Technical University of Catalonia

Department of Signal Theory and Communications

Stochastic Optimization and Interactive Machine Learning for Human Motion Analysis

Ph.D. Thesis Dissertation

by

Marcel Alcoverro Vidal

Submitted to the Universitat Politècnica de Catalunya

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Supervised by Dr. Montse Pardàs and Dr. Josep R. Casas

Ph.D. program on Signal Theory and Communications

July 2014

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola de Doctorat

Acta de qualificació de tesi doctoral

Curs acadèmic:

Nom i cognoms

Programa de doctorat

Unitat estructural responsable del programa

Resolució del Tribunal

Reunit el Tribunal designat a l'efecte, el doctorand / la doctoranda exposa el tema de la seva tesi doctoral titulada

_____

_____.

Acabada la lectura i després de donar resposta a les qüestions formulades pels membres titulars del tribunal, aquest atorga la qualificació:

☐ NO APTE ☐ APROVAT ☐ NOTABLE ☐ EXCEL·LENT

(Nom, cognoms i signatura) President/a	(Nom, cognoms i signatura) Secretari/ària	
(Nom, cognoms i signatura) Vocal	(Nom, cognoms i signatura) Vocal	(Nom, cognoms i signatura) Vocal

_____, _____ d'/de _____ de _____

El resultat de l'escrutini dels vots emesos pels membres titulars del tribunal, efectuat per l'Escola de Doctorat, a instància de la Comissió de Doctorat de la UPC, atorga la MENCIÓ CUM LAUDE:

☐ SÍ ☐ NO

(Nom, cognoms i signatura) President de la Comissió Permanent de l'Escola de Doctorat	(Nom, cognoms i signatura) Secretari de la Comissió Permanent de l'Escola de Doctorat

Barcelona, _____ d'/de _____ de _____

A la Lara,

sense tu això no hagués estat.

Summary

The analysis of human motion from visual data is a central issue in the computer vision research community as it enables a wide range of applications and it still remains a challenging problem when dealing with unconstrained scenarios and general conditions. Human motion analysis is used in the entertainment industry for movies or videogame production, in medical applications for rehabilitation or biomechanical studies. It is also used for human computer interaction in any kind of environment, and moreover, it is used for big data analysis from social networks such as Youtube or Flickr, to mention some of its use cases.

In this thesis we have studied human motion analysis techniques with a focus on its application for smart room environments. That is, we have studied methods that will support the analysis of people behavior in the room, allowing interaction with computers in a natural manner and in general, methods that introduce computers in human activity environments to enable new kind of services but in an unobstrusive mode. The thesis is structured in two parts, where we study the problem of 3D pose estimation from multiple views and the recognition of gestures using range sensors.

First, we propose a generic framework for hierarchically layered particle filtering (HPF) specially suited for motion capture tasks. Human motion capture problem generally involve tracking or optimization of high-dimensional state vectors where also one have to deal with multi-modal pdfs. HPF allow to overcome the problem by means of multiple passes through substate space variables. Then, based on the HPF framework, we propose a method to estimate the anthropometry of the subject, which at the end allows to obtain a human body model adjusted to the subject. Moreover, we introduce a new weighting function strategy for approximate partitioning of observations and a method that employs body part detections to improve particle propagation and weight evaluation, both integrated within the HPF framework.

The second part of this thesis is centered in the detection of gestures, and we have focused the problem of reducing annotation and training efforts required to train a specific gesture. In order to reduce the efforts required to train a gesture detector, we propose a solution based on online random forests that allows training in real-time, while receiving new data in sequence. The main aspect that makes the solution effective is the method we propose to collect the hard negatives examples while training the forests. The method uses the detector trained up to the current frame to test on that frame, and then collects samples based on the response of the detector such that they will be more relevant for training. In this manner, training is more effective in terms of the number of annotated frames required.

Resum

L'anàlisi del moviment humà a partir de dades visuals és un tema central en la recerca en visió per computador, per una banda perquè habilita un ampli espectre d'aplicacions i per altra perquè encara és un problema no resolt quan és aplicat en escenaris no controlats. L'analisi del moviment humà s'utilitza a l'indústria de l'entreteniment per la producció de pel·lícules i videojocs, en aplicacions mèdiques per rehabilitació o per estudis bio-mecànics. També s'utilitza en el camp de la interacció amb computadors o també per l'analisi de grans volums de dades de xarxes socials com Youtube o Flickr, per mencionar alguns exemples.

En aquesta tesi s'han estudiat tècniques per l'anàlisi de moviment humà enfocant la seva aplicació en entorns de sales intel·ligents. És a dir, s'ha enfocat a mètodes que puguin permetre l'anàlisi del comportament de les persones a la sala, que permetin la interacció amb els dispositius d'una manera natural i, en general, mètodes que incorporin les computadores en espais on hi ha activitat de persones, per habilitar nous serveis de manera que no interfereixin en la activitat.

A la primera part, es proposa un marc genèric per l'ús de filtres de partícules jeràrquics (HPF) especialment adequat per tasques de captura de moviment humà. La captura de moviment humà generalment implica seguiment i optimització de vectors d'estat de molt alta dimensió on a la vegada també s'han de tractar pdf's multi-modals. Els HPF permeten tractar aquest problema mitjançant multiples passades en subdivisions del vector d'estat. Basant-nos en el marc dels HPF, es proposa un mètode per estimar l'antropometria del subjecte, que a la vegada permet obtenir un model acurat del subjecte. També proposem dos nous mètodes per la captura de moviment humà. Per una banda, el APO es basa en una nova estratègia per les funcions de cost basada en la partició de les observacions. Per altra, el DD-HPF utilitza deteccions de parts del cos per millorar la propagació de partícules i l'avaluació de pesos. Ambdós mètodes són integrats dins el marc dels HPF.

La segona part de la tesi es centra en la detecció de gestos, i s'ha enfocat en el problema de reduir els esforços d'anotació i entrenament requerits per entrenar un detector per un gest concret. Per tal de reduir els esforços requerits per entrenar un detector de gestos, proposem una solució basada en online random forests que permet l'entrenament en temps real, mentre es reben noves dades sequencialment. El principal aspecte que fa la solució efectiva és el mètode que proposem per obtenir mostres negatives rellevants, mentre s'entrenen els arbres de decisió. El mètode utilitza el detector entrenat fins al moment per recollir mostres basades en la resposta del detector, de manera que siguin més rellevants per l'entrenament. D'aquesta manera l'entrenament és més efectiu pel que fa al nombre de mostres anotades que es requereixen.

Agraïments

Primer de tot, vull agrair a la Montse Pardàs i al Josep Ramon Casas haver-me donat l'oportunitat de fer aquest doctorat i de participar en els projectes de recerca de GPI, així com tota la seva dedicació durant els anys d'elaboració d'aquesta tesi. També agraeixo especialment a l'Adolfo López tots els consells, idees i debats que hem tingut durant aquests anys. La seva feina està present en diverses parts de la tesi, i el treball conjunt és el que ens ha permès superar els problemes que hem anat trobant. Un agraïment també a l'Albert Gil pel suport en programació, ús de recursos i la dedicació que m'ha donat per sortir dels diferents "atzucacs computacionals" que m'he trobat durant aquesta tesi. Gràcies també al Xavier Suau per les fructíferes discussions i aportacions per fer les interfícies gestuals més usables. Així mateix un agraïment a tots els companys de GPI per haver fet de tots aquests anys una molt bona estada, amenitzada a base de pastissos, tertúlies i partits de futbol.

Finalment, un agraïment als meus pares, que sempre m'han donat suport i motivació per tirar endavant la tesi, i que segur que hi tenen un pes molt important en tot això.

Contents

CHAPTER 1

Introduction

The analysis of human motion from visual data is a central issue in the computer vision research community as it enables a wide range of applications and it still remains a challenging problem when dealing with unconstrained scenarios and general conditions. Human motion analysis is used in the entertainment industry for movies or videogame production, in medical applications for rehabilitation or biomechanical studies. It is also used for human computer interaction in any kind of environment, and moreover, it is used for big data analysis from social networks such as Youtube or Flickr, to mention some of its use cases. The requirements of the analysis depend on the application. One may need the exact pose of the bones and articulations of the human body, or a classification of gestures, or the recognition of certain actions involving elements of a specific context. These issues have been studied from the beginning of computer vision research evolving in wide research fields such as pose estimation, motion capture or action recognition.

In this thesis we have studied human motion analysis techniques with a focus on its

application for smart room environments. That is, we have studied methods that will support the analysis of people behavior in the room, allowing interaction with computers in a natural manner and in general, methods that introduce computers in human activity environments to enable new kind of services but in an unobstrusive mode. The proposed methods have been used in three European and Spanish research projects: VISION [5], which focused on immersive communication systems, ACTIBIO [1] dedicated to activity recognition and biometrics and FASCINATE [2] dedicated to new forms of interaction with broadcasting content.

1.1 Objectives

Human pose conveys relevant information of what is happening in a visual scene and it is a basic description than can facilitate a higher level understanding of the scene, such as the actions, activities or events occurring in the scene. The pose can also be associated to other semantic meanings to enable new human computer interfaces, for example to activate actions in graphical user interfaces.

The objectives of this thesis are:

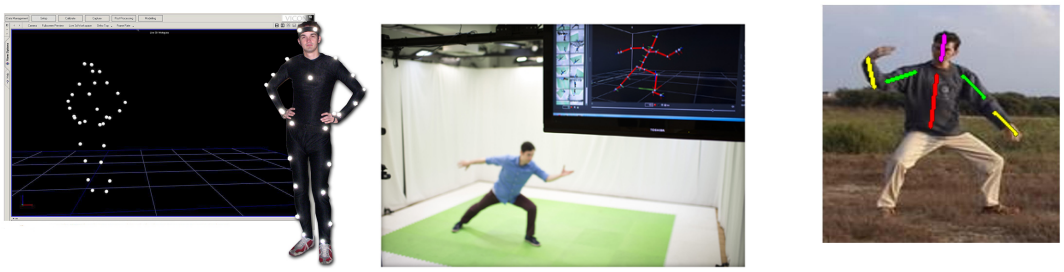
- Investigate techniques to obtain the pose of the subject in the scene, to provide such information to higher level applications.
- Study the application of the techniques in real world scenarios, thus with an emphasis in approaches robust to clutter, capable to perform with limited set of views.
- Propose methods that could be used by non experienced users, taking the approach closer to the end-user application.

We consider that these are main challenges in human motion analysis research, as it is a field that has reached certain level of maturity but methods are still hardly implemented out of laboratories.

1.2 System Input/Output Perspective: Human Pose from Visual Data

From the perspective of the input and output of the system, this thesis focuses on the computation or classification of the human pose given image data. On the one side, in the first part of the thesis, human pose is described by the position of the articulations or joints of the human body. The goal of the proposed methods is to determine these points or, equivalently, the rotational angles that adopts each skeleton bone with respect to the joint where it is attached. In the second part of the thesis, we restrict ourselves to a subset of human poses that involve certain configurations of arms and hands, which we denote as *static gestures*. The goal of the methods proposed in the second part is to localize and classify instances of these poses to the class they belong or, otherwise, to classify them as the background class. If we consider it from the point of view of the input data, we deal with color video data from multiple cameras and with depth data captured from a single camera. In the following we define these concepts within the context of this work.

Motion Capture The obtention of the position of the joints of the human body is commonly known as motion capture. Motion capture has been usually achieved by means of optical sensors and markers attached to the body of the performer. When the system uses color video cameras and the performer does not wear any marker, it is called Markerless Motion Capture (See Figure 1.1). The concept *pose estimation* is also widely used in computer vision literature, but it usually refers to methods that determine the pose of a subject in 2D from a single image, that is described with body part bounding boxes or stick figures (See Figure 1.1.c), which is not the goal of this thesis. Thus, we will refer to the methods proposed in this thesis as Markerless Motion Capture (MMC) systems. Motion capture data can be an input for activity recognition or can be used for avatar animation. Therefore, it is a basic step for several human motion analysis systems.



(a) (b) (c)

Figure 1.1: (a) Motion capture suit and application from VICON [4] (b) Markerless motion capture scene in the Organic Motion setup [3] (c) 2D pose estimation from a single image by Ferrari et. al [43]

Gesture Localization Pose estimation has also been approached as a pixel classification problem, where the goal of the method is to classify each pixel of the image as belonging to a body part. This approach obtains the pose with an ensemble of body part detectors or a single body part classifier [108]. Pixel -or part- based classification techniques have achieved best performance with the development of depth cameras. From depth data, the bones positions and joint angles can be obtained by exploiting the 3D information per pixel. These techniques have a strong background coming from the object detection and image parsing and classification research fields. The recognition of the hand and arms pose that conform a static gesture is a problem that can also be approached as pixel classification. In this thesis, we call this problem *gesture localization*. The methods we propose determine the position in the image and the type of gesture that is performed, which inherently is a hand and arms pose. Gesture localization is a building block of gesture-based human computer interfaces.

Multi-view video Capturing video from multiple cameras simultaneously has advantages with respect to monocular video, as it facilitates the extraction of 3D information from the scene and people, and allows to overcome occlusions. In this thesis, we make use of multi-camera setups for the motion capture experiments. Such settings require synchro-

nized cameras and a calibration [21] that will permit projections and back-projections to obtain 2D-3D correspondences. Some recording studios use a high number of cameras (8 to 16 cameras or more) with high frame rate ($\approx$ 60 fps) and high resolution, as their goal is to obtain accurate 3D reconstructions of the performers. In our case, we consider settings with lower number of cameras (4 to 6) and regular frame rate (25 fps), in order to propose solutions that may bring motion capture out of studios, in scenarios with less technical resources. Note that this introduces new challenges, as some approaches employed in the former case may not be applied to the latter case due to occlusions, lack of resolution and difficulties to deal with abrupt motion. The methods proposed in this thesis are specially designed to overcome these problems.

Depth cameras Depth cameras have recently evolved to provide good resolution at affordable cost. Common difficulties in human motion analysis when using color images such as illumination changes, color and texture variability, foreground background segmentation or scale, are not present or become straightforward when using depth data. Thus, recent methods proposed for motion capture use depth data to take advantage of its potential to describe human body shape and its parts. Motion capture using depth sensors is a very robust and accurate solution for certain scenarios, but depth sensors have a limited range (usually <5m), and its usage outdoors is still limited. Also, multi-camera setup is limited due to interferences between sensors. Thus, in outdoor scenarios or large spaces one should rely on color cameras. Even though, depth sensors are ideally suited for human computer interaction and we demonstrate its usage by means of a gesture localization method as described in Chapter 3.

1.3 Probabilistic Model Perspective: Generative and Discriminative Models

With respect to the probabilistic models adopted to infer the pose of the human body, one can rely on *generative* or *discriminative* models, or a combination of both [93]. In this thesis we have explored both approaches: in the first part we adopt a generative model and, in the second part, we rely on a discriminative model.

