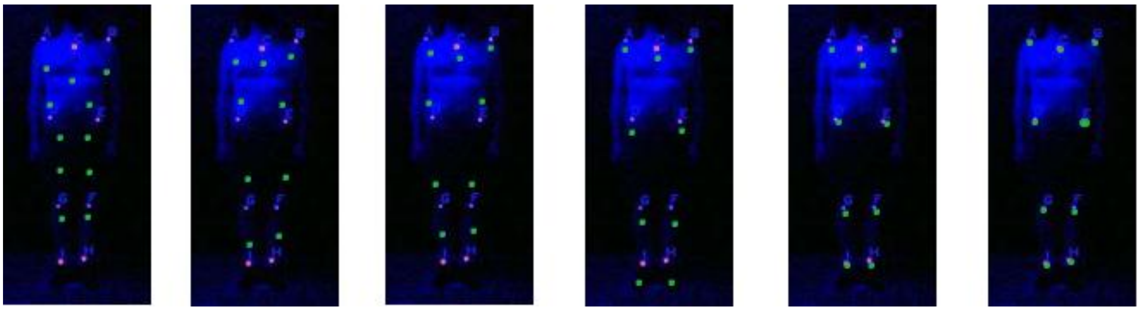


Figure 6.7 Convergence example with Shape Context

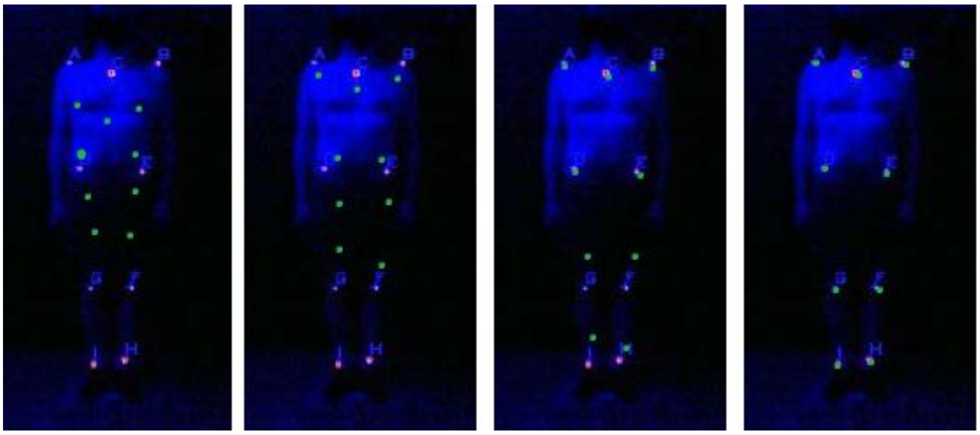


Figure 6.8 Convergence example with Active Shape Models

As seen in the previous figures the results are sufficiently accurate to use this methodology as long as the pose is simple enough to compare. Another important feature to note is its low computational complexity, obtaining results in a time less than 300ms in the case of Active Shape and less than 1500ms in the case of Shape Context (shape formed by nine points). Shape Context, the high computational cost limits its usefulness, as they would be launching it frame by frame.

6.2 Getting the model articulated using depth maps

In this section is presented a generic framework for object segmentation using depth maps based on Random Forest and Graph-cuts theory, and apply it to the segmentation of human limbs. First, from a set of random depth features, Random Forest is used to infer a set of label probabilities for each data sample. This vector of probabilities is used as unary term in α-expansion Graph-cuts algorithm. Moreover, depth of neighbor data points are used as boundary potentials. Results on a new multi-label human depth data set show high performance in terms of segmentation overlapping of the novel methodology compared to classical approaches.

Many researchers have obtained their first results in the field of human motion capture using this technology. In particular, Shotton et al. [12] present one of the greatest advances in the extraction of the human body pose from depth images, that also forms the core of the Kinect human recognition framework. The method is based on inferring pixel label probabilities through Random Forest (RF), using mean shift to estimate human joints, and representing the body in skeletal form. Other recent work uses the skeletal model in conjunction with computer vision techniques to detect complex poses in situations where there are many interacting actors [69].

In this paper we present a generic framework for object segmentation using depth maps based on RF and Graphcuts theory (GC), and apply it to the segmentation of human limbs. RF is used to infer a set of probabilities for each data sample, each one indicating the probability of a pixel to belong to a particular label. Then, this vector of probabilities is used as unary term in the ***α*-expansion** GC algorithm. Moreover, depth of neighbor data points are used as boundary potentials. As a result, we obtain a globally optimal segmentation of depth images based on the defined energy terms. The use of GC theory has been recently applied to the problem of image segmentation, obtaining successful results [70]. Our method is evaluated on a 3D data set designed in our lab, obtaining higher segmentation accuracy compared to classical RF approach.

6.2.1 Framework description

The depth-image based approach suggested in [12] interprets the complex pose estimation task as an object classification problem, by evaluating each depth pixel affiliation with a body part label, i.e. the probability for representing that body part. The pose recognition phase is addressed by re-projecting the pixel classification results and inferring the 3D positions of several skeletal joints using RF and mean-shift algorithms. The work of [12] shows a number of achievements and improvements over previous work, as the randomized decision forest classifier of T decision trees applied on simple and computationally efficient depth

features. Our goal is to extend the work of [12] and combine it with a general segmentation optimization procedure to define a globally optimum segmentation of objects in depth images.

As a case of study, we segment pixels belonging to the following seven body parts1: LU/LW/RU/RW arm, L/R hand, and torso (from Left, Right, Upper, and lower, respectively). The pipeline of the segmentation framework is illustrated in Figure 6.9.

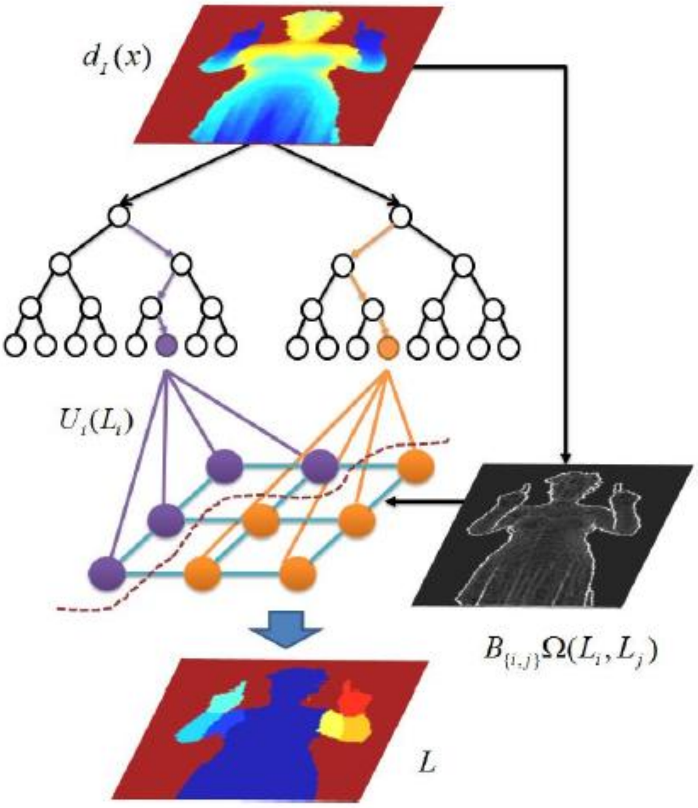


Figure 6.9

6.2.2 Random Forest

Considering a priori segmented human body from the background in a training set, the procedure for training a randomized decision tree t is formulated over the definition of a depth comparison feature as follows:

$$f_\theta(I,x) = \mathrm{d_I}\left(\mathrm{x}+\frac{\mathrm{u}}{\mathrm{d_I(x)}}\right) - \mathrm{d_I}\left(\mathrm{x}+\frac{\mathrm{v}}{\mathrm{d_I(x)}}\right)$$

Equation 6.16

where $\mathrm{d_I}(\mathrm{x})$ is the depth at pixel **x** in image *I*, $\theta = (u, v)$, and $u, v \in \mathrm{R}^2$ is a pair of offsets. The offset normalization ensures depth invariance. The tree training over a set of ground truth images runs through the following steps:

1. Random selection of a set of node splitting criteria (θ, τ) , where θ = (u, v), and $\tau \subset \mathrm{R}$ is a set of splitting thresholds for each θ.

2. Definition of a set of pixels Q = {(I, x)} over a unique set of training images (per tree) by random selection of a fixed number of uniformly distributed pixels for each image. Q constitutes the set of pixels at the root node of the tree.

3. Estimation of the best splitting criteria $\varphi*$ at the current node such that the information gain of partitioning the original set of pixels Q into left and right subsets is maximum. The partitioning decision is taken per pixel so that:

$$Q_{left}(\varphi) = \{\, (I, x) | f_{\theta}(I, x) < \tau \,\}$$
$$Q_{right}(\varphi) = Q / Q_{left}$$

Equation 6.17

4. Repetition of step 3 for Qleft (φ *) and Qright (φ *) recursively until some preset stop conditions are met: the tree reaches maximum depth; the information gain or the number of pixels in the node fall below a minimum. The node where the stop condition occurred is treated as a leaf node.

5. Estimation of the probability distribution $P_t(c_i|I, x)$ for class c_i at each leaf node of the tree, calculated on the normalized histogram of body part labels *c* with each histogram bin being normalized over the total number of training pixels for that label in *Q*.

In that manner each tree *t* of the randomized forest can serve as a per pixel classifier for a test image. To avoid misclassification due to tree overfitting, the inferred pixel probability distribution is averaged over all trees in the forest as follows:

$$P(c_i|I, x) = \frac{1}{T}\sum_{t=1}^{T} P_t(c_i|I, x)$$

Equation 6.18

6.2.3 Graph-Cuts segmentation

GC [70] is an energy minimization framework which has been considerably applied in image segmentation –both binary and multi-label–, with highly successful results. In this work, we extend the GC theory to be used in depth images and optimize the results obtained from the RF approach in order to deal with automatic multi-label segmentation.

Given $X = (x_1, \dots, x_i, \dots, x_{|P|})$ the set of pixels of the depth image I, lets define $P = (1, \dots, i, \dots, |P)$ the set of indexes of I; N the set of unordered pairs {i, j} of neighboring pixels of P under a defined neighborhood system –typically 4- or 8-connectivity–, and $L = (L_1, \dots, L_i, \dots, L_P)$ a vector whose components L_i specify the labels assigned to pixels $i \in P$.

This framework defines an energy function E(L) which combines local and contextual information, and whose minimum value corresponds to the optimal solution of the problem – in our case, the optimal segmentation:

$$E(L) = U(L) + \lambda B(L)$$

Equation 6.19

The first term of the energy function is called the “unary potential”. This potential encodes the local likelihood of the data by assigning individual penalties to each pixel for each one of the defined labels:

$$U(L) = \sum_{i \in P} U_i(L_i).$$

Equation 6.20

The second term or “boundary potential” encodes contextual information by introducing penalties to each pair of neighboring pixels as follows:

$$B(L) = \sum_{\{i,j\} \in N} B_{\{i,j\}} \Omega(L_i, L_j)$$

Equation 6.21

being $\Omega(L_i, L_j)$ a function that introduces prior costs between each possible pair of neighboring labels. Finally, $\lambda \in R^+$ is a weight that specifies the relative importance of the boundary term against the unary term.

Once the energy function is defined, a graph is built following the neighborhood system used in the boundary potential $B(L)$, and the energy function is transferred to this graph. In the case of binary segmentation, i.e. $L_i \in \{0,1\}, \forall_i \in P$, the min-cut algorithm [70] finds the minimum cut of this graph –which corresponds to the minimum energy– and thus, the optimal segmentation. When $L_i \in \{0, \dots, N_L\}, N_L > 1$, two main algorithms used to be applied in order to find not the minimum energy, but a suboptimal approximation of it: *α*-*β*-swap and *α*-expansion [70]. While the first one is less restrictive and can be applied in a broader range of energy functions, the second one has been proved to obtain better results, as long as the energy function fulfills some conditions [70]. In our case, we based our segmentation methodology for depth maps on α-expansion GC. In the following subsections, the specific energy function potentials we designed for our problem are defined.

6.2.3.1 Unary potential

The unary potential encodes local likelihood for each pixel belonging to each one of the labels *L_i* of our problem. In our case, we have used the log-likelihood of the probabilities returned by the RF for the computation of the unary potential:

$$U_i(L_i) = -\ln\left(P(c_i|I, x)\right)$$

Equation 6.22

obtaining a unary cost potential for each class c_i – corresponding to label L_i in GC. This step is shown at the top of Figure 6.9, where the output probabilities of the leafs of RF trees are used to compute the unary potentials $U_i(L_i)$ at the input edges of the GC graph.

6.2.3.2 Boundary potential

In the case of the boundary potential, we tested three different approaches, all of them based on the same formulation:

$$B_{\{i,j\}} = \frac{1}{dist(i,j)} e^{-\beta\|x_i - x_j\|^2}$$

Equation 6.23

where $\beta = (2\langle(x_i - x_j)^2\rangle)^{-1}$ and dist(*i, j*) computes the Euclidean distance between the cartesian coordinates of pixels $\mathbf{x}_i$ and $\mathbf{x}_j$. The difference among the three different approaches we tested remains in the information of the pixels $\mathbf{x}_i$ and $\mathbf{x}_j$ we use in the exponential function. Firstly, we just used RGB information, as in the standard GrabCut algorithm [71]. Secondly, we used only depth information, and, finally we tried both of them together. For this last approach, we normalized the depth information in the range [0...255], and concatenated it with the RGB information, resulting in a 4-D RGBD vector.

Finally, we defined two different $\Omega(L_i, L_j)$ functions in order to introduce some prior costs between different labels. On one hand, we considered the trivial case where all different labels have the same cost:

$$\Omega_1(L_i, L_j) = \begin{cases} 0 \text{ for } L_i \neq L_j \\ 1 \text{ for } L_i \neq L_j \end{cases}$$

Equation 6.24

On the other hand, we introduced some spatial coherence between the different labels, taking into account kinematic constraints of the human body limbs:

$$\Omega_2(L_i, L_j) = \begin{cases} 0 & \text{for} \quad L_i = L_j \\ 10 & \text{for} \quad L_i = \mathrm{LU}, L_j = \mathrm{RU} \\ & \phantom{\text{for}} \quad L_i = \mathrm{LH}, L_j = \mathrm{RH} \\ 5 & \text{for} \quad L_i = \mathrm{LW}, L_j = \mathrm{RH} \\ & \phantom{\text{for}} \quad L_i = \mathrm{RW}, L_j = \mathrm{LH} \\ 1 & \text{otherwise} \end{cases}$$

Equation 6.25

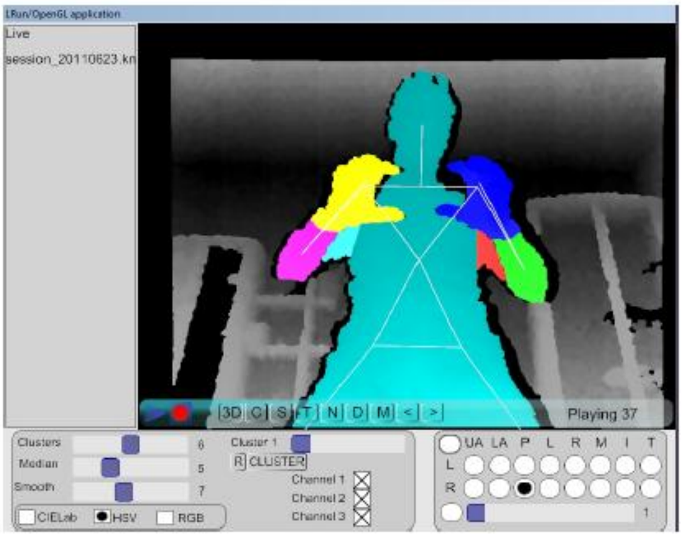
With this definition of the inter-label costs, we are making difficult for the optimization algorithm to find a segmentation in which there exists a frontier between whether the right and left upper-arms, right and left hands, or in lower measure, between left hand and right lower-arm, and viceversa. Therefore, we are assuming that poses in which the two hands are touching are not probable. This label coherence cost should be estimated for each particular problem domain. In our particular data set of poses, the values of 1, 5, and 10 were experimentally computed.

6.2.4 Experiments and results

Before the presentation of the results, we describe the data considered and the different methods, parameters, and validation protocol of the evaluation.

Data: For the purposes of gathering ground truth data, we defined a new data set of different sessions where the actors are performing different gestures with his/her hands – only upper body is considered. Each frame size is 640×480 and contains 24 bit RGB image, 12 bit depth and human skeletal graph. The data set was defined in a semisupervised manner. From the human skeletal and the depth buffer an additional label buffer is roughly generated. The
upper and lower arms are labeled by the pixels bounded by from the cylinders between the enclosing joints of shoulder, elbow and hand, while the palm is labeled by the pixels bounded by a sphere centered in the joint of the hand. The RGB, depth, and skeletal data are directly obtained via the OpenNI library [10]. As a final stage, each frame was manually edited to correct the automatically generated labels. The ground truth is composed by capturing 3 actors in 3 sessions gathering 500 frames in total. It is important to mention that after the manual editing there still exist around 1% of false positive labels due to editor mistakes. An example of the developed interface for semi-automatic ground-truth generation is shown in Figure 6.9.

We also defined an extra small test consisting of 63 images of hand regions with six labels per image. Inspired by the reported test parameters and accuracy results in [12], our experiments rest on the following setup: we perform a 5-fold cross validation over the available 500 frames by training random forest of three trees of depth 20, 130 unique training images per tree, 1000 uniformly distributed pixels per image, 100 candidate features Q, and 20 thresholds τ per feature. Each test set consists of 100 images. Carrying a randomized test trial, we analyze the effect of the choice of test parameters on the classification accuracy and compare the results with another set of features: a mixture of depth Equation 6.16 and gradient Equation 6.26 features:



Clusters 6
Median 5
Smooth 7
□ CIELab ● HSV □ RGB
Channel 1 ☒
Channel 2 ☒
Channel 3 ☒

Figure 6.9 Example of body parts segmentation

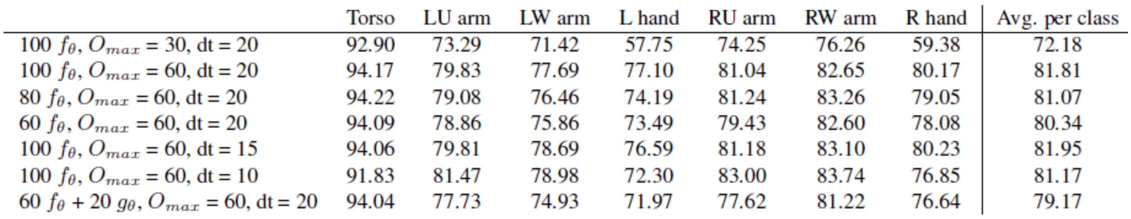
$$g_{\theta}(I, x) = \measuredangle\left(\nabla I\left(x + \frac{u}{\mathrm{d_I(x)}}\right) - \nabla I\left(x + \frac{v}{\mathrm{d}_I(x)}\right)\right)$$

Equation 6.26

where $\nabla I(x)$ is the gradient vector at pixel **x**, and the feature $g_{\theta}(I, x)$ represents the angle between the two gradient vectors at offsets **u** and **v** from **x**. When applying GC, the λ parameter was set to 50 for all the performed experiments.

6.2.4.1 Random forest results

Table 6.2 shows the estimated average classification accuracy for each of the considered labels, using the maximum RF probabilities for each pixel. Without claiming exhaustiveness of our experiments, the results from Table 6.2 allow us to make the following analysis: The maximum offset *Omax* has the greatest impact on the accuracy results at the hands regions, which are with the smallest area in our body part definition. Doubling the size of O_{max} leads to an increase in the accuracy of 20% for the hands and 6% for the other body parts. In other words, O_{max} increases the feature diversity and the global ability to represent spatial detail. The number of candidate features *Q* would not have such a tremendous impact on the accuracy as the O_{max} parameter, though a higher number helps identifying the most discriminative features. A decrease from 100 to 80 features drops the hands accuracy with 1-3%. We also tested the impact of the depth of the decision trees. Trimming the trees to depth 15 has a very little impact, showing an improvement of 0.1% on the average accuracy that may weekly be attributed to better classification at the lower arm regions. Trimming to depth 10 shows a 4% decrease in the accuracy at the hands, i.e. the tree is not trained well enough. Our analysis indicates that we may be witnessing slight overfitting at tree depth of 20 due to the small amount of training images. Our final test includes comparison over combination of both features f_θ and g_θ. Since the depth data provided by Kinect is noisy, we apply a Gaussian smoothing filter before calculating the image gradients and the feature from Equation 6.26. We chose the gradient feature since it complements the relations of depth features with information about the orientation of local surfaces. However, in our test we did not found significant differences in the performance of the RF approach when including this kind of features.