Generative model In the motion capture problem, we define a state representation (a 3D human model with joint positions or angles) and the goal is to compute the conditional distribution for the model state given image observations. When adopting a generative model, we require a constructive form of the observation (the observation likelihood or cost function) and the 3D model is explicitly used for inference. Then, the process consists in exploring the search space to locate the peaks of the likelihood function. The model state conditional distribution is obtained using the Bayes rule with the observation conditional and the state prior.

Discriminative model In discriminative methods the state conditional distribution is estimated directly, such that it simplifies inference. To do so, a supervised approach is adopted by using a set of examples of 3D human configurations paired with their images appearances (observations). Most of monocular pose estimation research follows discriminative approaches, in scenes where generative approaches fail due to the ambiguities present when only a single viewpoint of the actor is available. But in these cases, the performance of the discriminative approaches is limited either to viewpoints or to the type of actions present in the training examples of the dataset. The amount of data required to generalize in viewpoints and actions is massive and, even in the case when data was available, obtaining accurate results is still a problem that requires further study.

We rely on a generative model for motion capture in multi-camera setups, as it allows to cover an unconstrained set of actions, taking advantage of the multiple viewpoints to solve inference. On the other side, we adopt a discriminative model in the gesture localization task, as in this case the viewpoints are constrained to a frontal view, and the set of poses to recognize is limited to a small number of gestures.

Training data Human motion analysis by means of supervised approaches requires large amount of training data, mainly because pose, viewpoint and clothing introduce high variability in the appearance of body parts, which makes harder the generalization of detectors or classifiers. Collecting training data requires expensive setups, with optical sensors or range scanners, and recording long sequences of actors performing desired actions. Moreover, one should add the time dedicated to annotation tasks. This issue is relevant for all the techniques proposed in this thesis. In the first part of the thesis, dedicated to motion capture, a generative approach is proposed in order to overcome the difficulties of collecting the massive amount of training data required to employ supervised discriminative approaches. In the second part, an interactive learning approach for gesture recognition is proposed, which reduces considerably the amount of time needed to record, annotate and train the classifiers, by fusing all these steps in a single semi-automatic interactive loop.

1.4 Contributions

In this thesis we study the problem of 3D pose estimation from multiple views and from range sensors towards the obtention of full body pose or the recognition of gestures. The thesis is structured in two parts:

1. **Stochastic Optimization for Motion Capture** We focus on the problem of markerless motion capture from multiple views, and these are the contributions:

- An stochastic optimization framework called *layered particle filtering* (LPF) that generalizes existing approaches of the state-of-the-art and can be employed in distinct motion capture multi-modal multi-dimensional optimization problems.
- A method for human body anthropometry estimation from silhouettes in multiple views that uses the LPF optimization.
- An approach to approximate partitioning of observations used in LPF optimization towards motion capture from multiple views.
- A detector-driven approach in the LPF framework where body part detectors are used to guide the optimization.

2. **Interactive Machine Learning for Gesture Localization** We focus on the problem of gesture localization using depth data, and we propose an approach to reduce the efforts in the task of recording, annotating and training classifiers by using an interactive machine learning perspective. These are the contributions:

   - A depth clipping test for a random forests approach to gesture localization that improves performance in presence of background objects or clutter.
   - A gesture localization approach using online random forests with hard negatives mining by means of on-the-fly detection. Training is performed online and the time for training is considerably reduced. The experiments show improvements in performance compared to the offline version, when both are trained with datasets with a low number of examples.
   - An interactive machine learning method for gesture localization training using online random forests.

A comprehensive list of the contributions of this thesis, including related references of journal papers and conference publications is detailed in the conclusions chapter.

1.5 Thesis organization

The thesis is divided in two chapters that describe the contributions of each part.

In chapter 2, the Layered Particle Filtering stochastic optimization framework is described. Then, the contributions towards markerless motion capture using LPF are presented with their corresponding experimental results.

Chapter 3 presents the approaches towards gesture localization. First, we introduce the method to localize gestures on depth data using random forests and the depth clipping test and the corresponding experimental results. Then, we present the gesture localization method using online training and we compare it experimentally with the offline counterpart. Finally, we introduce the interactive machine learning approach.

Chapter 4 draws the conclusions of this dissertation.

CHAPTER 2

Stochastic Optimization for Markerless Motion Capture

The markerless motion capture task from multiple views in unconstrained scenarios is a complex problem that has been addressed by a great number of researchers. It is a high dimensional estimation problem where the observation models density distributions are multimodal due to the nature of image features together with viewpoint ambiguities and human body parts self-occlusions. Using a high number of cameras at high capturing frame rate the problem can be solved by means of local optimization, starting from the pose estimated at the previous frame. But the local optimization approach is not suitable when the number of views is low (4 to 6) and capturing frame rates are relatively low (25 fps) because the methods are not able to capture abrupt motion and self occlusions, and do not recover from errors. In this thesis, we propose an stochastic optimization framework that allows to deal with multimodal densities in high dimensional estimation problems. The framework, called *Layered Particle Filtering* (LPF) is explained in Section 2.2. The LPF method has been employed successfully for motion capture tasks in environments with

few cameras and unconstrained background.

One of the tasks where LPF is used is anthropometry estimation. In model-based markerless motion capture a human body model is needed in order to infer the pose of the subject by means of cost functions. Motion capture is improved if an accurate human body model of the subject being tracked is available. Some research studies assume that a body scan of the subject is available, and the methods rely on such scan to construct an articulated model useful for inference. For certain applications or scenarios such human body scans are not available. In this thesis, we propose a method to adjust a generic human body model to the subject being tracked using multiple views, so that we obtain the anthropometry and the articulated model adjusted to the subject. The method is presented in Section 2.3.

An advantage of using a layered approach to multi-dimensional optimization, where each layer focus to a substate space vector, is that the weighting functions can be configured to measure the cost in the specific subspace region of the layer. In human motion capture, measuring substate space variables (pose variables) independently is not possible. Due to the nature of the observation space (image features) weighting functions usually perform measurements along the whole observation space. In Section 2.4, we propose a method to partition this observation space such that weighting functions can measure approximately the cost of substate space variables. Experimental results show better accuracy in motion capture tasks, specially in presence of clutter or body part self-occlusions and ambiguities.

In particle filtering based estimation, usually the propagation of particles is governed by Gaussian diffusion, which makes the search blind with respect to evidence. In order to perform a more efficient search we propose the use of body part detections that will guide the propagation of particles. Such approach is presented in Section 2.5 together with experimental results that demonstrate better accuracy than an equivalent method using Gaussian diffusion with the same number of particles.

Part of the work presented in this chapter has been published in [8], [81] and [87]. This work has been done in collaboration with Adolfo López-Méndez.

Next section reviews related work on markerless motion capture.

2.1 Related Work

Markerless vision-based human motion analysis has received much interest over the last two decades, and it continues to be an active research domain. Several surveys have been written within the domain of human motion analysis [86], [51], [92] and they present different taxonomies. We can distinguish, as presented in a recent publication by Sigal and Black [112], generative methods (model-based), discriminative methods (model-free) and methods using part-based models (body parts detection). We will focus the review in generative approaches and in methods using body parts detection, as they are more related to our work.

In discriminative approaches, an explicit human body model is not available. In these cases, the pose estimation is based on machine learning techniques using training data, and current approaches do not generalize well to complex motions not present in the training data. Generative approaches, also called model-based approaches, use a body model and a synthesis-and-test strategy to infer the pose at each frame. We will describe the recent work in generative markerless human motion analysis in section 2.1.1.

Pose inference in a generative method requires solving an optimization problem in a high-dimensional space, which is a computationally expensive task. A better alternative for certain applications due to its reduced computational cost is the detection of body parts directly from image data and then inferring the pose with inverse kinematics. We review recent work in body parts detection and inverse kinematics in section 2.1.1.

The presence of objects or people occluding parts of the human body may interfere with the motion capture task. In section 2.1.2 we review recent methods concerning the

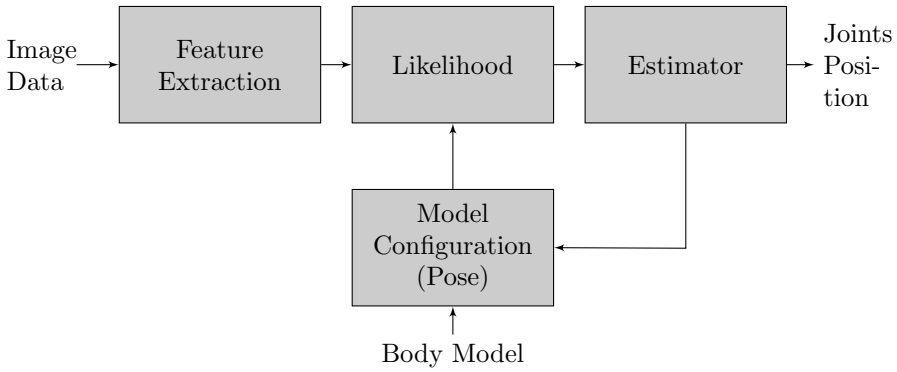


Figure 2.1: Generative MMC System Diagram

problem of occlusions.

2.1.1 Generative Markerless Motion Capture

In a generative MMC system, the pose estimation process can be split in a *modeling* phase and an *estimation* phase. The modeling phase consists in the construction of the likelihood function using image features, the human body model and a matching function. The estimation phase consists in finding the most likely pose using the likelihood function. Figure 2.1 shows a diagram of the components involved in a generative MMC system.

Modeling Phase

The parameters involved in the construction of the likelihood function are, first, those related to the body model used: the body configuration parameters, body shape and appearance parameters. The likelihood is also related to the kind of observations that we have, the camera viewpoint and the image features that we use. Model-based approaches use a body model which includes the kinematic structure and the body dimensions.

Human Body Models The human body models describe the kinematic properties and the shape and appearance.

Kinematic structure: Most of the models describe the kinematics of the skeleton as a tree, consisting of segments that are linked by joints. Every joint contains a number of degrees of freedom (DOF) depending on the possible rotations at the joint. The number of DOF of the skeleton varies between the methods from 25 DOF until more than 50 when considering hand and feet movements. The range of the rotation angles is usually restricted by kinematic constraints, in order to limit the pose space to human feasible poses. Constraints to avoid interpenetration of the body parts also are useful, but in this case we need additional information of the shape of the body.

Model shape and appearance: The human body shape is modeled in different ways.

On one side, we have models based on separate rigid shapes, as cylinders [36], ellipsoids [85], and more generally as a combination of tapered super-quadrics [52]. These volumetric shapes depend on only a few parameters.

On the other side, surface-based models employ a single surface for the entire body. These models consist usually on a mesh of polygons than can be deformed by changes on the underlying skeleton. Several studies use a mesh obtained from a 3D scan (Cyberware technology [32]) of the specific individual to track, as in [14], [28]. Vlasic et al. [123] use either meshes from a 3D scan, or meshes obtained from a 3D multiview stereo reconstruction algorithm. Anguelov et al. [10] propose the SCAPE method to model deformable surfaces, which is based on models of pose and body shape variation that are learned from a database of 3D scans. The SCAPE model is used for human motion analysis in [13]. Bandouch et al. [15] use the RAMSIS model [115] which is widely used in the automotive industry. Its design has been guided by ergonomic considerations and it is precise in the inner joint locations. Plänkers and Fua [91] use a more complex model based on three layers: skeleton; ellipsoidal balls to simulate muscles and fat tissue; and a polygonal surface representation for the skin.

The deformation process of the skin mesh according to the pose parameters is performed

using techniques such as *Skeletal Subspace Deformation* also called *Linear Blend Skinning* (LBS) [63]. LBS determines the new position of a vertex of the mesh by linearly combining the results of the vertex transformed rigidly with each bone. A scalar weight is given to each bone, and the weighted sum gives the final vertex position. LBS has a number of shortcomings, such as the *"elbow collapsing"* effect. The technique is improved by using data-interpolation techniques using a number of example meshes, as proposed in [76] and [114], which allow correcting the error in the vertex positions introduced by the LBS. Some models, as for example the SCAPE model [10], obtain the example meshes from a database of 3D scans of the same individual adopting distinct poses. Then, the deformation model is based on a linear blend skinning corrected according to some parameters learned from the database. Vlasic et al. [123] improve the LBS results using an iterative framework that deform the shape to better approximate the silhouette contours of the actual image data.

Model initialization: In case of using scanned models the parameters of the shape model (lengths, widths, etc.) may be assumed fixed. However, when using a generic model, the parameters should be adjusted to the specific individual in order to avoid innaccurate pose estimations. Carranza et al. [26] propose an initialization phase to adjust these parameters, where the subject has to adopt a specified pose. Plänkers and Fua [91] perform the estimation of the parameters during motion over a sequence to determine more precisely the position of the articulations inside the skin surface.

Image Features and Matching Functions We are interested in methods that recover the kinematic configuration of a person, using image data captured from several views. The appearance of people in images varies due to distinct clothing or lighting. To generalize over these varying conditions, motion tracking systems use cost functions which evaluate the model configuration against extracted image features or descriptors.

Several approaches use silhouettes of the active people in the scene as image descriptors

to define the matching functions [36], [14], [26]. Silhouettes are insensitive to variations in the surface, such as color, and encode great information to recover 3D poses. The matching functions account for the difference between the projection of the model into a certain view and the silhouette in this view. However, sometimes it is not possible to recover certain DOF, when using only silhouettes, due to the ambiguities inherent in the shape described by a silhouette.

Edges are image descriptors also used to construct the matching functions [36]. Edges appear in the image at the sharp changes in intensity. Within a silhouette usually provide a good outline of visible arms and legs. However, when the subject is wearing baggy clothes edges may lose usefulness. Matching functions take into account the normalized distance between model's synthesized edges and the closest edge found in the image.

Edges and silhouettes lack depth information which makes hard to detect self-occlusions. A 3D reconstruction can be created from the silhouettes of each view, and then use this reconstruction to build a matching function. The visual hull is the volume obtained from the intersection of the silhouette cones [75]. Several methods use matching functions that account for the intersection between the model and the visual hull of the individual [123], [28], [85]. Instead of using the visual hull, Plänkers and Fua [91] use depth maps, computed by stereometry using two or more cameras. The matching function in this case is based on closest point distance between the model and the points extracted from the depth map.

Motion measurements such as optical flow are also used as image descriptors. The optical flow information consists in a set of correspondences between pixels of consecutive frames. The pixel displacement in the image is used by Bregler et al. [23]. Ballan and Cortelazzo [14] compare the pixel correspondences in consecutive frames with the projection of the vertices of a deformable model in the same consecutive frames.

Color and texture is also used to describe the appearance of individual body parts. Skin color can be a good cue for finding head and hands and is used as feature by López

and Casas [79].