	Torso	LU arm	LW arm	L hand	RU arm	RW arm	R hand	Avg. per class
100 f_θ, O_{max} = 30, dt = 20	92.90	73.29	71.42	57.75	74.25	76.26	59.38	72.18
100 f_θ, O_{max} = 60, dt = 20	94.17	79.83	77.69	77.10	81.04	82.65	80.17	81.81
80 f_θ, O_{max} = 60, dt = 20	94.22	79.08	76.46	74.19	81.24	83.26	79.05	81.07
60 f_θ, O_{max} = 60, dt = 20	94.09	78.86	75.86	73.49	79.43	82.60	78.08	80.34
100 f_θ, O_{max} = 60, dt = 15	94.06	79.81	78.69	76.59	81.18	83.10	80.23	81.95
100 f_θ, O_{max} = 60, dt = 10	91.83	81.47	78.98	72.30	83.00	83.74	76.85	81.17
60 f_θ + 20 g_θ, O_{max} = 60, dt = 20	94.04	77.73	74.93	71.97	77.62	81.22	76.64	79.17

Table 6.2 Average per class accuracy in % calculated over the test samples in a 5-fold cross validation. f_θ represents features of the depth comparison type from Equation 6.16, while g_θ - the gradient comparison feature from Equation 6.26. O_{max} indicates the maximum absolute value for the *x; y* coordinates of the offsets **u** and **v**. Parameter *dt* stands for tree depth

In order to show the generalization capability of the proposed approach, we tested an extra case of study consisting of segmenting finger regions. For this test we only considered a manual annotated depth video of 63 frames. The results applying the same validation than in the previous case show the best performance for the following setup: 1 tree of depth 15, 500 pixels per image, 100 candidate features *Q*, 20 thresholds τ per feature, and O_{max} = 45. The estimated average per class accuracy was 58.5% mostly due to the small number of training images. Figure 6.10 displays a couple of test images comparing the ground truth and the inferred labels. The results are promising, showing the generality of the presented approach for general multi-class labeling in depth images.

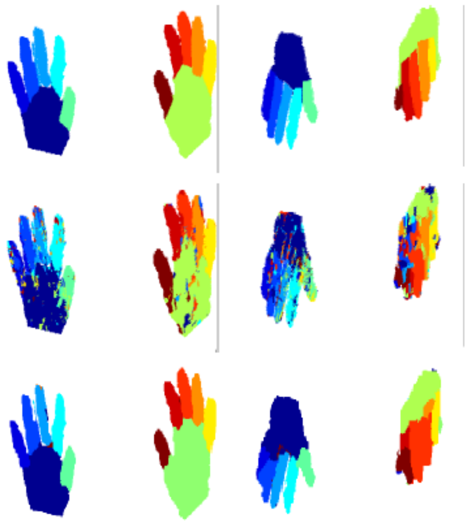


Figure 6.10 Results from RF classification in the case of hands. First row shows the ground-truth for two examples. Second row shows the RF classification results. Third row shows the final α-expansion GC segmentation results

6.2.4.2 Graphs-cut results

The results we obtained when applying GC over the probabilities returned by the RF are detailed in Table 6.3. We can see how these results improved the labelling obtained by the RF approach. If we take a closer look to the measurements, we can see that we obtain the best results when using only depth information for the computation of the boundary potential. In our case of study, adding RGB to the depth information reduce generalization of the boundary potential. In Figure 6.11 we can see some qualitative results of the segmentations.

	Torso	LU arm	LW arm	L hand	RU arm	RW arm	R hand	Avg. per class
Depth, $\Omega_1 (L_i, L_j)$	98.86	75.05	82.87	91.45	77.57	87.35	93.96	86.73
Depth, $\Omega_2 (L_i, L_j)$	98.86	0.7503	83.36	92.41	77.54	87.67	94.20	87.01
RGB+Depth, $\Omega_1 (L_i, L_j)$	99.02	72.02	81.86	90.29	76.56	86.84	92.14	85.53
RGB+Depth, $\Omega_2 (L_i, L_j)$	99.02	72.03	81.95	91.19	76.53	87.12	92.12	85.71

Table 6.3 Average per class accuracy in% obtained when applying the different GC approaches

Another interesting result is the influence of the prior costs given by the different $\Omega(L_i, L_j)$ functions. Clearly, when introducing spatial coherence with $\Omega_2(L_i, L_j)$, we obtain better results, specially in the segmentation of the hands, which are the parts with more confusion between them. Figure 6.12 shows a qualitative example of both approaches. In the second experiment, labelling pixels from hands, we achieve an average per class accuracy of 70.9%, which supposes even a greater improvement than in the case of human limbs segmentation. Figure 6.10 also shows some qualitative results of the GC approach, where we can appreciate that regions are more consistent and are better defined than in the case of just using RF probabilities. It is worth mentioning that for this experiment, we used $\Omega_1(L_i, L_j)$ as the cost function between labels, and yet we obtained consistent results.

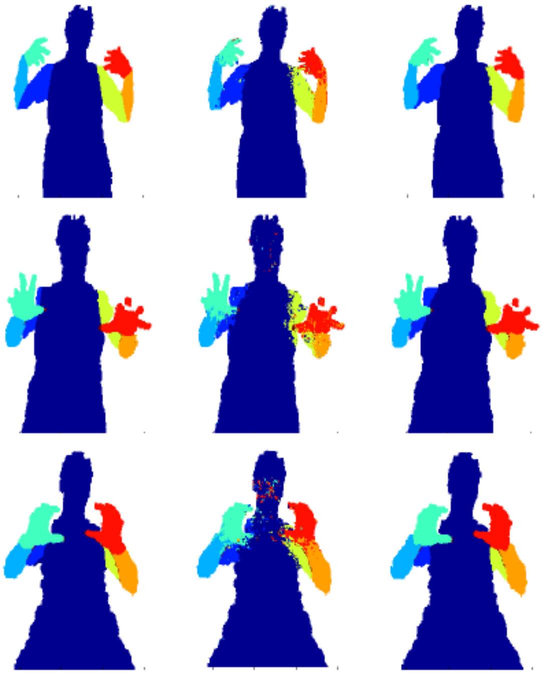


Figure 6.11 From left to right: Depth map Ground Truth and labels, RF inferred results, and GC result over RF output probabilities, respectively

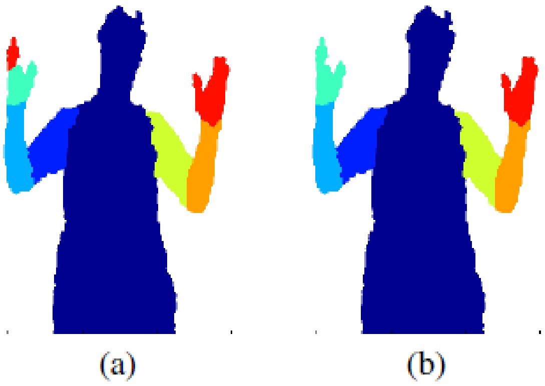


Figure 6.12 Comparison of results without (a) and with (b) spatially consistent labels

Chapter 7

Gesture Recognition

In this chapter, we use data collected from a Kinect sensor, described by vectors of features that form the skeletal pattern ordered in time, to explore the feasibility of gesture recognition on a small case of study. We attempt to identify simple gestures that a person might perform with his or her full body (e.g. waving, jumping). To do this, we use the vector of features obtained from the skeletal model described in the previous chapter, to first identify the location and pose of a person's body, and from there, recognize patterns in the body's movement over time. We compare and propose novel adaptations to two state-of-the-art approaches for gesture recognition: Dynamic Time Warping and Hidden Markov Models.

7.1 Gesture recognition with an improved DTW

Our proposal is focused within the Dynamic Time Warping framework (DTW) [72]. Dynamic Time Warping allows to align two temporal sequences taking into account that sequences may vary in time based on the subject that performs the gesture. The alignment cost can be then used as a gesture appearance indicator.

The main contribution of our methodology is the introduction of a new method based on DTW for gesture recognition using depth data. We propose a Feature Weighting approach within the DTW framework to improve gesture/action recognition. First, we estimate a temporal feature vector of subjects based on the 3D spatial coordinates of fifteen skeletal human joints. From a set of different ground truth behaviors of different length, DTW is used to compute the inter-class and intra-class gesture joint variability. These weights are used in the DTW cost function in order to improve gesture recognition performance. We test our approach on several human behavior sequences captured by the Kinect sensor. We show the robustness of the novel approach recognizing multiple gestures, identifying beginning and end of gestures in long term sequences, and showing performance improvements compared with classical DTW framework.

The articulated human model is defined by the set of 15 reference points shown in Figure 6.5. This model has the advantage of being highly deformable, and thus, able to fit to complex human poses.

In order to subsequently make comparisons and analyze the different extracted skeletal models, we need to normalize them. In this sense, we use the neck joint of the skeletal model as the origin or coordinates (OC). Then, the neck is not used in the frame descriptor, and the remaining 14 joints are using in the frame descriptor computing their 3D coordinates with respect to the OC. This transformation allows us to relate pose models that are at different depths, being invariant to translation, scale, and tolerant to corporal differences of subjects. Thus, the final feature vector $\boldsymbol{V_j}$ at frame j that defines the human pose is described by 42 elements (14 joints x three spatial coordinates):

$$V_j = \{\{v_{j,x}^{1}, v_{j,y}^{1}, v_{j,z}^{1}\}, \dots, \{v_{j,x}^{14}, v_{j,y}^{14}, v_{j,z}^{1}\}$$

Equation 7.1

The original DTW algorithm [73] was defined to match temporal distortions between two models, finding an alignment warping path between the two time series $Q = \{q_1, \dots, q_n\}$ and $C = \{c_1, \dots, c_n\}$. In order to align these two sequences, a M_{mxn} matrix is designed, where the position (i, j) of the matrix contains the distance between c_i and q_j. The Euclidean distance is the most frequently applied. Then, a warping path:

$$W = \{w_1, \dots, w_T\}, \max(m, n) \leq T < m + n + 1$$

Equation 7.2

is defined as a set of “contiguous” matrix elements that defines a mapping between C and Q. This warping path is typically subjected to several constraints:

Boundary conditions: $w_1 = (1,1)$ and $w_T = (m, n)$.
Continuity: Given $w_{t-1} = (a', b')$, then $w_t = (a, b)$, $a - a' \leq 1$ and $b - b' \leq 1$.
Monoticity: Given $w_{t-1} = (a', b')$, then $w_t = (a, b)$, $a - a' \leq 1$ and $b - b' \leq 1$, this forces the points in *W* to be monotically spaced in time.
We are generally interested in the final warping path that satisfying these conditions minimizes the warping cost:

$$DTW = (Q, C) = \min \left\{ \frac{1}{T} \sqrt{\sum_{t=1}^{T} w_t} \right\}$$

Equation 7.3

where *T* compensates the different lengths of the warping paths. This path can be found very efficiently using dynamic programming to evaluate the following recurrence which defines the cumulative distance $\gamma(i, j)$ as the distance $d(i, j)$ found in the current cell and the minimum of the cumulative distance of the adjacent elements:

$$\gamma(i, j) = d(i, j) + \min \{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}$$

Equation 7.4

Given the nature of our system to work in uncontrolled environments, we continuously review the stage for possible actions or gestures. In this case, our input feature vector *Q* is of “infinite” length, and may contain segments related to gesture *C* at any part. Next, we describe our algorithm for begin-end of gesture recognition and the Feature Weighting proposal within the

DTW framework.