Image descriptors are usually combined to build more robust likelihood functions. Silhouettes are combined with edges [36], with optical flow [14], with depth maps [91] or color [79].

Estimation Frameworks

The estimation process is concerned with finding the set of pose parameters that maximize the likelihood function. Often, instead of a likelihood function, we define a cost function or error function, and we search for minima instead of maxima. In the following we will consider that we search for minima.

Cost functions usually have many local minima. Also, the dimensionality of the search space is high. Some methods rely on a estimation based on a single hypothesis, focusing on the efficiency of a local search. On the other side, we have approaches maintaining multiple hypothesis in order to reduce the probability of getting stuck at a local minimum.

Single Hypothesis Tracking Assuming that the time between subsequent frames is small, the distance between body configurations is likely to be small as well. This assumption provides us with an initial pose estimate, using the estimated pose of the previous frame. Single-hypothesis based tracking performs a local search around the initial pose estimate.

Several authors use local-optimization methods to perform the tracking. A common approach is to define an objective function in a least-squares framework [14], and then minimize the function using a gradient descent approach, using for example the Levenberg-Marquardt method [94]. Carranza et al. [26] use Powell's method and a simple downhill method [94] that does not need derivatives of the objective function. In our previous work in [7], we compared Powell's method with a multiple hypothesis tracking approach.

Delamarre and Faugeras [33] perform the search in the image domain, using forces between extracted silhouettes and the projected model to refine the pose estimation. Stoll et. al. [116] propose a method that performs nearly real-time with very accurate results, using local optimization and a body model based on spatial Gaussians specially suited for computational efficiency.

Multiple Hypothesis Tracking Single hypothesis tracking may introduce an accumulation of errors. If a wrong pose is obtained, due to ambiguities such as self-occlusions, maintaining a single hypothesis may propagate the error, and the recovery becomes difficult. To solve this problem, several approaches maintain multiple hypothesis.

Sampling-based approaches, such as particle filtering [62], [12], are able to track non-linear motion, as in the human motion case. In a particle filtering scheme, each particle or hypothesis has an associated weight, that is updated according to the cost function. The particles are propagated in time according to certain dynamics and including a noise component. In the case of human motion, the high dimensionality requires the use of many particles to sample with sufficient density the pose space.

A solution to deal with the high dimensionality is to spread the particles efficiently where a local minimum is more likely. For example, Deutscher et al. [36] use simulated annealing to focus the particles to the global maxima of the posterior. Another solution to the problem of the dimensionality is to partition the space into a number of lower-dimensional spaces [83], [25], considering the underlying hierarchical structure of the kinematic tree.

Gall et al. propose a framework in [48] that combines a multiple hypothesis method with local optimization and provide an experimental comparison of the method with particle filtering, annealed particle filtering and local optimization methods.

Body Parts Detection and Inverse Kinematics An alternative to full-body synthesis-and-test strategies, is to detect body parts and then infer the skeleton pose using this in-

formation. Some of these techniques are widely studied in the field of pose detection from a single image. An advantatge of those approaches is that they do not require initialization. Besides, in case of ambiguites due to lack of information, as in monocular settings, they can provide better results than synthesis-and-test tracking strategies.

Several methods use part-based models. In these methods, the human body is represented as a probabilistic graphical model, where nodes of the graph represent body parts and the edges model the relation between these distinct parts. Body parts may be detected using discriminative techniques widely studied in the field of object detection [122]. Also, some methods generate hypothesis for the body parts locations and evaluate the likelihood of the distinct body parts independently. Pictorial structures [42] is a part-based model commonly used [96][113]. Recently, Bergtholdt et. al. [18] presented a generic framework that allows detection of any class of part-based objects, which achieve relevant results in human body pose detection in a multiview setting. Sigal et. al. [110] presented a model specific for human body parts detection that is included in a tracking framework: it exploits temporal and spatial information by modeling the relationships between joints at a given time instant and over time. In [98] the detection of body parts is performed using depth maps obtained from range scans. They use a detector that matches the 3D model parts with the surface and edges discontinuites in the depth data.

In order to take advantage of the efficiency of body part detections, Ganapathi et. al. [50] include the detection of limbs and head in a synthesis-and-test tracking framework. The method achieves real-time performance and uses a single time-of-flight camera. The body parts detection used is based on geodesic distances computed from the depth maps captured [90]. Then, they propose probabilistic inverse kinematics to combine the detections with a generative tracking approach. In case of failure of detection of a part, the pose is estimated only considering the generative part of the system. Also based in a single depth camera, Shotton et. al. [108] presented a method for detection of body parts from single depth

images, that performs in real-time with accurate results, and is the building block of the pose recognition system implemented in the Microsoft Kinect SDK.

This kind of methods described above require a training phase using annotated data in order to learn the models parameters or to detect and classify the distinct body parts. The motion capture method proposed by Correa et. al. [29] extracts positions of the body extremities from 2D silhouettes using geodesic distances and mathematical morphology and does not require any training data. Alternatively, the extraction of the position of the extremities may be computed from 3D data obtained from multiview reconstruction or range scans. Extraction of features points from 3D data as polygonal meshes has been studied as a requirement for computer graphics methods for segmentation or shape matching and retrieval. Some authors propose to extract feature points from meshes using geodesic distances [67][132]. Hu et. al. [61] presented a method for extraction of salient features in the spectral domain using the Laplace-Bertrami operator.

Focusing in 3D interaction through full body movements, several methods rely on detecting the 3D position of hands, head and feet and then recovering the body posture using inverse kinematics. The inverse kinematics problem is under-constrained, so the resulting pose will depend on the inverse kinematics method employed. In [22], Boulic et. al. evaluate the efficiency and accuracy of several numerical and analytical methods. Jaume-i-Capó et. al. [64] propose a method to add image constraints to the inverse kinematics estimation such that image features guide the reconstruction of the pose. An alternative to numerical or analytical inverse kinematics solvers is to formulate the problem in a probabilistic framework, such as using inverse kinematics particle filtering [30]. In this case, constraints (e.g. image constraints or arbitrary constraints) may be considered also in the pose estimation. The method proposed in [58] performs tracking of the distinct end-effectors using particle filters, and then solves the pose using inverse kinematics with a numerical approach. In this manner, the number of DOF of the tracking search space is reduced which results in

a reduction in the computational cost.

2.1.2 Tracking in Cluttered Scenes

Although several probabilistic methods presented in the previous sections are motivated to face the problems that appear in cluttered scenes, we review in this section some papers that have presented techniques focusing specifically in the problem of occlusions or multiple people interacting in the scene.

Bandouch et. al. [16] propose solutions to deal with dynamic objects and with static occluders in a fullbody tracking framework based on foreground silhouettes. In this method, they use a color-based appearance model for the human body that is used to mask out the foreground detections of objects that do not resemble the human model appearance. For static objects they manually set masks in the image regions that are candidate for occlusion by static objects in the scene such as tables.

Egashira et. al. [40] focus on motion capture of multiple people interacting. The method proposed performs a 3D voxel based reconstruction using foreground silhouettes, and labels the voxels that belong to each person using color information. Then, a body tracking system for a single person is used using the multiview projection of the segmented 3D voxel data.

In multiview settings, the usage of 3D data obtained by reconstruction methods facilitates the task of resolving occlusions caused by other people or moving objects present in the scene. Unfortunately, 3D reconstruction methods tend to fail in presence of static occluders. Some authors have presented reconstruction methods robust to foreground detection misses, as it occurs in case of presence of occluders, by using the reprojection error [74][56]. From another point of view, some methods reconstruct the static occluders by analysing the foreground detection misses during a period of time [68][54]. Schick et. al [104] propose a fast reconstruction method using GPU that considers static occluders if its

3D position is known in advance.

A distinct approach focusing the problem of background clutter, dynamic background or occlusions is to perform segmentation and pose estimation jointly in an iterative manner [99][73]. In these methods, the shape of the model is used as a prior to improve the image segmentation, and then the image segmentation is used to estimate the pose. The segmentation is accurate as it uses a precise shape model of the object to segment. Also, these methods allow to give more relevance to the shape prior than to the image contours in order to improve robustness to occlusions. In [57], a method for joint segmentation and pose estimation is used for human motion capture using moving cameras.

2.2 Stochastic Optimization Framework

Markerless motion capture has been widely studied from a visual tracking perspective. In this context, a common approach is to formulate the problem as a Bayesian tracking problem. Let us denote the configuration of the target at time t as $\mathbf{x}_t$ (e.g. human body pose parameters at time t), which evolve according to an underlying stochastic process

$$\mathbf{x}_{t+1} = f_t(\mathbf{x}_t) + \mathbf{v}_t, \tag{2.1}$$

where f_t is a possibly nonlinear function and $\mathbf{v}_t$ is noise. Measurements acquired up to frame t are denoted as $\mathcal{Z}_t = \mathbf{z}_1, \ldots, \mathbf{z}_t$, with

$$\mathbf{z}_t = h_t(\mathbf{x}_t) + \mathbf{n}_t, \tag{2.2}$$

where h_t is again a possibly nonlinear function and $\mathbf{n}_t$ is noise.

The Bayesian tracking problem, also called the filtering problem, consists in applying Bayes theorem at each time step, obtaining a posterior $p_t(\mathbf{x}_t|\mathcal{Z}_t)$ based on available

observations as

$$p_t(\mathbf{x}_t|\mathcal{Z}_t) = \frac{p_t(\mathbf{z}_t|\mathbf{x}_t)p_{t-1}(\mathbf{x}_t|\mathcal{Z}_{t-1})}{p_t(\mathbf{z}_t)}, \tag{2.3}$$

where we can write $p_t(\mathbf{z}_t|\mathbf{x}_t)$ instead of $p_t(\mathbf{z}_t|\mathbf{x}_t, \mathcal{Z}_{t-1})$ because the measurements $\mathcal{Z}_t$ are assumed to be conditionally independent. A model for the expected motion or the dynamics of the system is introduced with the form of a conditional distribution $p_t(\mathbf{x_t}|\mathbf{x}_{t-1})$ such that eq. (2.3) becomes

$$p_t(\mathbf{x}_t|\mathcal{Z}_t) = \frac{p_t(\mathbf{z}_t|\mathbf{x}_t)\int p_t(\mathbf{x_t}|\mathbf{x}_{t-1})p_{t-1}(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1})d\mathbf{x}_{t-1}}{p_t(\mathbf{z}_t)}. \tag{2.4}$$

Particle filters [12] are introduced to approximate the solution of this equation for the case of non-Gaussian observation densities. The idea behind particle filtering is to simulate the operations on density functions in eq. (2.4) by means of a *weighted particle set*: a list of n pairs $(\mathbf{x}^{(i)}, \pi^{(i)}), i = 1, \ldots, n$, where $\mathbf{x}^{(i)} \in \mathcal{X}$ (the configuration space) and $\pi \in [0, 1]$ with $\sum_{i=1}^{n} \pi^{(i)} = 1$. The particle set represents a probability distribution $p(\mathbf{x})$ in the sense that choosing one of $\mathbf{x}^{(i)}$ with probability $\pi^{(i)}$ is approximately the same as drawing a random sample from the distribution $p(\mathbf{x})$. Such representation is convenient as it allows to easily perform operations (e.g. products, convolution) between distributions that are not Gaussian.

In a markerless motion capture system, models for f_t, h_t and noise ($\mathbf{v}_t$, $\mathbf{w}_t$) usually are not available, so methods rely on approximations or weak models which lead to a poor performance of particle filtering for the motion capture task. Although models for the observation process and noise are difficult to obtain, it is easier to design a weight function w_t which measures the quality of particles using image features. In this manner, methods combine particle filtering with optimization techniques in order to guide the particles to the maximum of such weight function w_t. Such approaches are not solving anymore the filtering problem but an optimization problem, so can be seen as *multiple-hypothesis stochastic*

optimization methods.

So, markerless motion capture can be formulated also from an optimization perspective. An error function measures the correspondence between the human body model and input images, and the goal is to find the optimal model parameters $\mathbf{x} \in \mathcal{X}$(the configuration space) that minimize such function. Such error function can be formulated inversely such that it behaves as a weight function (i.e. w_t introduced above), so the goal would be to maximize the weight function. In optimization literature, authors usually formulate the problem as a minimization problem. In this work, in general we will formulate the problem as maximization of a weight function, but we may use minimization when introducing well-known optimization techniques.

Local optimization methods such as gradient descent approaches may work if the starting point is close to the global minimum, but if it is not the case, the solution obtained by these methods frequently would be a local minimum far from the best solution. In the articulated tracking problem, the pose parameters of previous frames in the sequence may be used as starting point in current frame pose optimization. In case of fast movements or occlusion the optimization error will accumulate.

In this work, we build on multiple-hypothesis stochastic optimization methods in order to exploit the advantages of filtering based approaches such as temporal correlation, robustness to noise and recovering from errors, combined by the precision obtained by optimization techniques when reaching the maximum of the weight function. We will first describe more in detail the particle filtering principles to then proceed to stochastic optimization based on these principles.

2.2.1 Particle Filtering

Particle filtering is a Monte Carlo or random sampling method to recursevely estimate the posterior $p_t(\mathbf{x}_t|\mathcal{Z}_t)$ (eq. 2.4) based on the *importance sampling* technique. The posterior is

represented by a weighted particle set $(\mathbf{x}_t^{(i)}, \pi_t^{(i)}), i = 1, \ldots, N_s$ so that the posterior density can be approximated as

$$p_t(\mathbf{x}_t|\mathcal{Z}_t) \approx \sum_{i=1}^{N_s} \pi_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}). \tag{2.5}$$

Following the importance sampling principle, samples $\mathbf{x}_t^{(i)}$ are generated from a proposal distribution $q(\mathbf{x}_t|\mathcal{Z}_t, \mathbf{x}_{t-1})$ from which it is easy to draw samples. The weights are computed recursively as

$$\pi_t^{(i)} \propto \pi_{t-1}^{(i)} p_t(\mathbf{z}_t|\mathbf{x}_t) r_t^{(i)} \tag{2.6}$$

where $r_t^{(i)}$ is a factor given by

$$r_t^{(i)} = \frac{p_t(\mathbf{x_t}|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathcal{Z}_t, \mathbf{x}_{t-1})} \tag{2.7}$$

that corrects the bias introduced by sampling from a wrong distribution. It is a common choice to use the motion prior as a proposal distribution (i.e $q(\mathbf{x}_t|\mathcal{Z}_t, \mathbf{x}_{t-1}) = p_t(\mathbf{x_t}|\mathbf{x}_{t-1})$) so that $r_t^{(i)} = 1$ and weights are easily evaluated. The drawback of this choice is that the proposal distribution is blind to the current observation, so samples may be placed far from the modes of the likelihood and in consequence from the modes of the posterior.

As introduced above, in the motion capture task, the likelihood is approximated by a weighting function designed to achieve maximum values when a human body model fits observation data. Moreover, the motion prior and proposal distribution are also design choices validated experimentally because its models are not available in a closed form.