7.1.1 Begin-end of gesture detection

In order to detect a begin-end of gesture $C = \{c_1, \ldots, c_m\}$ in a maybe infinite sequence $Q = \{q_1, \ldots, q_\infty\}$, a $M_{mx\infty}$ matrix is designed, where the position (i, j) of the matrix contains the distance between c_1 and q_j, quantifying its value by the Euclidean distance, as comment before. Finally, our warping path is defined by $W = (w_1, \ldots, w_\infty)$ as in the standard DTW approach. Our aim is focused on finding segments of ***Q*** sufficiently similar to the sequence ***C***. The system considers that there is correspondence between the current block **k** in ***Q*** and a gesture if satisfying the following condition:

$$M(m, k) < \mu, k \ \in \ [1, \ldots, \infty]$$

Equation 7.5

for a given cost threshold µ. This threshold value is estimated in advance for each of the categories of actions or gestures using leave-one-out cross-validation strategy. This involves using a single observation from the original sample as the validation data, and the remaining observations as the training data. This is repeated such that each observation in the sample is used once as the validation data. At each iteration, we evaluate the similarity value between the candidate and the rest of the training set. Finally we choose the threshold value which is associated with the largest number of hits within a category.

Once detected a possible end of pattern of gesture or action, the working path W can be found through backtracking of the minimum path from $M(m, k)$ to $M(0, z)$, , being z the instant of time in Q where the gesture begins. The algorithm for begin-end of gesture detection for a particular gesture C in a large sequence Q using DTW is summarized in Table 7.1. Note that $d(i, j)$ is the cost function which measures the difference among our descriptors V_i and V_j. An example of a begin-end gesture recognition for a model and infinite sequence together with the working path estimation is shown in Figure 7.1.

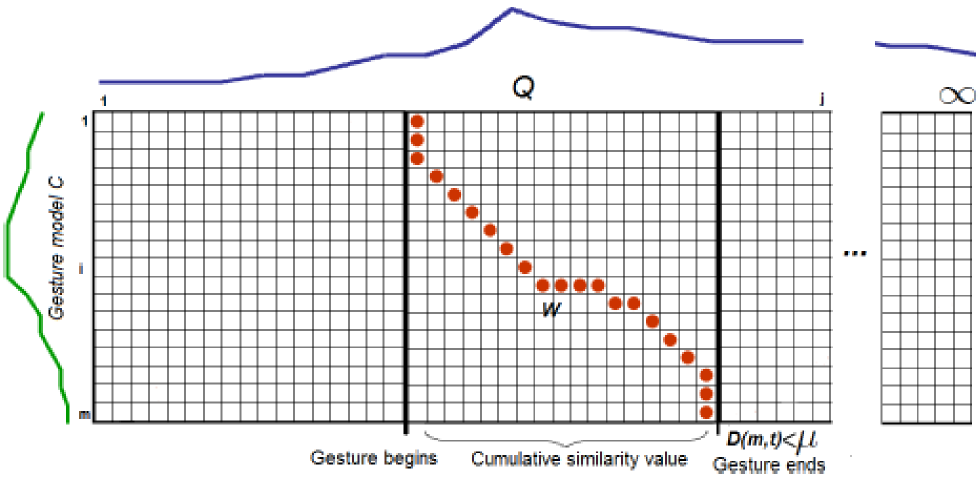
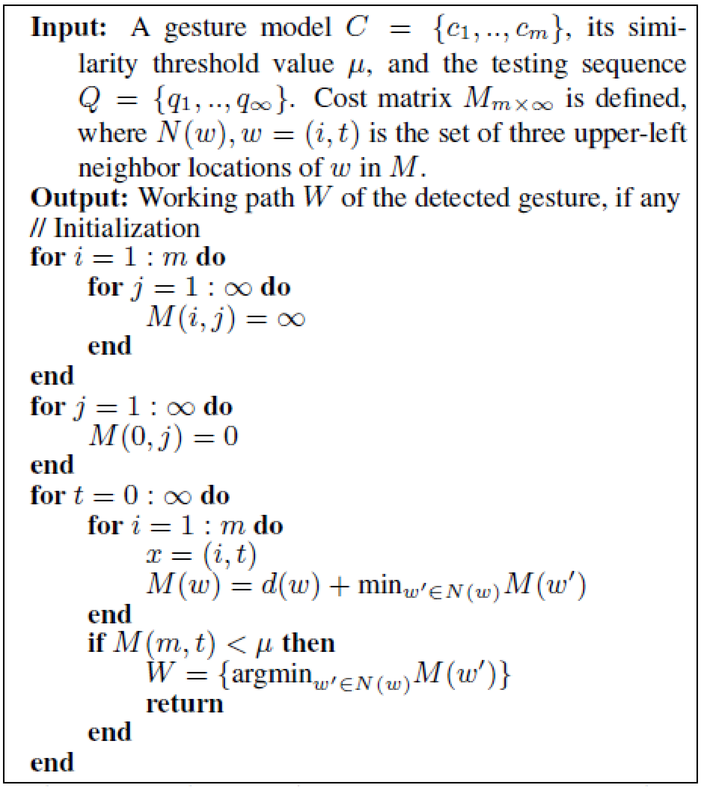


Figure 7.1 Begin-end of gesture recognition of a model C in an infinite sequence Q

7.1.2 Feature Weighting in DTW

In this section, we propose a Feature Weighting approach to improve the cost distance computation $d(w)$ of previous begin-end DTW algorithm.

In standard DTW algorithm, cost distances among feature vectors c_i and q_j (3D coordinates of the skeletal models in our case) are computed equally for each feature of the descriptors. However, it is intuitive that not all skeletal elements of the model participate equally for discriminating the performed gesture. For instance, the movement of the legs when performing hand shaking should not have influence, and thus, computing their deviation to a correspondence model of the gesture adds noise to the cost similarity function. In this sense, our proposal is based on associating a discriminatory weight to each joint of the skeletal model depending on its participation in a particular gesture. In order to automatically compute this weight per each joint, we propose an inter-intra gesture similarity algorithm.



```
Input: A gesture model $C = \{c_1, .., c_m\}$, its similarity threshold value $\mu$, and the testing sequence $Q = \{q_1, .., q_\infty\}$. Cost matrix $M_{m \times \infty}$ is defined, where $N(w), w = (i, t)$ is the set of three upper-left neighbor locations of $w$ in $M$.
Output: Working path $W$ of the detected gesture, if any
// Initialization
for $i = 1 : m$ do
    for $j = 1 : \infty$ do
        $M(i, j) = \infty$
    end
end
for $j = 1 : \infty$ do
    $M(0, j) = 0$
end
for $t = 0 : \infty$ do
    for $i = 1 : m$ do
        $x = (i, t)$
        $M(w) = d(w) + \min_{w' \in N(w)} M(w')$
    end
    if $M(m, t) < \mu$ then
        $W = \{\operatorname{argmin}_{w' \in N(w)} M(w')\}$
        return
    end
end
```

Table 7.1 DTW begin-end of gesture recognition algorithm

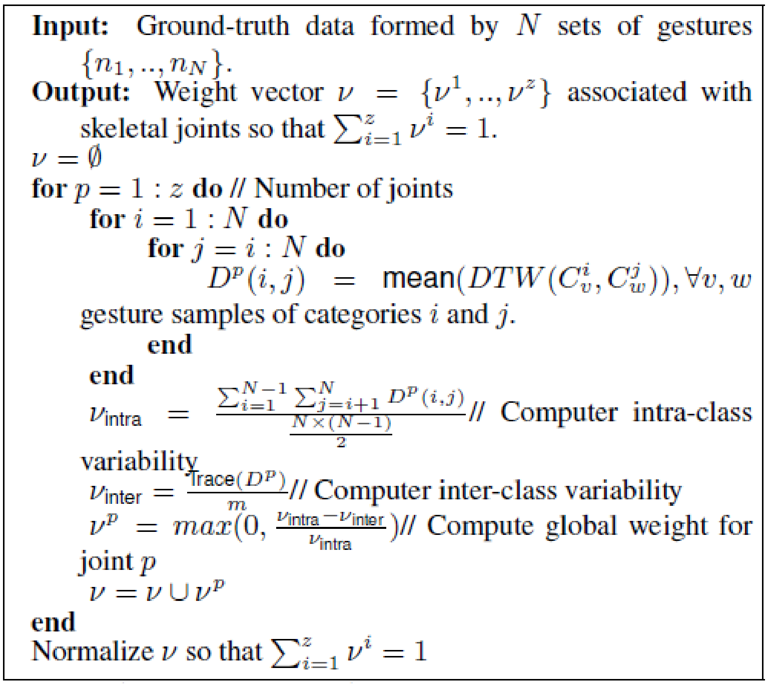
First, we perform a weight training algorithm based on a ground truth data of gestures. Given the data composed by $\{n_1, \dots, n_N\}$ gesture categories described using skeletal descriptors, the objective is to obtain the inter-intra coefficient of the joints for the data set. This estimation is performed per each joint using a symmetric cost matrix D_{NxN}. Each matrix element $D^p(i,j)$ for the matrix of joint ***p*** contains the mean DTW cost between all pairs of samples $C_i, C_j, \forall C_i \in n_i, \forall C_j \in n_j$ only considering the features of the descriptor related to the *p*-th joint, where n_i and n_j represent the set of samples for gesture categories *i* and *j* of the data set.

The mean DTW value at each position of the matrix D^p represents the variability of joint *p* between a pair of gestures. Note that the diagonal of D represents the intragesture variability per joint for all the gesture categories, meanwhile the rest of the elements compare the variability of joint p for two different gesture categories, codifying the inter-gesture variability. Since gestures, as any other object recognition system, will be more discriminative when increasing inter-distance and reducing intra-distance, a discriminative weight is defined as shown in Table 7.2, which assigns high cost to joints high high intra-inter difference values and low cost otherwise. Moreover, the assigned weight is normalized in the same range to be comparable for all joints. Note that at the end of this procedure we have a final global weight vector $v = \{v^1, \ldots, v^z\}$, with a weight value v^p for the *p*-th joint, which is included in the re-definition of the begin-end DTW algorithm cost function $d(w)$ to improve gesture recognition performance as follows:

$$d\big(c_i, c_j\big) = \sqrt{\sum_{p=1}^{|c_i|} ((c_i^p - c_j^p) \cdot v^p)^2},$$

Equation 7.6

where $|c_i|$ is the length of the feature vector c_i. The Feature Weighting algorithm for computing the weight vector $v = \{v^1, \ldots, v^z\}$ is summarized in Equation 7.3.



```
Input: Ground-truth data formed by $N$ sets of gestures $\{n_1, .., n_N\}$.
Output: Weight vector $\nu = \{\nu^1, .., \nu^z\}$ associated with skeletal joints so that $\sum_{i=1}^{z} \nu^i = 1$.
$\nu = \emptyset$
for $p = 1 : z$ do // Number of joints
    for $i = 1 : N$ do
        for $j = i : N$ do
            $D^p(i,j) = \text{mean}(DTW(C_v^i, C_w^j)), \forall v, w$ gesture samples of categories $i$ and $j$.
        end
    end
    $\nu_{\text{intra}} = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} D^p(i,j)}{\frac{N \times (N-1)}{2}}$ // Computer intra-class variability
    $\nu_{\text{inter}} = \frac{\text{Trace}(D^p)}{m}$ // Computer inter-class variability
    $\nu^p = max(0, \frac{\nu_{\text{intra}} - \nu_{\text{inter}}}{\nu_{\text{intra}}})$ // Compute global weight for joint $p$
    $\nu = \nu \cup \nu^p$
end
Normalize $\nu$ so that $\sum_{i=1}^{z} \nu^i = 1$
```

Table 7.2 Feature Weighting in DTW cost measure

7.1.3 Experiments and results

We designed a new data set of gestures using the Kinect device consisting of five different categories: jumping, bendding, clapping, greeting, and noting with the hand. It has been considered 10 different actors, 10 different backgrounds, and 100 sequences per subject for recording the data set. Thus, the data set contains the high variability from uncontrolled environments. The resolution of the video depth sequences is 340x280 at 30 FPS. The data set contains a total of 1000 gesture samples considering all the categories. The ground-truth of each sequence is performed manually by examining and noting the position in the video when some actor begin-ends a gesture. Some samples of the captured gestures for different categories are shown in Figure 7.2.

For the implementation of the system we used C/C++, efficiently using dynamic
programming to evaluate the recurrence which defines the cumulative distance between vectors of features on each frame. The people detection system used is provided by the public library OpenNI [74]. This library has a high accuracy in people detection, allowing multiple detection even in cases of partial occlusions. The detection is accurate as people remain at a minimum of 60cm from the camera and up to 4m, but can reach up to 6m but with less robust and reliable detection.

For the validation our approach and classical DTW algorithm, we compute the Feature Weighting vector v and gesture cost threshold µ over a leave-one-out validation. The validation sequences may have different length size since they can be aligned using DTW algorithm and trained for the different estimated values of µ. We validate the begin-end of gesture DTW approach and compare with the Feature Weighting methodology within the same framework. As a validation measurement we compute the confusion matrix for each test sample of the leaveout-out strategy. This methodology allows us to perform an exhaustive analysis of the methods and data set. Adding all test confusion matrices in a performance matrix C_m, final accuracy A is computed using the following formula:

$$A = 100 \cdot \frac{Trace(C_m)}{NC + \sum_{i=1}^{m} \sum_{j=1}^{m} C_m(i,j)}$$

Equation 7.7

Where NC contains the number of samples of the data set that has not been classified by any gesture since the classification threshold µ has not been satisfied. This evaluation is pessimistic and realistic since both a sample which is not classified or is classified more than once penalizes the final evaluation measurement.

The obtained results applying DTW begin-end gesture recognition and including the Feature Weighting approach on the new data set are shown in Table 7.3. The results show the final performance per gesture over the whole data set using both classification strategies. The best performance per category is marked in bold. Note that for all gesture categories, the begin-end DTW technique with Feature Weighting improves the accuracy of standard DTW.

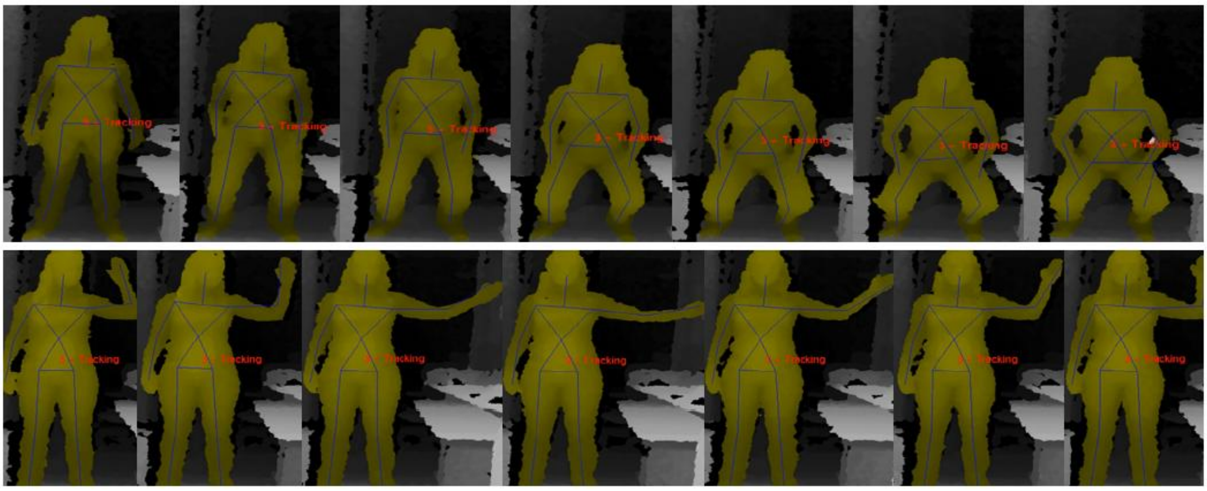
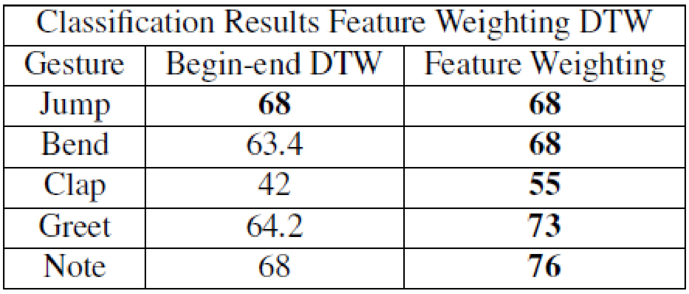


Figure 7.3



Classification Results Feature Weighting DTW		
Gesture	Begin-end DTW	Feature Weighting
Jump	**68**	**68**
Bend	63.4	**68**
Clap	42	**55**
Greet	64.2	**73**
Note	68	**76**

Table 7.3 Comparative table of results obtained

Chapter 8

Applications developed

In this chapter we are going to apply the methodology explained during this work. These applications are the result of the methodology that is explained in the previous chapters. Applications developed are mainly based on the recognition of human posture. In the first application we focus on evaluating the static pose for a specialized analytical study in functional rehabilitation. In this first application is very important the robustness and reliability of the data obtained, for this reason we place special emphasis on calibration and phase distortion of the image, analyzing the areas of interest through body marker placement. In the second application is a study in dynamic conditions of the position does not control. In this second application key markers will not be used as a major application feature is no invasion of the system on the actors. In this second application will place special emphasis on the results of the classification and recognition of gestures.

8.1 ADIBAS Posture

The analysis of posture and range of motion are essential to understanding the optimization of the gesture and improve, thus, the detection performance and possible injury. This quantification is especially interesting in athletes or in patients with neurological injury or muscle-skeletal system, allowing know the evolutionary process of these patients, evaluate the effectiveness of therapy applied and propose, if necessary, a modification of the treatment protocol. We present an automated system that allows, through a non-invasive technology, the automatic acquisition of LED markers placed on the patient and further analysis to show the specialist objective data to enable better support for diagnosis. It also describes an analytical system of the body posture without markers, where it has operated in dynamic sequences provides a high degree of naturalness to the patient when performing functional exercises.

8.1.1 Motivation

Corporal evaluation is a physical analysis procedure included in the American Medical Association policies. An incorrect postural alignment can alter the distribution of the articulation efforts, and produce an irreversible articular degeneration, inadequate muscular tension, and back pain. In particular, near 80% of the world population will be affected of back pain during his life. Current practices to analyze back problems are expensive and invasive, and alternative procedures should be required.

Digital biometry is defined by the American Society of photogrammetry as: “*the technology to obtain reliable information from physical objects or from the environment though recording processes, measurements or image interpretation*”. The systems based on this technology are able to estimate morphological or functional alternations, being an accessible resource for professionals from health experts as well as professionals from sports care. If we are able to report a robust and automatic system based on digital biometry to the expert, he/she may optimize the time dedicated to each test and reduce the risk os systematic error computation in the measurements.

8.1.1 State of the art

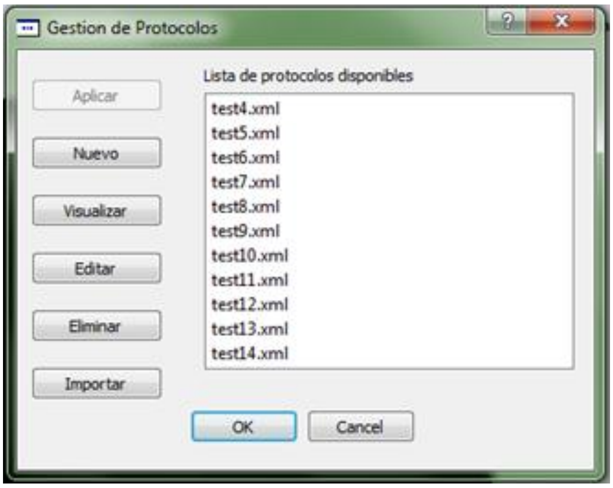
In the last decade, some works have been reported based on digital biometry applied to corporal analysis, being used by different health sectors because of the low implementation cost, high friability, and easiness of application. Most of the reported works are applied in the area of physiotherapy. Although the benefits of this technology are clear to the community, most of the reported systems are far from being fully-automatic. Two of the most well-known and used applications are Posture Print [1], from Brazil (SAPo), and Italy (DBIS), which require from continuous user interaction. Some other works are reported in literature. In [2], the authors present an initial approach to analyze the posture as a combination of rotation and translation matrices of three main corporal blobs. The results of the approach are satisfactory, though the analysis of only three human blobs reduces the system reliability. The work of [3] bases on the previous reference to define an environment of multiple cameras to calibrate the scene. Although the system estimate metrics among markers on the subjects, the system requires from full user interaction in order to select the relevant image points. Finally, in the work of [4], a large analysis of data capture by the previous approaches is performed. In this work, we present a fully-automatic system that is able to segment the human body as well as the markers distributed among the human body.

8.1.2 Methodology

To carry out this project we have subdivided the problem into four main modules. Each of the modules to be cited below may be viewed as black boxes of incoming and outgoing data, but for the good overall performance of the application, there must be a continuous flow of data communication and synchronization between the different modules. Description of each module is as follows:

1. Module I: handles communication between the multisensor device and PC. This module must be capable of encoding the visual signal from the RGB camera and infrared sensor from the deep to be able to treat this information later. These data must be processed with high fluidity, in order to get good feedback from the user application. To use this feature OpenNI middleware, while our system is a multisensor device PrimeSense company.