In this work, we propose methods based on particle filtering principles, but we rely also on approximated forms of the likelihood, motion prior and proposal. For this reason we formulate the technique from a more generic perspective, where the common particle filtering run is considered a building component for further development of stochastic optimization methods. We follow the concepts introduced by Gall et. al [49] about inter-

active particle filtering strategies, but we rely on heuristically justified algorithms rather than studying the convergence and mathematical properties of the transition kernels and weighting functions as it is done in [49].

In this context, the generic particle flltering run requires two main design choices:

- to define the Markov transition kernels K_t on $\mathcal{X}$.
- to define the weighting functions $w_t : \mathcal{X} \mapsto \mathbb{R}$.

With such functions defined, the basic algorithm, which runs for each iteration parameter t, can be divided in three steps as described in algorithm 1:

Algorithm 1: Generic particle filtering run (`GPFRun`)

Input: initial unweighted particle set $(\hat{\mathbf{x}}_t^{(i)})_{i=1,\dots,N_s}$, K_t, w_t

Output: next unweighted particle set $(\hat{\mathbf{x}}_{t+1}^{(i)})_{i=1,\dots,N_s}$

1. *Update*:
   - Set $\pi_t^{(i)} \leftarrow w_t(\hat{\mathbf{x}}_t^{(i)})\ \forall i$
   - Set $\pi_t^{(i)} \leftarrow \frac{\pi_t^{(i)}}{\sum_{j=1}^{n} \pi_t^{(j)}} \forall i$
2. *Resample*:
   - Set $(\mathbf{x}_t^{(i)})_{i=1,\dots,N_s}$ by drawing N_s times with replacement from the set $(\hat{\mathbf{x}}_t^{(j)})_{j=1,\dots,N_s}$ with probabilities $\pi_t^{(j)}$
3. *Propagate*:
   - Sample $\hat{\mathbf{x}}_{t+1}^{(i)}$ from $K_t(\mathbf{x}_t^{(i)})\ \forall i$

return $(\hat{\mathbf{x}}_{t+1}^{(i)})_{i=1,\dots,N_s}$

Note that in general, the weighting function w_t will depend on images $\mathcal{Z}_t$, but also

the kernel K_t may consider $\mathcal{Z}_t$ in order to take into account data in the propagation step. Following an optimization perspective, the propagation step is related with the concept of *exploration* of the search space used in metaheuristics optimization literature [20]. The update step evaluates the samples with the actual weight function. The resample step reduce the so called effect of *degeneracy* of the particles: after several iterations of propagate and update steps, a large proportion of the particles would have negligible weight, and the computational effort would focus in updating particles with no contribution to the posterior approximation. By introducing the resample step, such degeneracy problem is avoided, particles that have small weights are eliminated and replaced by particles with large weights. The resample step takes the role of *exploitation*, a term introduced in metaheuristics optimization algorithms design, that is to focus the search in local regions of the search space knowing that a good solution is found in this region. In fact, the generic particle filtering algorithm has close similarities with Particle Swarm Optimization [69], a metaheuristics optimization method which also makes use of a particle system. These similarities have been described by Zhang et. al. [130] in the context of visual tracking. Figure 2.2.1 shows a graphic example of the operation of the steps of a generic particle filtering run.

The baseline kernel and weighting functions that would be used for motion capture tasks throughout this work are the following functions:

- **Gaussian kernel function**: the simpler approach to propagate particles would employ a Gaussian transition kernel. Thus, in this case $K_t(\mathbf{x}_t^{(i)}) = \mathbf{x}_t^{(i)} + \mathcal{N}$, where $\mathcal{N}$ would be a multi-variate Gaussian distribution with zero mean and a diagonal covariance matrix.

- **Silhouette XOR weighting function**: the basic weighting function for motion capture tasks would be the Silhouette XOR weighting function. Consider input

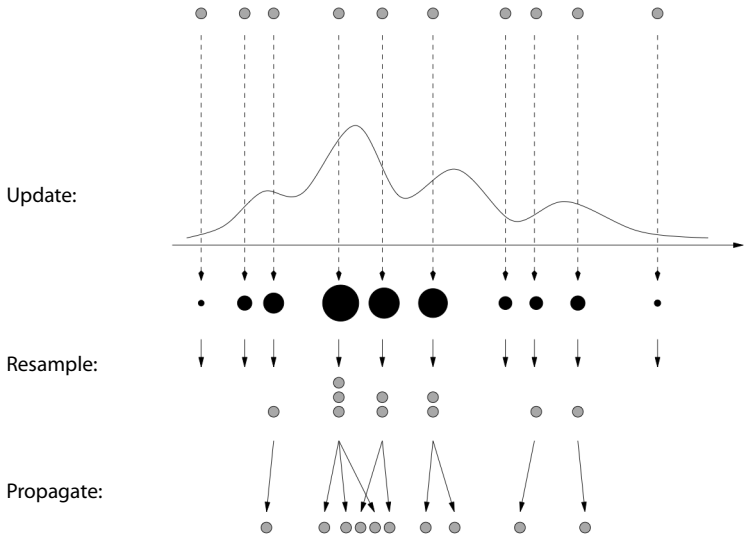


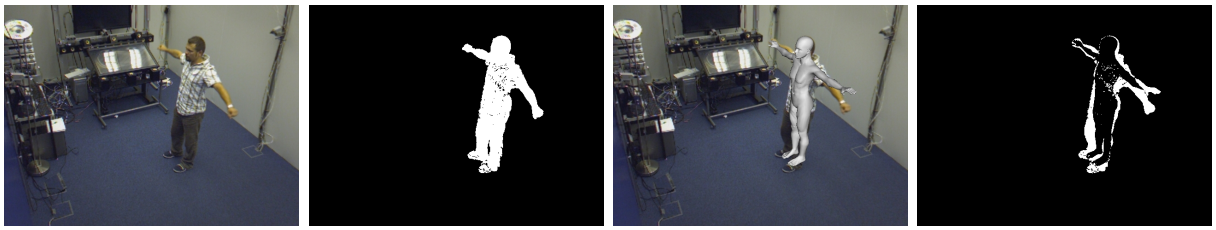
Figure 2.2: Graphical example of the generic particle filtering steps. Extracted from [62].

images from which the silhouette of the subject can be computed, for instance through background subtraction. Then the weighting function is defined in terms of a pixel-wise XOR between the silhouette $\mathcal{S}_v$ in each view v and the projection of the outer shape of a human body model in the corresponding views, $\mathcal{M}_v$, configured according to the pose parameters $\mathbf{x} \in \mathcal{X}$. Then we have the following definition for the weighting function

$$w_t = exp(-\sum_v \frac{1}{\mathcal{C}(\mathcal{S}_v)} XOR(\mathcal{M}_v^i, \mathcal{S}_v)) \tag{2.8}$$

where XOR denotes the number of non-zero pixel values after a pixel-wise XOR and $\mathcal{C}(\mathcal{S}_v)$ is the number of pixels of the silhouette on the v-th view. Figure 2.3 shows an example of pixel-wise XOR.

A well known problem of the standard particle filtering method is that it fails to ap-



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

Figure 2.3: Silhouette XOR weighting function example. (a) Subject view. (b) Foreground silhouette. (c) Model outer shape superposed. (d) Silhouette pixel-wise XOR

proximate well the *pdf*s when dealing with a high dimensional state space. The number of particles required to represent well the pdf grows exponentially respect to the number of dimensions, which makes the method computationally infeasible. This is the case for body tracking approaches as the number of variables involved in the human body pose is high. Two main strategies have been proposed to overcome this drawback: the *annealed particle filter* (APF) [36] and *partitioned sampling* (PS) [83]. We describe in the following sections the main ideas of both techniques as they will be used throughout this work.

2.2.2 Annealing strategy

Deutscher et. al. [36] proposed a particle based stochastic method, the annealed particle filter, which uses the ideas of simulated annealing [71] within a particle filtering framework. The strategy focuses on moving the particles towards the maximum of the weighting function w_t. This is achieved by performing several particle filtering runs, where particles are weighted by smoothed versions of the weighting function. Such smoothing factor is decreased while running iterations such that at first the influence of local maxima is reduced, but it increases gradually to obtain more peaked functions at the end. This help that particles do not get stuck in local maxima.

The common way to obtain this behavior is by defining a series of weighting functions

of the form

$$w_t^k(\mathbf{x}) = w_t(\mathbf{x})^{\beta(k)} \tag{2.9}$$

where $0 < \beta(k) \leq 1$ is a monotonic increasing function that determines the annealing rate, so as k grows, $\beta(k)$ grows producing more peaky functions.

The APF is presented for body tracking application. In this tracking context, the method performs a particle filtering run for each layer $k = 0, \ldots, K$ and the whole set of layered filtering runs is repeated for each time step t. The method is designed to make a distinction between the propagation at layers $k \in [0, K-1]$, which relates to the amount of diffusion of particles in order to reach the local maxima, and the propagation at layer $k = K$, which involves the propagation between time steps, which is related with the body motion model. Let us term them *inter-layer propagation* and *inter-frame propagation* respectively in order to make more clear the distinction. We describe the APF method with the formulation introduced for the generic particle filtering (in algorithm 2), where the inter-layer propagation is achieved by means of a series of transition kernel denoted as K_t^k for $t \in \mathbb{N}$ and $k \in [0, K-1]$, and the inter-frame propagation is denoted with a series of transition kernels $(K_t)_{t\in\mathbb{N}}$. Note that it uses the function `GPFRun` presented in algorithm 1.

Algorithm 2: Annealed particle filtering run (`APFRun`)

Input: initial unweighted particle set $(\hat{\mathbf{x}}_t^{(i)})_{i=1,\ldots,N_s}$, $(K_t)_{t\in\mathbb{N}}$, $(w_t^k)_{t\in\mathbb{N}}$, $(K_t^k)_{t\in\mathbb{N},k\in[0,K-1]}$

Output: next unweighted particle set $(\hat{\mathbf{x}}_{t+1}^{(i)})_{i=1,\ldots,N_s}$

1 **foreach** $k \in [0, K-1]$ **do**

2 $\quad (\hat{\mathbf{x}}_t^{(i)})_{i=1,\ldots,N_s} \leftarrow$ `GPFRun(` $(\hat{\mathbf{x}}_t^{(i)})_{i=1,\ldots,N_s}$, K_t^k, w_t^k`);`

3 $(\hat{\mathbf{x}}_{t+1}^{(i)})_{i=1,\ldots,N_s} \leftarrow$ `GPFRun(` $(\hat{\mathbf{x}}_t^{(i)})_{i=1,\ldots,N_s}$, K_t, w_t^K`);`

4 **return** $(\hat{\mathbf{x}}_{t+1}^{(i)})_{i=1,\ldots,N_s}$

In the original APF proposed in [36] the transition kernels for inter-layer propagation

K_t^k are multi-variate Gaussian kernels with a covariance matrix specific for each layer as

$$\mathbf{\Sigma}^k = \alpha_0 \times \cdots \times \alpha_{k-1} \times \mathbf{\Sigma_0} \tag{2.10}$$

where $\alpha_0, \ldots, \alpha_k < 1$ are a *variance decay rate*, and $\mathbf{\Sigma_0}$ is the default variance that accounts for reasonable motion from frame to frame. Such variance decay rate is related with the rate of annealing, with the idea that at initial layers the diffusion applied to particles is greater, and as k grows the variance of the Gaussian diffusion is reduced to concentrate the particles at the maxima of the weighting function, just when it becomes more peaked.

2.2.3 Partitioned sampling strategy

Partitioned Sampling (PS) was introduced by McCormick et al. [83] to overcome the problem of tracking in high-dimensional state space. The approach is based on dividing the state space into partitions an sequentially applying simple particle filter runs to each partition. In such way, the number of particles required to sample the state space is reduced. Given a partitioning of the state space as a Cartesian product $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_k$ with k the number of partitions, the following conditions should hold:

1. The dynamics model $p_t(\mathbf{x_t}|\mathbf{x}_{t-1}) \equiv d$ can be decomposed as $d = d_1 * \ldots * d_k$ with each d_j acting on $\mathcal{X} = \mathcal{X}_j \times \ldots \times \mathcal{X}_k$. For example, given $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$, such condition means that if $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ and $\mathbf{x}' = (\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3)$ with $\mathbf{x}_i, \mathbf{x}'_i \in \mathcal{X}_i$, and $\mathbf{x}'$ is a random draw from $d_2(\cdot|\mathbf{x})$, then $\mathbf{x}'_1 = \mathbf{x}_1$. By means of such partitions of the dynamics, dynamics at each filtering run do not change the values of the projection of any particle into the preceding partitions of the configuration space.

2. Weighting functions $w_1, w_2, \ldots, w_{k-1}$ are available with w_j peaked in the same region as the posterior restricted to $\mathcal{X}_j$.

Such partitioning strategy in the dynamics model is convenient for articulated objects tracking because pose variables have an inherent ordering within kinematic chains. By setting the partitioning according to the kinematic chain ordering the first condition for dynamics is straightforward. The second condition is a hard requirement because designing weight functions that measure the matching of a particular part of the object would require a segmentation of the input data features to separate the features belonging to each part, which is not easily achieved in unconstrained image-based tracking conditions.

Duffner et al. [39] proposed a relaxed partitioning strategy in order to allow more precise weight functions. The approach performs a partitioning related to the available image feature cues, such that subspaces of the state space may not be independent (in PS they are independent) but are grouped such that a weight function can measure with reliability the particle variables for the particular subspace. In order to manage the fact that substates are no more independent they introduce a prior in the dynamical model to define the pair-wise interaction between substate spaces [39].

2.2.4 Layered Particle Filtering

Both strategies, annealing and partitioned sampling, can be combined in a single method to take profit of its benefits. On one side to move particles towards the maximum of the weight function, and on the other side to reduce the number of required particles to efficiently sample high dimensional search spaces.

Let us introduce some notation. Let us denote $\mathbf{X} = (x_1, \ldots, x_n, \ldots, x_N)$ the random vector that conforms the state space, where x_n are real-valued random variables. Let us denote $\Psi_l = \{\psi_0, \ldots, \psi_d, \ldots, \psi_D\}$, with $\psi_d \in \{1, \ldots, N\}$, the set of coordinate indices that conform a sub-state space vector of $\mathbf{X}$. We denote such sub-state space vector $\mathbf{X}^{\Psi_i} = (x_{\psi_0}, \ldots, x_{\psi_d}, \ldots, x_{\psi_D})$.

A layered filtering strategy consists on performing several particle filtering runs, each of them focusing on different sub-state space vectors. Each of these runs is called a layer. For each layer l a sub-state space vector is defined by Ψ_l. The algorithm is similar to the annealed particle filtering run, except that in this case the Kernel and weighting functions will be dependent on the sub-state space of the current layer. In fact, the annealed particle filter can be considered a particular case of the layered particle filter.