2. Module II: the alignment occurs between the RGB and depth maps. In this module is implemented the primitives to obtain the world coordinates from the RGBD space, it is for this reason that this module will provide the reliability of the data the system. Within this module also must use image processing techniques to extract the body markers. In this module we have used OpenCV library. For these functions call this module as a module of computer vision.


Gestion de Protocolos

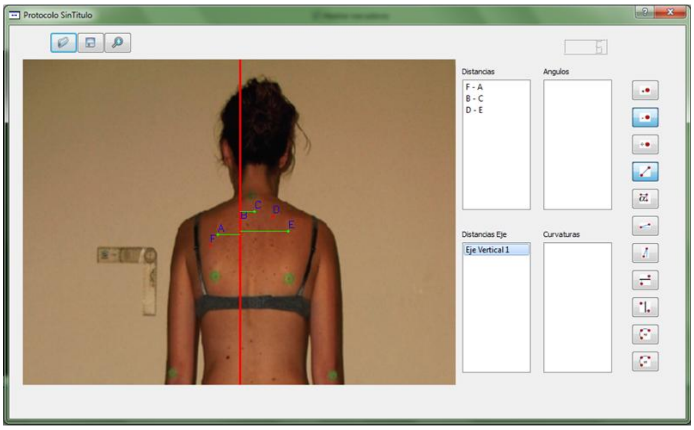
Lista de protocolos disponibles

test4.xml
test5.xml
test6.xml
test7.xml
test8.xml
test9.xml
test10.xml
test11.xml
test12.xml
test13.xml
test14.xml

Figure 8.1 Intelligent analysis example

3. Module III: after obtaining the positions of the markers body, is reconstructed the skeletal model for subsequent postural analysis. To reconstruct the posture which is required to analyze are used techniques of alignment and shape reconstruction like Active Shape Models. This reconstruction will allow the therapist to perform intelligent analysis depending on the protocol which they want to make. Because of these features this is the artificial intelligence module.

4. This module collects all analysis values to be displayed in an intuitive interface. This is the visualization module. It has been used NokiaQt library.


Protocolo SinTitulo

Distancias
F - A
B - C
D - E

Angulos

Distancias Eje
Eje Vertical 1

Curvaturas

Figure 8.2 Intuitive interface

Some modules mentioned before require constant must have been pre-computed. This is the case of Computer Vision module. For the world has been necessary to coordinate a calibration process using two-dimensional template. Then we detail this process.

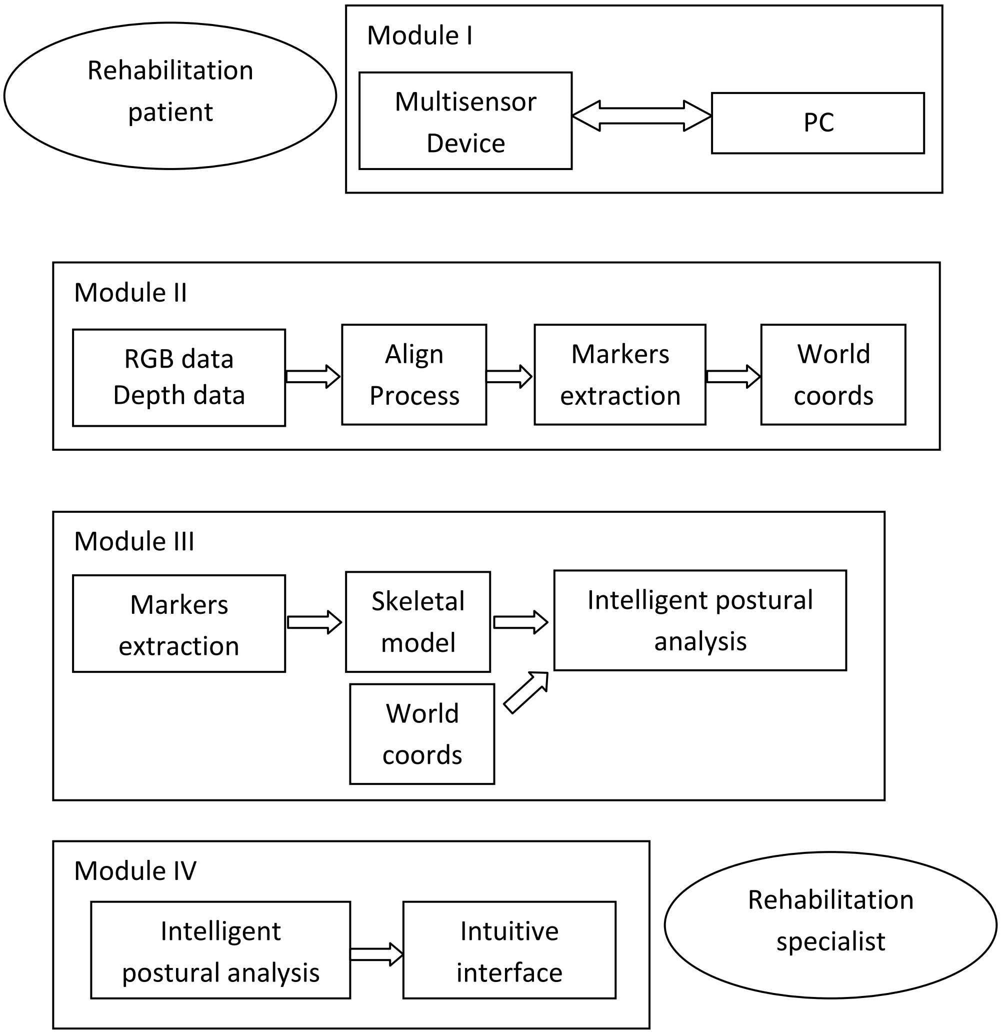


Figure 8.1 ADIBAS [75] posture architecture

8.1.3 Pre-process: Calibration

The calibration process was conducted using the methodology discussed in Chapter 2, calibration using two-dimensional template. The software and hardware used is as follows:

-PC Intel i7 2,4GHz CPU double kernel

- two-dimensional template of size: 115,5cmx84cm

- RGB camera resolution 640x480 pixels.

- OpenCv library

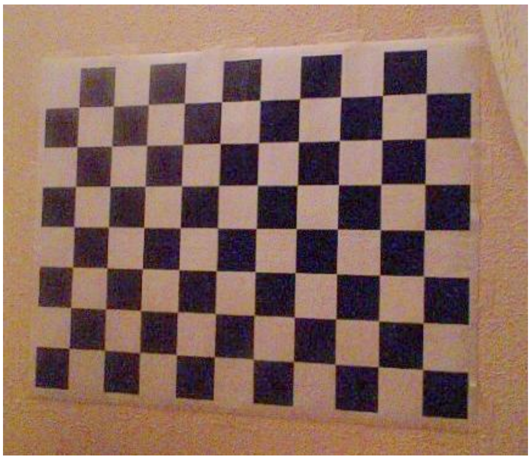


Figure 8.2 2D Calibration template

One of the main features of the visual characteristics calibration template is the complementary colors. This feature will help us to implement a detection of corners and edges on the template. We will use an algorithm very common in computer vision, edge detector Harris corners. As shown in the figure below, the detection of corners and edges is very precise.

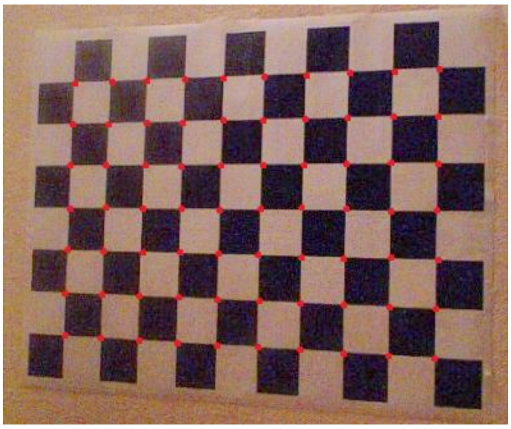


Figure 8.3 Corners detected by Harris Algorithm (pixel error =0.13 pixels)

Intrinsic values:

Focal distance x = 5.3009194943536181e+02

Focal distance y = 5.2635860167133876e+02

Principal point x = 3.2821930715948992e+02

Principal point y = 2.6872781351282777e+02

Distortion values matrix:

kc = [2.6172416643533958e-01, -8.2104703257074252e-01, -1.0637850248230928e-03, 8.4946289275097779e-04]

The following table shows the results from the calibration data obtained (shown values of 10 random samples referred to corners).

Corners	Pixel postion		Real coord.		Computed coord.		Directional error	
	u (pixels)	v (pixels)	x (cm)	y(cm)	$\hat{x}$ (cm)	$\hat{y}$ (cm)	e_x cm	e_y cm
1	52	32	18.5	8.5	18.25634	8.785612	0.24366	-0.285
2	108	65	29	19	29.45855	19.32213	-0.4585	0.322
3	166	100	39.5	29.5	39.81254	29.49242	-0.3125	-0.008
4	226	140	50	40	49.95812	40.28745	0.0418	0.287
5	286	182	60.5	50.5	60.65413	50.12544	-0.1541	0.375
6	347	225	71	61	71.63214	61.48521	-0.6432	-0.485
7	408	272	81.5	71.5	81.1256	71.36515	0.3744	0.135
8	471	322	92	82	92.2325	82.36541	-0.2325	-0.365
9	534	377	102.5	92.5	102.5478	92.87125	-0.0478	-0.371
10	597	437	113	103	112.8542	103.35481	0.1458	0.354

In order to verify the accuracy of the calibration algorithm, keep the camera position and reposition the template for different shots (or vice versa). The error can be evaluated by the following expression:

$$Q_k = \frac{\sum_{i=1}^{m} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}}{m}$$

Equation 8.1

The average value is 0.3547cm, and this value is the mean error in the postural analysis posture.

8.2 ADHD disorder analysis

The goal is to make a proof of concept to show the influence of the variable "motivation" in the symptoms of the ADHD construct. The results of this analysis can be applied directly to support an objective diagnosis of ADHD in clinical practice, avoiding the subjective observational during short periods of time of experts involved in the process (teachers and pedagogues, parents, psychiatrists, psychologists, etc.).