In this case, we should choose particular transition kernels that consider the sub-state space vector at each layer. For instance, a baseline choice is to define the transition kernel as a multivariate Gaussian distribution with zero mean, and a covariance matrix computed as follows. Let's denote $\mathbf{\Sigma}_0 = diag\{\sigma_0^2, \ldots, \sigma_i^2, \ldots, \sigma_N^2\}$, where σ_i^2 are default variances for each state variable. The covariance matrix $\mathbf{\Sigma}' = diag\{\sigma_1'^2, \ldots, \sigma_i'^2, \ldots, \sigma_N'^2\}$ of the Gaussian distribution for the layer l, $K^l = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}')$ is computed as in Algorithm 3:

Algorithm 3: Compute $\mathbf{\Sigma}'$

Input: current level l, decay rate ν, coordinate indices $(\Psi_m)_{m\in[1,L-1]}$, $\mathbf{\Sigma}_0$

Output: $\mathbf{\Sigma}' = diag\{\sigma_1'^2, \ldots, \sigma_i'^2, \ldots, \sigma_N'^2\}$

```
foreach ψ ∈ [1, N] do
    if ψ ∈ Ψ_l and ψ ∉ Ψ_{l−1} then
        σ'²_ψ = σ²_ψ ;
    else
        foreach m ∈ [1, l − 1] do
            if ψ ∈ Ψ_m then
                σ'²_ψ = (ν^{l−m} σ_ψ)² ;
            else
                σ'²_ψ = 0;
```

Similarly, we restrict resampling to variables that have been already filtered (see Algorithm 4). By doing so, the sample set is able to retain some diversity on dimensions of the state space for which neither dynamics nor observations have been introduced yet in the filtering process.

Algorithm 4: Layered Resampling

Input: current level l, weighted particle set $(\mathbf{x}^{(i)}, \pi^{(i)})_{i=1,\dots,N_s}$, coordinate indices $(\Psi_m)_{m\in[1,l]}$

Output: resampled particle set $(\hat{\mathbf{x}}^{(i)})_{i=1,\dots,N_s}$

$CDF = \{cdf_0, \dots, cdf_{N_S}\} = \mathbf{0}$;
foreach $i \in [1, N_s]$ **do**
    $cdf_i = cdf_{i-1} + \pi^{(i)}$;
$i = 1$;
$u_0 \sim \mathcal{U}_{[0,N_s^{-1}]}$; `// Draw a starting point`
foreach $j \in [1, N_s]$ **do**
    $u_j = u_0 + N_s^{-1}(j-1)$; `// Move along the CDF`
    while $u_j < cdf_i$ **do**
        $i = i + 1$;
    foreach $m \in [1, l]$ **do** `// Assign variables up to layer` l
        foreach $\psi \in \Psi_m$ **do** $\hat{x}_\psi^{(j)} \leftarrow x_\psi^{(i)}$;

Let us denote $\texttt{GPFRun}_{-\texttt{with}-\texttt{layered}-\texttt{resampling}}$, the generic particle filtering run introduced in section 2.2.1 with the particularity that it employs the layered resampling method described in algorithm 4. Then, the Layered Particle Filtering method is summarized by the following algorithm:

Algorithm 5: Layered particle filtering run (`LPFRun`)

Input: initial unweighted particle set $(\hat{\mathbf{x}}_t^{(i)})_{i=1,\dots,N_s}$, $(K_t)_{t\in\mathbb{N}}$, $(w_t^l)_{t\in\mathbb{N},l\in[0,L]}$, $(K_t^l)_{t\in\mathbb{N},l\in[0,L-1]}$

Output: next unweighted particle set $(\hat{\mathbf{x}}_{t+1}^{(i)})_{i=1,\dots,N_s}$

1 **foreach** $l \in [0, L-1]$ **do**

2 $\quad (\hat{\mathbf{x}}_t^{(i)})_{i=1,\dots,N_s} \leftarrow$ `GPFRun-with-layered-resampling`$(\ (\hat{\mathbf{x}}_t^{(i)})_{i=1,\dots,N_s},\ K_t^l(\Psi_l),\ w_t^l)$;

3 $(\hat{\mathbf{x}}_{t+1}^{(i)})_{i=1,\dots,N_s} \leftarrow$ `GPFRun`$(\ (\hat{\mathbf{x}}_t^{(i)})_{i=1,\dots,N_s},\ K_t,\ w_t^L)$;

4 **return** $(\hat{\mathbf{x}}_{t+1}^{(i)})_{i=1,\dots,N_s}$

In the following sections we introduce our proposed methods for motion capture tasks, which use the LPF framework configured according to the requirements of the task. When the LPF layering configuration follows a hierarchy of sub-state space vectors we denote the strategy Hierarchical Particle Filtering HPF.

2.3 Anthropometry estimation using multiple view silhouettes

In this section we introduce a method to estimate the anthropometry of a subject using multiple views. The goal is to adjust a generic human body model to fit the shape of the subject being tracked, and in this way improve the accuracy of motion capture methods that employ the resulting body model for inference.

First, the approach used to build the articulated body model and how it is deformed to adopt distinct poses is explained. Then, we present the skeleton based shape deformation framework. This approach allows to obtain a range of human shapes according to a set of body part scale parameters.

The skeleton based shape deformation framework is used for inference of the anthropometric parameters of the subject body in LPF optimization using weighting functions

based on multiple view silhouettes.

2.3.1 Human Body Modeling

The human body model comprises two components: an articulated skeleton that describes the kinematic properties and a triangular surface mesh that describes the shape.

The body skeleton can be described by the *kinematic tree* concept. A kinematic tree is a set of D reference systems organized in a tree structure, and it represents the connectivity of the joints and bones of the skeleton. A *kinematic chain* is an ordered subset of joints such that all joints are father and son of each other. We call Λ_j the kinematic chain that ends at joint j.

The rigid body motions associated to each joint can be represented by twists ξ_j. The homogeneous matrix $\mathbf{M} \in SE(3)$, which represents the transformation from the *model reference system* to the *joint reference system*, may be constructed from a given twist by computing the exponential map as $\mathbf{M} = \exp(\theta\hat{\xi})$, where $\theta\hat{\xi}$ is the matrix representation of a twist ξ [23].

The rigid body motion associated to a joint can be obtained as the product of the exponential maps along the corresponding kinematic chain,

$$\mathbf{M}_j = \prod_{i=1}^{n_j} \exp(\theta_{\Lambda_j(i)}\hat{\xi}_{\Lambda_j(i)}) \tag{2.11}$$

where n_j is the number of joints involved in the kinematic chain Λ_j and $\Lambda_j(i)$ is a mapping that represents the order in the kinematic chain. The parameters of $\hat{\xi}_j$ are known, as the location of the rotation axes for each joint is part of the model. Thus, the state of the kinematic tree, i.e the pose of the body, is defined by the joint angles state vector and the 6 parameters of the twist ξ_0 associated to the model reference system, $\mathbf{\Theta} := \{(\theta_1, \ldots, \theta_D) \cup (\xi_0)\}$

We model the skin with a 3D triangular mesh whose vertices can move in space according to weights assigned to each vertex. The mesh deformation is achieved with the *linear blend skinning* (LBS) technique [76]. If $\mathbf{v}_i$ is the position of the vertex i, $\mathbf{M}_j$ is the transformation of the bone j, and $\varphi_{i,j}$ is the weight of the bone j for vertex i, the position of the transformed vertex is given according to LBS as

$$\mathbf{v}'_i = \sum_{j=1}^{D} \varphi_{i,j}(\mathbf{M}_j \mathbf{v}_i) \tag{2.12}$$

The skinning weights $\varphi_{i,j}$ are generated with the automatic rigging software Pinocchio [17]. The weights are proportional to the distance from the vertex to the bone, and vary smoothly along the surface. These weights are normalized such that $\sum_{j=1}^{D} \varphi_{i,j} = 1$.

2.3.2 Skeleton based Shape Deformation

In order to obtain different configurations of the shape and pose of the model, the LBS technique for deformation of the skin mesh is applied for three types of transformations $\mathbf{M}$ (see Figure 2.4) related to the bones of the skeleton:

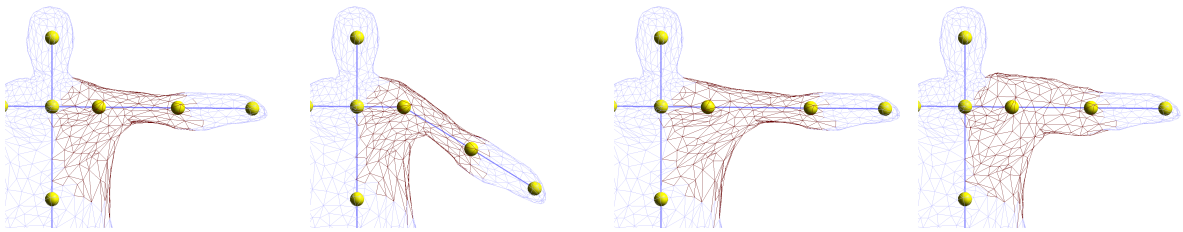
- *Pose deformation*: The mesh is deformed to achieve a specific pose represented by the state vector $\mathbf{\Theta}$, where the bone transformations $\mathbf{M}_j$ are obtained by the product of maps described in equation 2.11. Then the final position for each vertex can be computed using equation 2.12.

- *Scale deformation*: In this case, the mesh is deformed according to the scaling of a bone of the skeleton. Consider a bone j with length S that is scaled such that its final length is $S' = (1 + \gamma_j)S$. Then, the transformation associated to a bone $\mathbf{M}_j$ corresponds to a translation along the direction of the bone, by an amount of translation $S' - S$. Note that scaling a bone implies that the resulting translation

must be applied also to the child joints along the corresponding kinematic chain. Once the translation matrices for each bone are already defined, the new position for the vertices of the mesh can be computed again as in equation 2.12.

- *Deformation in radial direction*: The mesh also can be deformed along the radial direction of the bone. In this case, we compute a translation direction $\mathbf{t}_{i,j}$ for each vertex of the mesh i and bone j, defined as the direction from the vertex position $\mathbf{v}_i$ to the closest point on the considered bone j. For each vertex and bone we obtain $\mathbf{M}_{i,j}$, as the transformation equivalent to the translation $\beta_j \mathbf{t}_{i,j}$. The parameter β_j is the *radial scale* associated to the bone j. In this case, the vertex positions are obtained in the same way than for LBS described in equation 2.12 but, in this case the transformation matrix $\mathbf{M}_{i,j}$ is specific for each vertex and bone.

  This type of deformation is slightly modified to account for radial deformations predominant along a certain direction. This is useful for example for the torso, where we can apply deformations along x or y independently, to describe torso width or depth. To obtain this type of deformation, the translation $\mathbf{t}_{i,j}$ is weighted by its scalar product with the main direction of the deformation.



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

Figure 2.4: Shape deformations associated with left upper arm bone (in red, edges whose vertices have $w_{i,j} > 0.1$) (a) Model at default configuration. (b) Pose deformation. (c) Scale deformation . (d) Deformation in radial direction.

Φ_k	M_k	σ_k^2	Φ_k	M_k	σ_k^2
shoulder height	140.0	17.0	upperarms	25.5	4.0
head	25.0	5.0	lowerarms	30.5	10.0
clavicles	35.0	7.0	legs	90.5	17.0
arms	55.5	10.0	torso	52.0	8.0

Table 2.1: Anthropometric entities size parameters (mean M_k and variance σ_k^2) used to model the probability of acceptance of a particle.

2.3.3 Anthropometry Estimation using LPF

We formulate the problem of estimation of the body shape as an optimization problem where the variables to optimize are the shape parameters and pose parameters. The number of parameters that conform the shape of the body is high, as they consist in the shape deformation parameters γ_j, β_j for each bone, that we described in section 2.3.2. There is also a dependence of these parameters with the pose, in the way that the pose should be refined in order to obtain the optimal bone scale parameters.

The shape variables are grouped to conform meaningful *anthropometric entities* and to respect symmetry. We define the set of anthropometric entities denoted in table 2.1. For example, a meaningful anthropometric entity commonly considered in anthropometric studies is the *shoulders height*. To estimate the shoulders height all bones of the torso and legs are scaled together with the same parameter. In a similar manner, for example, variables are defined for clavicles, arms, legs and head scale. We denote by $\mathbf{\Phi} = \{\Phi_0, \ldots, \Phi_K\}$ the anthropometric entities configuration vector, which consists of mappings to the variables γ_j and β_j. To consider pose and shape parameters in the optimization the state vector is defined as $\mathbf{X} \in \{\mathbf{\Phi} \cup \mathbf{\Theta}\}$, where $\mathbf{\Theta}$ are the pose parameters described in section 2.3.1.

To tackle this estimation problem we combine a local optimization method with a hierarchical layered filtering strategy.

To initialize we perform a sequence of optimization steps using Powell's method [94] to

Ψ_k	Anthropometric entity and Pose	Annealing rate
Ψ_0	Head, Upper Arms, Shoulder Height and $\mathbf{\Theta}$	0.1
$\cdots$	$\cdots$	0.3
$\cdots$	$\cdots$	0.8
$\cdots$	$\cdots$	1.0
Ψ_4	Head and ξ_0	1.0
Ψ_5	Upper Arms anthropometry and pose	1.0
Ψ_6	Lower Arms anthropometry and pose	1.0
Ψ_7	Legs lenght and upper legs pose	1.0
Ψ_8	Lower legs anthropometry and pose	1.0
Ψ_9	Torso, legs and hip width	1.0

Table 2.2: Layering configuration

adjust the pose and global scale of the model, assuming a known initial pose. In this case, we split the variables of each limb in separate optimization steps, which helps avoiding local minima, as proposed in [26].

Next, the the pose and shape parameters $\mathbf{X}$ are optimized by means of hierarchical layered filtering. To set the layered filtering optimization we should define the layering of the search space, the transition kernel functions and the weighting functions.

Layering The partition of the search space for each layer l, denoted by

$$\Psi_l = \{\psi_0, \ldots, \psi_d, \ldots, \psi_D\}$$

(see Section 2.2.4), is designed such that the scaling of the bones does not affect hierarchically preceding scale parameters. To do so the ordering is established following the traversal of the kinematic tree starting from the root. The resulting layering configuration is detailed in Table 2.2. Note that the first 4 layers implement an annealing scheme with its corresponding annealing rate.

Kernel functions The Kernel functions $K_t^l(\Psi_l)$ for each partition or layer should be defined. We propose a set of functions based on the *rejection sampling* concept [19] that

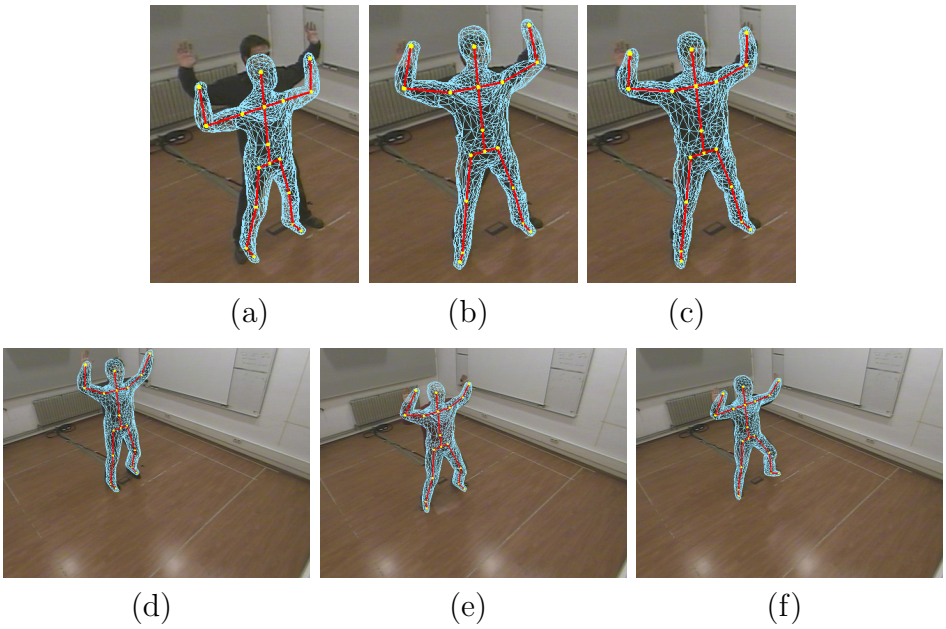


Figure 2.5: Model adjustment. (a) Model set at the initial configuration. (b) Model after global scale and pose estimation using Powell method. (c) Model after shape and pose parameters estimation ($\{\mathbf{\Phi} \cup \mathbf{\Theta}\}$) using LPF. (d,e,f) Model adjustment for distinct subjects.

are configured according to an anthropometric measurements database. For a given layer l, with its corresponding partition Ψ_l, and given the current particle $\mathbf{x}_t^{(i)}$, the function K_t^l should generate a new particle $\mathbf{x}_{t+1}^{(i)}$. The procedure K_t^l to draw new samples is the following:

1. Create a candidate particle $\breve{\mathbf{x}}_t^{(i)}$ by adding Gaussian noise as $\breve{\mathbf{x}}_t^{(i)} = \mathbf{x}_t^{(i)} + \mathbf{N}$.

2. Given $\breve{\mathbf{x}}_t^{(i)}$, calculate the actual size S_k for each anthropometric entity $k \in \{0, \ldots, M-1\}$, and compute the candidate propability as

$$P(\breve{\mathbf{x}}_t^{(i)}) = \prod_{k=0}^{k=M-1} e^{-(\frac{(S_k - M_k)^2}{2\sigma_k^2})} \tag{2.13}$$

3. If $P(\breve{\mathbf{x}}_t^{(i)}) > \tau P(\breve{\mathbf{x}}_t^{(i)})$ the candidate particle is accepted, thus $\mathbf{x}_{t+1}^{(i)} = \breve{\mathbf{x}}_t^{(i)}$. Otherwise, jump again to step 1, to generate another candidate. (For practical reasons this

process is executed for a limited number of trials).

Note that in this way, the size of each anthropometric entity k is modeled with a Gaussian distribution $\mathcal{N}(M_k, \sigma_k^2)$, where its parameters are set according to Table 2.1. These values were obtained from the anthropometric studies presented in the project DinBelg2005 [38]. The acceptance rate τ is set experimentally to 0.7.

Weighting functions the weighting functions w_t are defined as Silhouette XOR cost functions as introduced in section 2.2.1.

2.3.4 Experimental Tests

The method presented has been tested in a smart room environment with 5 cameras. For the model adjustment, the subject is expected to adopt a pose with the hands up, the legs slightly open, looking at a predefined direction. The initial position is computed from the visual hull centroid (generated with the available silhouettes) [75], and the model is scaled to fit the visual hull height. In figure 2.5.a we show the model set at the initial position and scale. Note that, as the initial pose is not exactly the subject pose, placing the model at the visual hull centroid is not accurate. After the optimization of the pose and global scale using Powell's method (Figure 2.5.b) the model is better adjusted, but for example, arms length and head size do not correspond with subject anthropometry. After the estimation of the shape parameters using the LPF (Figure 2.5.c) the model accurately fits the subject. Note that the model does not consider wrist joints and hands, thus those parts are less accurate. Figure 2.5(d,e,f) shows the adjusted model for different subjects.

2.4 Human Body Tracking using Approximate partitioning of observations

This section presents a markerless motion capture system that incorporates a novel class of weighting functions based on a variable bounding box system, that allows to approximately

partition the observations. The method aims at defining regions where pixels are likely to be representing the body part of interest, thus reducing the influence of errors in regions where information is irrelevant. The proposed technique is generic in the sense that it can be applied to any hierarchical particle-based inference method for human pose estimation. The technique is introduced to increase the robustness of a hierarchical LPF method and addresses the problem of estimating the pose of humans in broad multi-view environments using a low number of fixed views. Experiments on the challenging TennisSense dataset [27], using silhouettes as observation model and at most 4 cameras per player show the effectiveness of the proposed method. In addition, we conducted experiments on the HumanEva dataset [111] to show that the approximate partitioning of observations based on silhouettes improves the performance even in a more controlled scenario.

In order to measure the weight of a particle, our algorithm defines an observation model with foreground silhouettes in multiple views, which are obtained by means of a single-Gaussian version of the algorithm proposed by Xu et. al. [128]. In most cases, having multiple views makes silhouettes a useful cue to retrieve the pose. Nonetheless, they also present several problems.

Let us recall that in section 2.2.4, we have presented the sampling strategy of a LPF as a particular case of Partitioned Sampling. It is worth mentioning that Partitioned Sampling requires a set of likelihood functions, one per partition, that must be properly peaked around the same region of the posterior restricted to the partition [83]. In practice, this requirement is difficult to fulfill with observation models based on silhouettes due to several reasons. First, in a wide base-line multi-camera scenario, the appearance of humans in distinct views may change considerably, causing the ratio between the number of pixels representing the human shape and the total number of pixels in an image to change drastically between views. Furthermore, this might be a reason for possible spurious detections of foreground objects to have more impact on the likelihood function (Fig.

2.6a). In the case of human body, silhouettes are rather incomplete as evidence; it is difficult to ensure that a body part is being matched with a set of pixels that constitute the actual representation (or a very good approximation) of that body part (Fig. 2.6b). This phenomenon is especially remarkable when self-occlusions between body parts occur.

In existing HPFs [16], a common solution is to render the body parts that are affected by variables being sampled in a partition or hierarchical level and the preceding ones. Our proposal is to go one step further by defining approximate partitions of the silhouettes in multiple views. The goal is to restrict the cost function to image regions that are likely to represent the body parts being affected by variables in a given hierarchical level. By doing so, we reduce the impact of pixels that are not providing meaningful information about variables in the hierarchical level of interest.

Similarly to the approach for anthropometry estimation presented in section 2.3, the layered filtering optimization comprises a definition of a *layering* of the search space, the *transition kernel functions* and the *weighting function*. In this case, the approach focus in the use of a novel type of weighting functions that exploit the current state and layers.

2.4.1 Weighting function by approximate partitioning of observations

The weighting function that we propose evaluates the cost of a particle on a partition of the observation space. Let us define the partitioning of observations as a set of hidden variables Ω per pixel that take value 1 if the pixel is relevant given a set of state-space variables and 0 otherwise. In human motion capture, we can translate the problem into a graphical model comprising two subgraphs: a first graph $\mathcal{K}(\mathcal{X}, \mathcal{E}_x)$ that encodes the state space variables and a second graph $\mathcal{G}(\mathcal{Z}, \mathcal{E}_z)$ encoding the observed variables (Fig. 2.7).

The edges between nodes from both graphs are determined by the projective transformations and the human mesh model. In existing partitioned and hierarchical sampling approaches, all the pixels from all images are used to define the likelihood functions: this

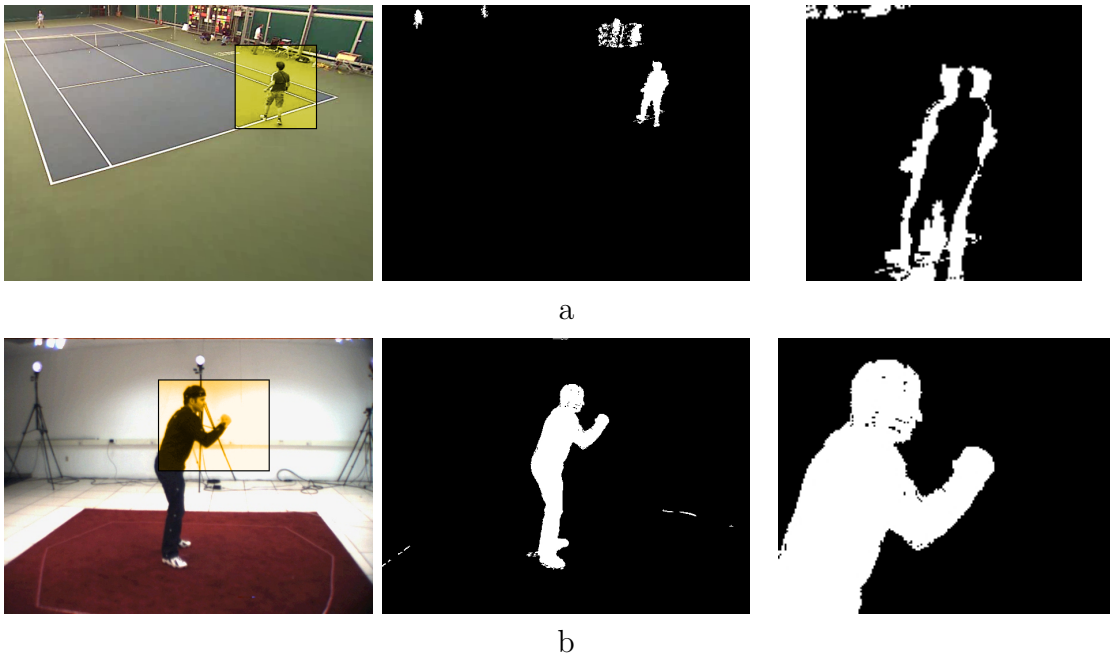


Figure 2.6: Examples where only a small region of the images may be relevant for specific state-space variables. **a**: Only a small region of the image is representing the target player. Overlapping the model projection onto the pixels enclosed by the bounding box yields error masks like the one shown on the rightmost, thus avoiding severe errors due to false positives. **b**: Pixels in the marked region are expected to be relevant for estimating the right arm pose. Note that errors due to leg configurations will not affect if pixels on the rightmost mask are used.

is equivalent to say that always exists an edge between $\mathcal{K}$ and $\mathcal{G}$ for any variable $\mathbf{x}$ in $\mathcal{X}$ and any pixel z in $\mathcal{Z}$. In our proposal, the hidden variables Ω in each hierarchical partition are introduced to block edges between $\mathcal{X}$ and $\mathcal{Z}$ with low influence on the variables of the hierarchical level.

Let us recall the silhouette XOR weighting function introduced in section 2.2.1. The weight of a particle is defined in terms of a pixel-wise XOR between the silhouettes $\mathcal{S}_v$ in each view v and the projection of the mesh modeling the outer shape of the body in the corresponding views, $\mathcal{M}_v$. Then, for the case in which all the observed variables (pixels)

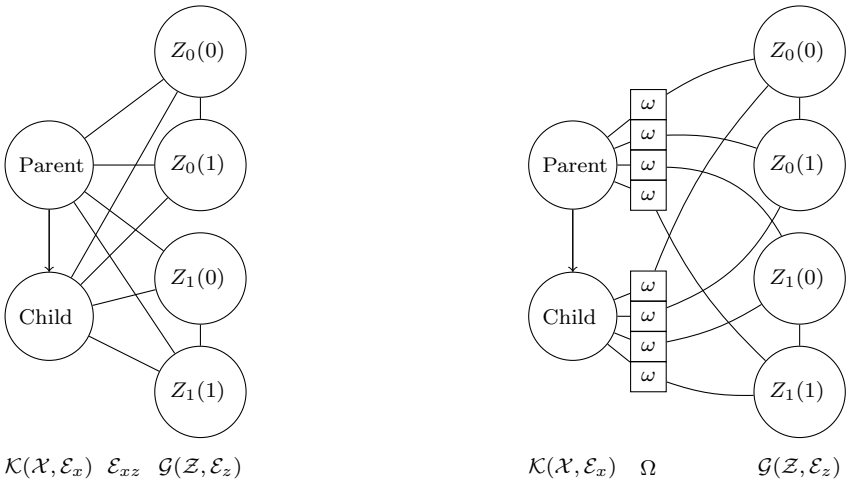


Figure 2.7: Graphical model detailing the influence of the bounding box system in the weighting function. The pose estimation problem can be split into two subgraphs: $\mathcal{K}$ represents the hierarchical relation between human model variables and $\mathcal{G}$ models the pixels in multi-view images (for simplicity, only 2 images, Z_0 and Z_1, and 2 pixels per image are depicted). **Left**: general case where all the pixels have influence over all the human model variables. **Right**: Hidden variables Ω measure how likely is a pixel to belong to a body part. Consequently, low values of Ω decouple pixels from human model variables.

have influence on the cost, we have the following definition for the weight measurement:

$$w_t(\hat{\mathbf{x}}_t^{(i)}) = exp(-\sum_v \frac{1}{\mathcal{C}(\mathcal{S}_v)} XOR(\mathcal{M}_v^i, \mathcal{S}_v)) \tag{2.14}$$

where XOR denotes the number of non-zero pixel values after a pixel-wise XOR and $\mathcal{C}(\mathcal{S}_v)$ is the number of pixels of the silhouette on the v-th view.

Our objective is to define the hidden variables Ω to restrict the pixel domain in equation 2.14 to a set of pixels with an expected relevance for the body part associated to variables in the l-th partition. To estimate the relevancy of a pixel in a given partition we simply use a measure based on spatial criteria. Let us recall that we have a human body model that provides us the necessary information to compute joint positions from state-space variables, and that we have also a set of projective transformations p_v, given by the camera calibration, that map points in $\mathbb{R}^3$ onto each view v. These elements allow us to know the

pixels onto which a joint is projected and thus, how far is a pixel from any projected joint. For a given hierarchical partition l, we have a set of particles $\mathbf{x}_{t,l}^i$ and associated importance weights $\pi_{t,l}^i$. Using this information and the fact that we have already filtered the parent variables (variables for partitions 0 to $l-1$), we compute hidden variables Ω_l as follows:

1. Compute the sample mean $\hat{\mathbf{x}}_{t,l-1} = \sum_{i=1}^{N_s} \pi_{t,l-1}^i \mathbf{x}_{t,l-1}^i$ and obtain the 3D position of the joints in the l-th partition, say $\mathcal{Y}_l$.