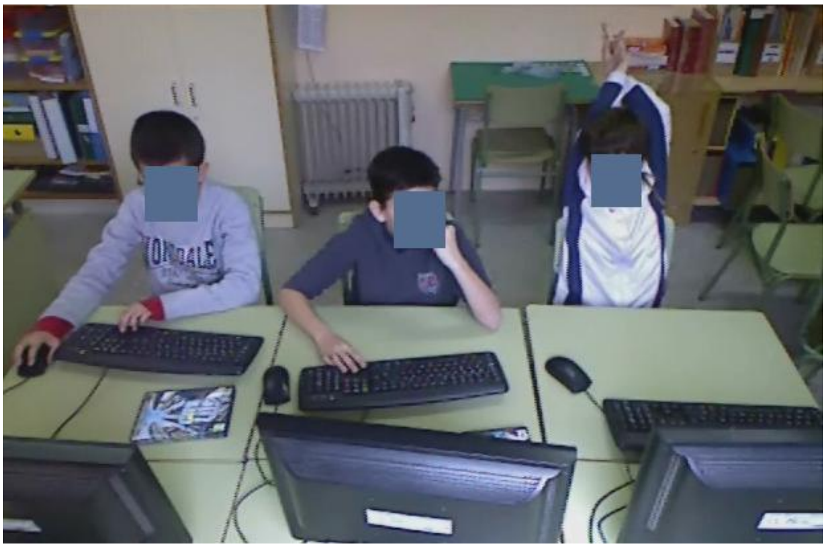


Figure 8.4 Environment example

8.2.1 Motivation

Epidemiological studies on ADHD have provided information both unclear and controversy in recent years. The reasons for this confusion are the use of unreliable diagnostic tools (observation by an adult) and, above all, the differences arising from the qualifying criteria adopted (Shin, 2000). Currently, the diagnosis is made following the criteria of the DSM IV-TR (American Psychiatric Association, 2000) or ICD-10. These criteria include three different blocks: attention deficit, hyperactivity, and impulsivity. These criteria do not include the variable "motivation". Thus, a child may show symptoms of inattention or hyperactivity in the context of little or no motivation but not shown in the context of strong motivation. From this thought explains the concept of motivation as the forms of connection and relationship between the successive states of the psychic event. Rodriguez and Violante (2010, 2006) bring us the concept of motivation as a set of features and psychological processes that constitute the mental activity of a person. The motivation operates a set of variables that trigger the behavior and / or guidance in determining a direction for achieving a goal.

8.2.2 Goal

A primary goal of this project is to analyze the reliability of automatic methods of Vision and Artificial Intelligence for automatic analysis of human behavior and help in TDAH clinical diagnosis.

8.2.3 Implementation

One of the principal conditions of the ADHD is the uncontrolled environment; the actors cannot interact with the sensors. So it is necessary to implement a full automatic framework. For this purpose, in this application we need to model the human pose without markers. To extract the skeletal model we use the methodology explicated in the chapter 6. In the framework we can use other techniques as support, like the RGB-D alignment, and DTW analysis for gesture recognition. In the next figure, we can see the architecture of the implementation.

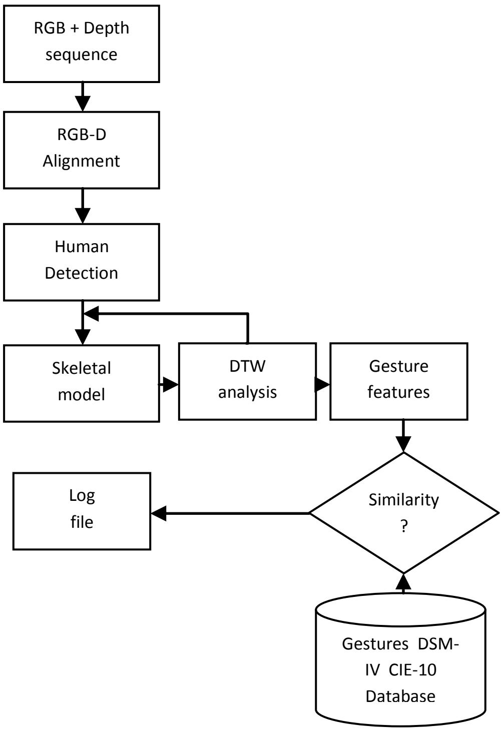


Figure 8.5 Architecture of the ADHD analysis

One of the most difficult tasks is the extraction of the skeletal model under uncontrolled environments. One example of the environment is shown in Figure 8.4. There exists several scenes with partially occlusion, and the pose of the actors would be very complex. We can observe some example of unstable skeletal model reconstruction in figure 8.6.

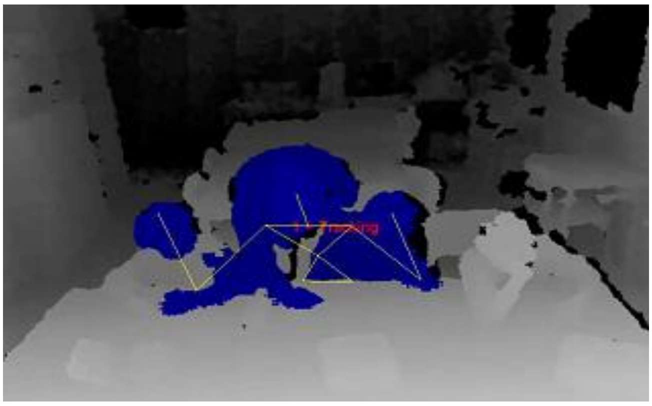


Figure 8.6 Gesture Example of instability of the system

Currently, the more robust part in the application is the gesture recognition for a good skeletal model sequence. In the next figure we can see how we can successfully extract a gesture from the DSM-IV and CIE-10 Database.

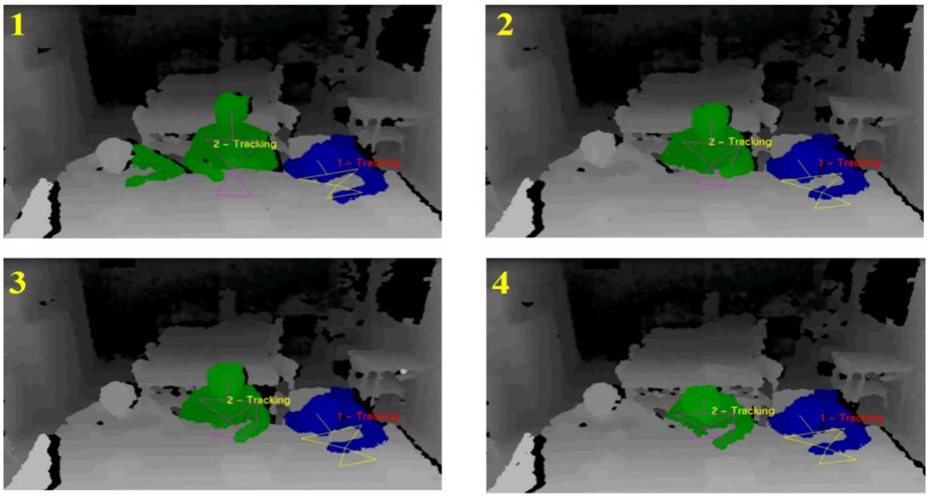


Figure 8.7 Short sequence of a gesture

Chapter 9

Conclusions

In this master thesis we proposed a general methodology for human pose recovery and behavior analysis using multi-modal RGB-Depth data.

The camera model is a necessary element if one needs to use the information in images to make measurements of the scene or to make 3D reconstructions of the same. This process involves estimating the intrinsic and extrinsic parameters of the camera or corresponding projection matrix. Two dimensional templates can be used for calibration purposes. The effectiveness of the calibration process depends on the quality of the measures involved, and the model used. **In conclusion, we defined a calibration method that is effective and valid for most calibration problems that may arise today**.

Once calibration and RGB-Depth alignment was performed, **we presented different methodologies for landmark detection and human pose recovery**. These methods are based on image segmentation using several state-of-the-art technologies, such as **Random Forest and Graph Cuts**. The presented results show robust pose segmentation for different configurations and points of view, improving previous state-of-the-art results in the field.

Moreover, **we presented different feature descriptions based on depth map information and also tested gesture recognition approaches for time series analysis. In particular, we focused on Dynamic Time Warping, and we showed that Feature Weighting improves classical DTW results.**

From an application point of view, **we presented different novel benchmarking data sets based on RGB-D information, and tested the novel approaches in a physiotherapy and TDAH clinical environments**, showing promising results.

As future lines of research we plan to test the proposed methodology in different real uncontrolled environments to analyze the main problematic of multi-modal behavior analysis and continue the research in the field.

Annex I

Contributions:
A.1.1 Papers and model utilities

Miguel Reyes, Gabriel Domínguez, and Sergio Escalera, Feature Weighting in Dynamic TimeWarping for Gesture Recognition in Depth Data, 1st IEEE Workshop on Consumer Depth Cameras for Computer Vision, International Conference in Computer Vision, Barcelona, 2012.

Miguel Reyes, José Ramírez-Moreno, Juan R. Revilla, Petia Radeva, and Sergio Escalera, ADiBAS: Sistema Multisensor de Adquisición Automática de Datos Corporales Objetivos, Robustos y Fiables para el Análisis de la Postura y el Movimiento, VI Congreso Iberoamericano de Tecnologíad de Apoyo a la Discapacidad, Mallorca, 16/06/2011-17/06/2011.

M. Reyes, S. Escalera, and Petia Radeva, Real-time head pose classification in uncontrolled environments with Spatio-Temporal Active Appearance Models, CVCRD'10, Achievements and New Opportunities in Computer Vision, pp. 101-104, CVC, 29/10/2010, Barcelona, ISBN 978-84-938351-0-1, 2010.

Antonio Hernández, Miguel Reyes, Sergio Escalera, and Petia Radeva, Spatio-Temporal GrabCut Human Segmentation for Face and Pose Recovery, IEEE International Workshop on Analysis and Modeling of Faces and Gestures, Computer Vision and Pattern Recognition, IEEE Computer Society, 13/06/2010-18/06/2010, San Francisco, ISBN 978-1-4244-7029-7, 2010.

Registered software number B3342-11, ADiBAS Posture: Automatic Digital Biometry Analysis System, Miguel Reyes, Sergio Escalera, José Ramírez, Juan Ramón Revilla, and Petia Radeva, 2011.

Laura Igual, Antonio Hernandez, Sergio Escalera, Miguel Reyes, Josep Moya, Joan Carles Soliva, Jordi Faquet,Oscar Vilarroya, Petia Radeva, Automatic Techniques for Studying Attention-Deficit/Hyperactivity Disorder, Jornada TIC Salut Girona, 04/05/2011-05/05/2011, Girona, Spain, 2011.

M. Reyes, J. Vitrià, P. Radeva, and S. Escalera, Real-Time Activity Monitoring of Inpatients, MICCAT, 28/10/2010, Gerona, 2010.

Annex II

A.1.2 Teaching

- Introduction program to intelligent systems for prospective university students. Universitat de Barcelona, 2010.

- Introduction program to computer vision systems for prospective university students (Campus Itaca). Universitat Autònoma de Barcelona, 2011.

- Introduction program to robotics for prospective university students (Campus Itaca). Universitat Autònoma de Barcelona, 2011.