2. Compute a 3D Bounding Box centered at the mean joint location $\mu_{\mathcal{Y}_l}$ and with fixed sizes proportional to the estimated anthropometric sizes of parts involved in the l-th partition. Compute a 2D Bounding Box on each image enclosing the projected 3D Bounding Box corners.

3. Set $\omega = 1$ for pixels inside the 2D Bounding Box and $\omega = 0$ otherwise.

Restricting these partitions to have the form of bounding boxes in every view makes the algorithm computationally simple, suitable for GPU processing and allows avoiding explicit 3D reconstructions of the data.

The hidden variables defining the partitioning are computed at the beginning of every layer of the hierarchical particle filter and used in the domain of the likelihood measure:

$$w_t(\hat{\mathbf{x}}_t^{(i)}) = exp(-\sum_v \frac{1}{\mathcal{C}(\Omega_{t,l,v}=1)} XOR(\mathcal{M}_v^i, \mathcal{S}_v, \{\Omega_{t,l,v}=1\})) \tag{2.15}$$

where $\Omega_{t,l,v} = 1$ stands for the set of relevant pixels at time t, for layer l and v-th view.

Note that the proposed partitioning method does not solve the problem of self-occlusions between body parts, but in some sense, alleviates it. Suppose that the arm is being occluded in one view by the torso. One must take into account that, when using silhouettes, views behaving in such a way are not informative and should have a marginal impact on the likelihood measurement. Since the proposed partitioning will often manage to block

errors originated by unrelated body parts, such as legs or the other arm, the final likelihood will be, in general, better defined in the presence of occlusions.

An important consideration of the proposed weight function is the fact that particle weights may reflect the posterior distribution of a partition χ_l rather than the whole state-space. This is due to the local nature of the measurement on the partition domain. To overcome this issue, we add an additional layer at the end of the HPF algorithm, in which we simply re-evaluate the weights of the particles using a bounding box that encloses the whole body model.

2.4.2 Experimental Results

In order to test the proposed algorithm, we conducted experiments on two different datasets: the TennisSense dataset [27] and the HumanEva dataset[111]. While in the case of the tennis dataset we take a clear benefit of the bounding boxes by suppressing errors caused by false positives, the HumanEva experiments show that with good observation models the approximate partitioning of observations yields a better performance of a hierarchical particle-based inference method.

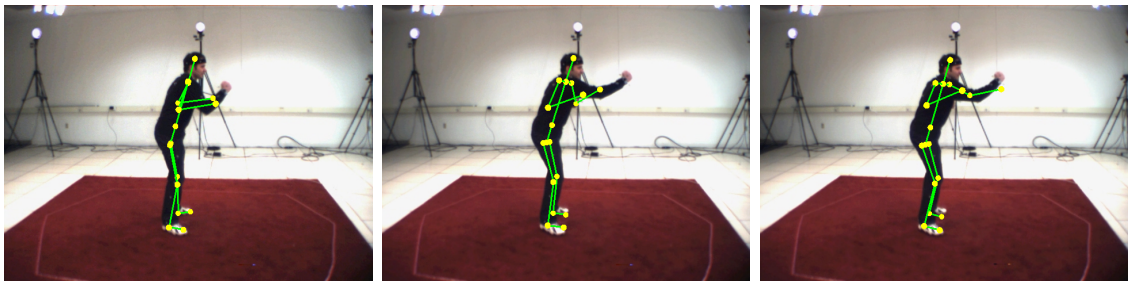
HumanEva DataSet

We evaluate the performance of the proposed approximate partitioning of observations using variable bounding boxes with several HumanEva video sequences. Specifically, we carry the experiments using the HumanEva II Combo sequence performed by Subject 2 and several HumanEva I complete sequences of Subject 3. We test our implementation of the HPF using the Variable Bounding Box Approximate Partitioning of Observations (HPF+VBB-APO) using the weighting function as formulated in equation 2.15. We compare this algorithm with our implementation of the Annealed Particle Filter (APF) [37] and the HPF - using the weighting function in equation 2.14. All the available cameras

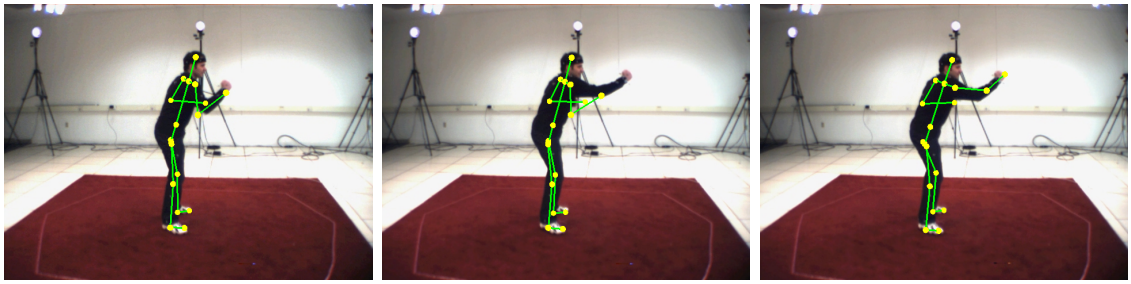
are used in order to have the best observation model (4 in HumanEva II and 7 in HumanEva I). In order to remove the influence of the anthropometric variables on the 3D error, we perform the same anthropometric estimation for all the tests, using the method presented in Section 2.3. In our experiments, the APF uses adaptive diffusion [37] and is configured with 15 layers and 240 particles per layer. The HPF, using the likelihood measurement in equation 2.14, has 7 layers with 572 particles per layer while the variable bounding box HPF has an extra layer - to compute a better estimate of the posterior in the whole state space - and 500 particles per layer. In overall, the three schemes perform almost the same number of evaluations per frame. The computational time for each approach are 21 sec/frame for APF, 21 sec/frame for HPF and 14 sec/frame for the HPF + VBB-APO.

Results in Table 2.3 show that the proposed partitioning of the silhouettes consistently outperforms the common weighting function definition. Although the provided error improvement is about 1 cm in mean, the results in boxing action objectively show that there is a major improvement on the arm pose estimation when using approximate partitioning of observations (Fig. 2.8). In addition, we found that our algorithm enhances the robustness to false positives in the silhouettes. On the contrary, a burst of misses in the pixels of an extremity will have a higher importance when using the variable bounding boxes. However, although the extracted foreground is noisy (due to the use of gray scale images) and there are important misses, they do not suppose a major impact on the final results.

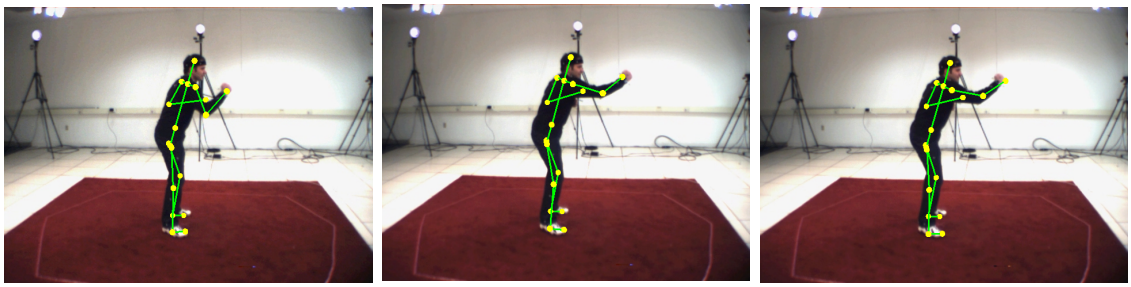
In the light of this observations, and because there is some bias when registering the markers to the ground truth, we argue that standard deviation becomes a highly important indicator of the consistency of the tracking algorithm (whenever the mean 3D error is sufficiently low). The HPF + VBB-APO error standard deviation is lowered to the half with respect to the APF, which denotes better tracking stability.



APF



HPF



HPF+VBB-APO

Figure 2.8: Examples of tracking for the Annealed Particle Filter (APF), the Hierarchical Particle Filter (HPF) and the Hierarchical Particle Filter using Variable Bounding Box - Approximate Partitioning of Observations (HPF+VBB-APO). Using the partitioning yields a better estimation of the arm pose.

TennisSense DataSet

Four of the nine available cameras in the TennisSense dataset [27] are used to track one of the players in the game. We use a set of available ground points to calibrate the extrinsic parameters of each one of the cameras. For each of the views, we pick four background frames to learn the background model.

The available videos of the dataset are timestamped but not synchronized, as the

Action	APF	HPF	HPF + VBB-APO
S2 Combo (34 sec.)	116.36(26.13)	104.86(13.24)	**98.80(9.27)**
S3 Walking (13 sec.)	92.69(15.34)	84.00(6.67)	**76.29(7.00)**
S3 Box (12 sec.)	126.32(56.87)	125.32(55.88)	**100.93(31.27)**
S3 Jog (14 sec.)	109.58(16.94)	97.74(9.68)	**90.87(8.61)**

Table 2.3: Means (and standard deviations) of the 3D tracking errors (in millimeters) for each of the algorithms on the Combo Sequence of Subject 2 in HumanEva II and on several actions of Subject 3 in HumanEva I dataset. Duration of the sequences is indicated in seconds.

Figure 2.9: Selected frames showing pose estimates on different phases of the tennis games. The skeletal model is overlaid on the four views used for the player tracking. For clarity, we only show a wired version of the mesh body model in the second view.

capture was performed at different frame rates. For improved synchronization, we manually locate some distinctive events in each one of the views to determine the shift in the provided timestamps. Then, once the timestamps are shifted to the same reference, we fix a master camera with a frame rate of 22 fps and select frames in other views having the closest timestamp. By visual inspection, we have noticed a remaining synchronization error of $\backsim 2$ frames after applying this procedure.

The tracking is triggered at the beginning of each point (just before the service takes place) and the results are obtained up to the time when the current point is finished.

After the beginning of each sequence, the system waits a fixed number of frames until the anthropometrics estimation is performed.

Since no ground truth is available, we provide a number of snapshots with the obtained results (see Figure 2.9). The system is able to track the pose of the player for a variety of complex movements, such as serves, and several kinds of strokes. In some fast movements, the system is not accurate in the estimation due to the low frame rate and the inaccuracy in the synchronization. Even though, once the fast movement is finished, it is able to recover from the error. The system fails when the player is going outside of the court and the visibility is reduced to less than 3 cameras because the information provided by the silhouettes is too ambiguous. In any case, using the proposed Variable Bounding Box system yields a higher successful number of tracked frames than the APF or the simple HPF.

2.5 Human Body Tracking using Part Detectors

In Layered Particle Filters, the propagation of particles by means of Gaussian diffusion makes the filter blind with respect to evidence. Some research trends focus on tackling such a problem by introducing priors based on trained motion models. This approach, however, has a limited scalability, as one needs to know which motions are performed in the scenario. Hence, it seems more reasonable to resort to image cues related to body parts in order to address the blindness problem. In such a context, partitioned or hierarchical definition of layers emerges as the most efficient approach towards drawing particles, since one can constrain the propagation and filtering to a subset of variables that are related to a particular body part. For that matter, we propose the Detection-Driven Hierarchical Particle filter, a novel Layered Particle filter combining hierarchical layers and body part detectors.

2.5.1 Multi-View Body Parts Detection

We present a method for 3D localization of body parts from multiple views. In this work, we focus on body extremities (hands, head and feet), as they are usually easier to detect and provide sufficient information to estimate the pose variables related to a kinematic chain. Our method takes advantage of 3D information to deal with occlusions and visibility issues, hence we can robustly fuse the outcomes of one or several image-based detectors working in different views. To achieve such a goal, we first obtain a set of points on the surface of the human body. Second, we compute the probability of each surface point to be an extremity, using the detections on multiple views. Then, the surface points are filtered using a threshold and clustering technique in order to obtain the most likely extremity locations.

The choice of the image-based detectors affects not only the performance, but how the probabilities in each surface point should be treated. We demonstrate and exemplify the method using a simple yet effective image-based hand detector. Provided that evaluations are performed in IXMAS dataset [126], where all the subjects have their hands uncovered, we employ the skin detector proposed by Jones and Rehg [65]. Note that this particular choice implies that we cannot distinguish left from right hand and that head is also detected. Hence, an additional classification step is proposed in order to robustly detect 3D hand positions.

Probability Surface

To robustly fuse detections in multiple views, we employ a set of points $\mathbf{q}$ with associated normals $\hat{\mathbf{n}}_q$ lying on the surface of the human body. A suitable set $\mathbf{Q}$ comprising such oriented points can be estimated in a two-step fashion by reconstructing a 3D volume and then computing normals on the volume surface. Alternatively, we opt for the method of Salvador et al. [102] that jointly estimates surface points and normals.

Then, we compute the *visibility factor* $\eta_c(\mathbf{q})$, a value representing the visibility of each oriented point $\mathbf{q}$ with respect to the camera c:

$$\eta_c(\mathbf{q}) = \begin{cases} -\hat{\mathbf{z}}_c \cdot \hat{\mathbf{n}}_\mathbf{q}, & \text{if } \mathbf{q} \text{ is visible} \\ 0, & \text{otherwise} \end{cases} \tag{2.16}$$

We determine if $\mathbf{q}$ is visible from camera c by means of a standard z-buffering test.

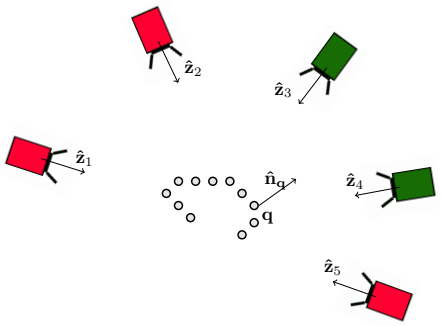


Figure 2.10: Visibility example for a surface point $\mathbf{q}$ (best viewed in color). The best oriented cameras are green whereas the red cameras have few or null visibility

This visibility factor serves to estimate the probability that a surface point $\mathbf{q}$ is representing an extremity:

$$Prob(\mathbf{q}) = \sum_{c=1}^{C} \eta_c(\mathbf{q}) \mathcal{T}(proj_c(\mathbf{q})) \tag{2.17}$$

where C is the total number of cameras, $proj_c(\mathbf{q}) \in \mathbb{R}^2$ are the pixel coordinates resulting from the projection of the surface point $\mathbf{q}$ in camera c, and $\mathcal{T}(proj_c(\mathbf{q}))$ is the probability that the pixel at $proj_c(\mathbf{q})$ is representing an extremity according to an image-based detector. Note that the visibility factor has to be normalized, so $\sum_{c=1}^{C} \eta_c(w) = 1$. We show an example in Figure 2.10.