  The teaching material used in these sessions has been prepared by the methodologies outlined in this master thesis.

Bibliography

[1] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. volume 2, pages 886–893, 2005.

[2] J.-Y. B. G. Cheung, T. Kanade and M. H. A. real time system for robust 3d voxel reconstruction of human motions. 2:714–720, 200. In Proceedings of CVPR'00, Hilton Head Island,(USA),

[3] D. H. L. D.W. Schwartz, A. Kembhavi. Human detection using partial least squares analysis. pages 24–31, 2009. ICCV.

[4] C. S. N. Dalal, B. Triggs. Human detection using oriented histograms of flow and appearance, 2006. European Conference on Computer Vision, Graz, Austria, May 713.

[5] J. K. A. B. Sabata, F. Arman. Segmentation of 3d range images using pyramidal data structures,. CVGIP: Image Understanding, 57(3):373–387, 1993.

[6] J. W. T. Darrell, G. Gordon and M. Harville. Integrated person tracking using stereo, color, and pattern detection. Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, pages 601 –608, 1998.

[7] S. L. HD. Yang. Reconstruction of 3d human body pose from stereo image sequences based on top-down learning. Pattern Recognition, 40(11):3120–3131, 2007.

[8] D. K. S. T. V. Ganapathi, C. Plagemann. Real time motion capture using a single time-of-flight camera. Proceedings of CVPR, pages 755–762, 2010.

[9] H.-C. P. J. Rodgers, D. Anguelov and D. Koller. Object pose detection in range scan data. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06), pages 2445 –2452, 2006.

[10] B. D. Y. Zhu and K. Fujimura. Controlled human pose estimation from depth image streams. CVPR Workshop on TOF Computer Vision, pages 1–8, 2008.

[11] H. Jain and A. Subramanian. Real-time upper-body human pose estimation using a depth camera. HP Technical Reports, 1(190), 2010.

[12] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. 2011.

[13] Schmid, Cordelia and Mohr, Roger and Bauckhage, Christian, "Evaluation of Interest Point Detectors", Int. J.Comput. Vision, vol. 37, no. 2, pp. 151-172.

[14] Bangpeng Yao, Li Fei-Fei , "Grouplet: A structured image representation for recognizing human and object interactions", Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 9-16, 2010.

[15] Mikolajczyk, Krystian and Schmid, Cordelia, "A Performance Evaluation of Local Descriptors", IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 10, 1615-1630.

[17] Y. Wexler, E. Shechtman and M. Irani, "Space-Time Completion of Video", In IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), March 2007.

[18] Lubomir Bourdev and Jitendra Malik, "Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations", International Conference on Computer Vision (ICCV), 2009.

[19] Long Zhu, Yuanhao Chen, Yifei Lu, Chenxi Lin, A. Yuille, "Max Margin AND/OR Graph learning for parsing the human body", Computer Vision and Pattern Recognition, pp. 1-8, 2008.

[20] Zhu, Long (Leo) and Chen, Yuanhao and Lin, Chenxi and Yuille, Alan, “Max Margin Learning of Hierarchical Configural Deformable Templates (HCDTs) for Efficient Object Parsing and Pose Estimation”, Int. J. Comput. Vision, vol. 93, no. 1, pp. 1-21, 2011.

[21] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. 2005, “Pictorial Structures for Object Recognition”, Int. J. Comput. Vision, Vol. 61, no. 1, pp. 55-79, 2005.

[22] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, “Object Detection with Discriminatively Trained Part-Based Models”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, no. 9. pp. 1627-1645, 2010.

[23] Yuri Y. Boykov and Marie-Pierre Jolly, “Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images”, International Conference on Computer Vision, 2001.

[24] Yuri Boykov, Olga Veksler, Ramin Zabih, “Fast approximate energy minimization via graph cuts”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, 2001.

[25] R. Larsen, M. B. Stegmann, S. Darkner, S. Forchhammer, T. F. Cootes, B. K. Ersbøll, Texture Enhanced Appearance Models, Computer Vision and Image Understanding, vol. 106, pp. 20-30, Elsevier, 2007.

[26] Shih S., Hung Y., Lin W. (1995) When should we consider lens distortion in camera calibration. Pattern recognition Vol. 28, No.3 pp 447-461

[27] Manual of Photogrammetry. American Society of Photogrammetry (1980) 4th edition.

[28] Brown D. C. (1966) Descentering distortion of lenses. PhotogrammetricEngineering and Remote Sensing

[29] Salvi J., Battle J., Mouraddib E. (1998). A robust-coded pattern projection for dynamic 3D scene measurement. International Journal of Pattern Recognition Lett. 19, pp 1055-1065

[30] Sun W., Cooperstock J. (2004). Requeriments for camera calibration: Must accuracy come with a high price?. IEEE Workshop on Applications of Computer Vision.

[31] Isern J. (2003) Estudio experimental de métodos de calibración y autocalibración de cámaras. Dr. Ing. Tesis. Departamento de Informática y Sistemas. Universidad de las Palmas de Gran Canaria.

[32] Hartley R. (1993), Euclidean reconstruction from uncalibrated views. Second European Workshop on Applications of Invariance in Computer Vision. Pp 237-257

[33] Haralick R., Shapiro L. (1993). Computer and Robot Vision. Vol. 2, Addison-Wesley Publishing Company, Reading.

[34] Casals A. (1989). Sensor devices and systems for robotics. Vol. 52, NATO ASISeries, Springer, Berlin, Heidelberg.

[35] Abdel-Aziz Y.I., Karara H. M. (1971) Direct linear transformation into objectspace coordinates in close-range photogrammetry. Procedings symposium on close-range photogrametry, Urbana, Illinois, p. 1-18

[36] Ahlers R., Lu J. (1989) Stereoscopic vision – an application oriented overview, SPIE-Opt. Illumination, Image sensing Mach. Vision IV, p. 298-307

[37] Batlle J., Mouaddib E., Salvi J. (1998). A survey: recent progress in coded structured light as a technique to solve the correspondence problem. Int. J. Pattern Recognition 31, p. 963-982

[38] Zhang Z. (1993) The matching problem: the state of the art. Technical report No. 2146, Institute National de Recherche en Informatique et en Automatique.

[39] Zhang Z. (1998) A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research.

[40] Zhang Z. (2000) A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22, pp 1330-1334.

[41] Zhang Z. (2002) Camera calibration with one-dimensional objects. Technical Report MSR-TR-2001-120 Microsoft research.

[42] Luong Q., Faugeras O., Maybank S. (1992) Camera self-calibration: theory and experiments. Proceeding of the Europena Coference on Computer Vision, pp 321- 334.

[43] Maybank S., Faugeras O. (1992) A theory of self-calibration of a miving camera. International Journal of Computer Vision 8(2), pp 123-151.

[44] Ahmed M., Farag A. (2001) Non metric calibration of camera lens distortion. Proceedings of the International Conference on Image Processing. Greece. pp. 157-160

[45] Tsai R., (1997) A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-self TV camera lenses. IEEE Journal of Robotics and Automation, Vol RA-3 No 4. pp. 323-344

[46]Salvi J., Armangué X., Batlle J. (2002) A Compartive review of camera calibrating methods with accuracy evaluation. Pattern recognition Vol. 35, pp 1617-1635

[47] Sun W., Cooperstock J. (2004). Requeriments for camera calibration: Must accuracy come with a high price?. IEEE Workshop on Applications of Computer Vision.

[48]. P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schr¨oter, L. Murphy, W. Churchill, D. Cole, and I. Reid. Navigating, recognising anddescribing urban spaces with vision and laser. International Journal of Robotics Research (IJRR), 28(11-12), 2009.

[49]. K. Konolige and M. Agrawal. FrameSLAM: From bundle adjustment to real-time visual mapping. IEEE Transactions on Robotics, 25(5), 2008.

[50]. N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In ACM Transactions on Graphics (Proc. of SIGGRAPH), 2006.

[51]. K. Konolige. Projected texture stereo. In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2010.

[52]. PrimeSense. http://www.primesense.com/.

[53]. Canesta. http://www.canesta.com/.

[54]. Mesa Imaging. http://www.mesa-imaging.ch/.

[55]. Microsoft. http://www.xbox.com/en-US/kinect, 2010.

[56]. P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 14(2), 1992.

[57]. S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2000.

[58]. S. May, D. Dr¨oschel, D. Holz, E. Fuchs, S. Malis, A. N¨uchter, and J. Hertzberg. Threedimensional mapping with time-of-flight cameras. Journal of Field Robotics (JFR), 26(11-12), 2009.

[59]A. Akbarzadeh, J. M. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, H. Towles, D. Nist´er, and M. Pollefeys. Towards urban 3D reconstruction from video. In Proc. of the Third International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT), 2006.

[60]. D. Nister. An efficient solution to the five-point relative pose problem. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 26(6):756–77, 2004.

[61]. A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In Proc. of Robotics: Science and Systems (RSS), 2009.

[62]. D. Lowe. Discriminative image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 2004.

[63]. Y. Furukawa and J. Ponce. Patch-based multi-view stereo software (PMVS): http:// grail.cs.washington.edu/software/pmvs/.

[64]. H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In ACM Transactions on Graphics (Proc. of SIGGRAPH), 2000.

[65]. M. Krainin, P. Henry, X. Ren, and D. Fox. Manipulator and object tracking for in hand 3D object modeling. Technical Report UW-CSE-10-09-01, University of Washington, 2010. http://www.cs.washington.edu/ai/Mobile_Robotics/projects/hand_tracking/..

[66] OpenCV library, http://sourceforge.net/projects/opencv/ 2011

[67] T.F. Cootes and C.J. Taylor and D.H. Cooper and J. Graham (1995). "Active shape models - their training and application". Computer Vision and Image Understanding (61): 38–59.

[68]S. Belongie, J. Malik, and J. Puzicha (April 2002). "Shape Matching and Object Recognition Using Shape Contexts". IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (24): 509–521.. http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/belongie-pami02.pdf.

[69] G. J. S. H.-P. Liu Y., Stoll C. and T. C. Markerless motioncapture of interacting characters using multi-view image segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR’11), 14(1):1249–1256, 2011.

[70] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. Int. J. Comput. Vision, 70:109–131,November 2006

[71] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In ACM SIGGRAPH 2004 Papers, SIGGRAPH ’04, pages 309–314,New York, NY, USA, 2004. ACM.

[72 F. Zhou, F. De la Torre, and J. K. Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In IEEE Conference on Automatic Face and Gestures Recognition (FG), September 2008.

[73] M. Parizeau and R. Plamondon. A comparative analysis of regional correlation, dynamic time warping, and skeletal tree matching for signature verification.

[74] Open natural interface. November 2010. www.openni.net

[75] Automatic Digital Biometry Analysis System. Instituto de Fisioterapia Global Mezieres. www.adibas.es