Due to the visibility factor, the proposed method effectively handles occlusions and inconsistencies of the visual cones computed directly from detections in multiple views.

Moreover, since only the best oriented cameras determine the probabilities of the surface points, our method can infer a 3D extremity location even if it is reliably detected only in a few views. In addition, as the probabilities of surface points are computed from detections inside the silhouettes, all false positives outside them are not taken into account.

Filtering

The filtering step aims at estimating candidate 3D locations of the detected extremities and it is analogous to finding relevant modes of the probability distribution lying on the surface manifold $\mathbf{Q}$. We start by computing the subset $\mathbf{Q}'$ of likely surface points:

$$\mathbf{Q}' = \{ \mathbf{q} \in \mathbf{Q} \mid Prob(\mathbf{q}) \geq \Gamma \} \tag{2.18}$$

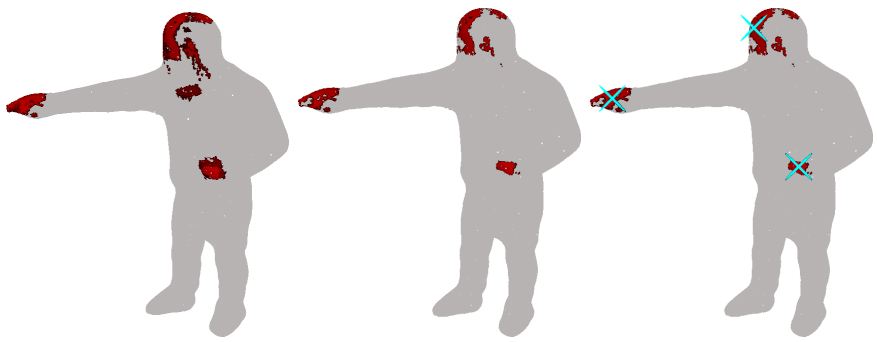
where $0 \leq \Gamma \leq 1$ is a threshold.

Then, we cluster the points in $\mathbf{Q}'$ using an efficient method [100] based on a kd-tree representation [46]. The parameters of the clustering method are distance tolerance, ϑ_{tol}, and the minimum and maximum cluster size, ϑ_{min} and ϑ_{max}, respectively. These parameters are very suitable for our problem, since they can be set by using anthropometric proportions. Finally, cluster centroids are taken as candidates for 3D extremity locations. We illustrate the process in Figure 2.11.

Extremes Classification

In order to determine if any candidate location represents an extremity location, we rely on the pose $\hat{\mathbf{x}}_{t-1}$ estimated on the previous frame. Let us denote $\mathbf{y'}_i = F_i(\hat{\mathbf{x}}_{t-1})$ the position of the end-effector in the i-th kinematic chain at previous time instant, where $F_i(.)$ is the Forward Kinematic operator for the chain i [59].

We formulate the classification problem as an optimal assignment problem. Let $\mathbf{D} \in$



(a) (b) (c)

Figure 2.11: Filtering process (best viewed in color). Probability values are represented with the red channel (a) Probability surface. (b) Probability surface after the threshold. (c) Cluster centroids (cyan crosses)

$\mathbb{R}^{I\times E}$ be the matrix gathering the distances between the I target end-effectors and the E point candidates, and let $\boldsymbol{\Upsilon} = \{\Upsilon_1, \ldots, \Upsilon_I\}$ denote the vector of maximum distance assignments (each Υ_i models both the expected movement and the size of a specific body part). Assignments are noted as an assignment matrix $\mathbf{P} \in \{0,1\}^{E\times I}$ such that each row contains at most a 1 indicating to which end-effector is the candidate assigned. We consider that an assignment is valid if it exists at least one detected point that satisfies a maximum assignment constraint Υ_i. In that case, the problem has a non-trivial solution ($\mathbf{P} \neq \mathbf{0}$) and is formulated as :

$$\begin{aligned} \min_{\mathbf{P}} \ & \frac{\mathrm{tr}(\mathbf{DP})}{\mathbf{1}_E^T \mathbf{P} \mathbf{1}_I} \\ & \text{s.t.} \quad \mathrm{diag}(\mathbf{DP}) \preceq \boldsymbol{\Upsilon} \\ & \text{s.t.} \quad \mathbf{P} \in \{0,1\}^{E\times I} \text{ is a valid assignment matrix} \end{aligned} \tag{2.19}$$

where $\mathrm{tr}(\mathbf{DP})$ denotes the trace of the matrix resulting of the product between distances and assignments, $\mathrm{diag}(\mathbf{DP})$ denotes the vector formed with the diagonal elements of the

same matrix, and $\mathbf{1}_I$ is a vector of ones of length I. The inequality constraint is formulated as a component-wise inequality between two vectors. Hence, we aim at minimizing the overall distance between candidates and end-effectors while maximizing the number of assignments (subject to the maximum distance constraint). In practice, we solve this problem for head and hands by iteratively assigning pairs with minimum distance until a minimum of Eq. 2.19 is attained. We experimentally set the left and right hand maximum distances $\Upsilon_1 = \Upsilon_2 = 35cm$ and the head maximum distance to $\Upsilon_3 = 25cm$.

2.5.2 Detector-Driven Hierarchical Particle Filter

In our method, the role of estimated end-effectors within the layered filtering is two-fold. On the one hand, they are used in a novel kernel function to enhance the particle propagation, thus reducing the blindness of the filter with respect to data. On the other hand, detections are used to enhance the observation model. The weighting function uses the detections in the last layer that we denote as *refinement layer*.

Detector-Driven Propagation

In order to benefit from the localization of end-effectors we define a Layered PF such that the set of variables related to each particular end-effector are filtered in different layers. In this manner, the position of a certain end-effector is used in the Kernel function $K_t^l(\Psi_l)$ of the corresponding layer l. The proposed Kernel function combines two proposals for drawing particles:

1. Detector-driven proposal: pose variables from the appropriate layer are drawn by means of Inverse Kinematics [66] using the detected end-effector locations. Particles drawn from this proposal are termed *IK particles*.

2. Gaussian proposal : particles are generated by Gaussian diffusion in the angle space,

in order to account for the uncertainty of having erroneous detections.

The combination of both proposals should consider the error rate of the extremity detections. Instead of estimating such an error rate offline, we propose an online approximation of the detection accuracy by using the *IK survival rate*, i.e., the estimated ratio of IK particles that will be resampled after weighting function evaluation. Whenever this rate is above 0.5, we mutate a small fraction of regular particles into IK particles. On the contrary, when the rate is lower, IK particles are transformed into regular particles. We constrain the algorithm to keep a minimum of 25% of particles of one kind.

Refinement Layer

The weighting function w_t^l is defined such that the last layer L the pose estimation is refined by introducing the set of end-effector detections, namely $\mathbf{C}_t = \{\mathbf{c}_t^1, .., \mathbf{c}_t^u, ..., \mathbf{c}_t^U\}$, in the weighting computation. We add a new cost function taking into account the distance between the observed detections and the model end-effectors. This cost is computed as:

$$\mathcal{D}(\mathbf{C}_t, \mathbf{x}_{t,l}^n) = \frac{1}{U} \sum_{u=1}^{U} \min_i \parallel \mathbf{c}_t^u - F_i(\mathbf{x}_{t,l}^n) \parallel \tag{2.20}$$

and it is introduced in the weighting function in the following manner:

$$w_t^L(\mathbf{z}_t, \mathbf{x}_{t,l}^n) \propto \mathbf{e}^{-(\mathcal{C}(\mathbf{z}_t, \mathbf{x}_{t,l}^n) + \kappa \mathcal{D}(\mathbf{C}_t, \mathbf{x}_{t,l}^n))} \tag{2.21}$$

where $\mathcal{C}(\mathbf{z}_t, \mathbf{x}_{t,l}^n)$ is a silhouettes-based cost function and κ is a scaling factor. κ is chosen to balance the importance of both cost terms. By using this weighting function, we improve the final pose estimate at each frame, solving mismatches produced by the classification algorithm.

The refinement layer implies a re-weighting of all the particles according to an improved observation model, and does not involve any propagation step, since we do not want to

add noise to particles that have been drawn after layered filtering. If no detections are found at time instant t, the refinement layer in t is not used.

Detection Driven Particle Filtering Evaluation

In order to evaluate our method, we conduct several experiments on the IXMAS dataset [126]. This dataset was recorded for action recognition and hence it does not contain pose ground truth. For this reason, we manually annotate the hands, head and feet of different sequences belonging to 6 different subjects. In particular, we are interested in the evaluation of the performance of our method for actions involving arm movements (i.e crossing arms, hand waving, punching, etc.), so we skip actions not involving relevant arms motion such as walking or turning. Annotations are performed in 1 of every 5 frames.

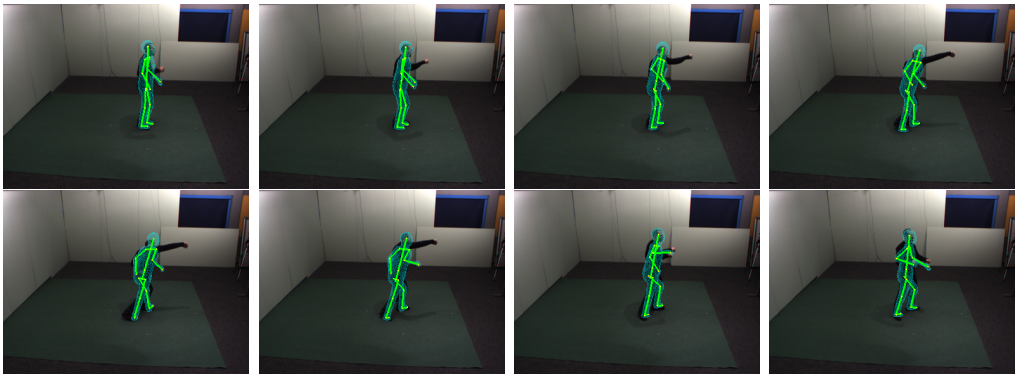
We compare our method with two state-of-the-art *Layered Particle Filters*: the APF and the HPF. In particular, we evaluate our method with (DD HPF$^+$) and without (DD HPF) the refinement layer. In order to perform the comparative, the APF is run with 14 layers and 250 particles per layer, HPF and DD HPF are run using 7 layers and 500 particles per layer, and the DD HPF$^+$ is run with a maximum of 8 layers (7+ refinement layer) and 500 particles per layer (recall that if no detections are found in a frame, the refinement layer is discarded). The 7 layers contain the variables related to torso (and head), left leg, right leg, left shoulder, right shoulder, left elbow and right elbow respectively. Since adding the refinement layer generally implies computing more particle weights, we also provide results for a DD HPF$^+$ using a total of 6 layers (5 + refinement layer) and 500 particles per layer. Using 5 layers implies filtering shoulder and elbow variables of one arm in the same layer.

We compute the mean and standard deviation of the 3D error using all available ground truth data. Results are shown in Table 2.4. The average computational time for each approach are 21 sec/frame for APF, 21 sec/frame for HPF and 25 sec/frame for the DD

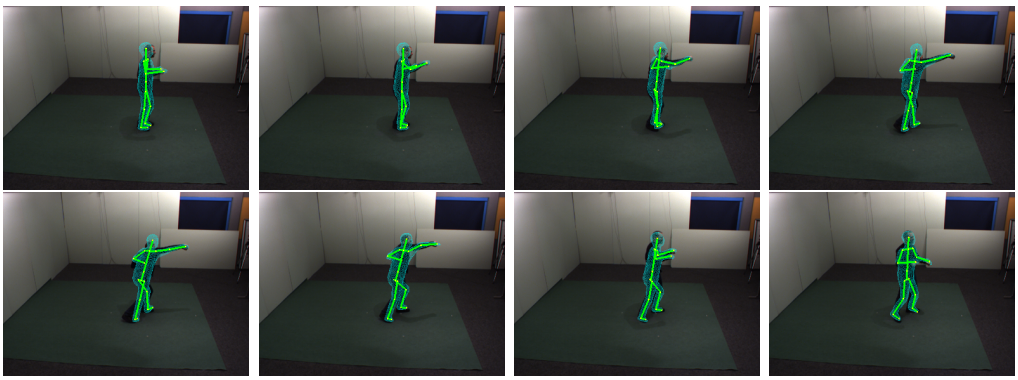
Sequence (Frames)	APF	HPF	DD HPF	DD HPF$^+$	DD HPF$^+$ (5+1)
Alba 1 (53-350) (11 sec.)	23.86 (14.79)	14.38(7.31)	11.68(5.73)	**10.32(6.02)**	11.73(7.37)
Alba 1 (658-1120) (19 sec.)	17.13 (7.93)	11.84(6.87)	10.86(6.03)	**9.37(5.82)**	10.38(7.96)
Chiara 3 (29-292) (11 sec.)	16.05 (7.14)	13.89(8.12)	10.87(6.40)	**9.85(4.43)**	10.08(5.93)
Chiara 3 (576-955) (15 sec.)	19.16 (10.16)	11.29(5.81)	8.36(5.88)	**6.76(3.22)**	7.98 (5.89)
Julien 1 (47-315) (11 sec.)	18.30 (8.37)	15.18(7.21)	13.86(6.85)	13.30(6.85)	**12.09(7.46)**
Julien 1 (596-957) (14 sec.)	26.01 (14.70)	11.85(6.06)	9.43(3.63)	7.87(2.73)	**7.57(3.35)**
Daniel 2 (15-306) (11 sec.)	20.13 (10.52)	16.69(11.41)	14.24(10.09)	11.94(8.49)	**11.74 (7.16)**
Daniel 2 (631-1119) (20 sec.)	16.92 (8.35)	10.22(3.74)	7.29(3.33)	**6.73(2.53)**	11.00 (8.35)
Srikumar 1 (43-368) (13 sec.)	20.06 (10.48)	15.77(9.84)	**14.20(11.22)**	14.58(12.14)	15.59 (16.56)
Srikumar 1 (704-1035) (13 sec.)	18.59 (10.23)	13.60(9.99)	13.46(8.47)	**10.62(5.59)**	12.07 (8.87)
Amel 1 (51-385) (13 sec.)	20.30 (9.32)	16.00(8.02)	16.76(8.47)	**14.40(6.11)**	15.08 (5.88)
Amel 1 (796-1295) (19 sec.)	22.97 (8.05)	13.07(7.07)	11.55(5.79)	11.40(6.90)	**11.35 (5.99)**

Table 2.4: Comparative results between the state-of-the-art methods and our proposals. We provide mean 3D error (and standard deviation) in centimeters. Bold figures highlight the result of the best method for each sequence.

HPF.



HPF



DD HPF

Figure 2.12: Tracking example of subject 3 "punching"action (set 2) for the HPF and DD HPF using 7 layers and 500 particles per layer for both schemes.

The proposed framework consistently outperforms both the APF and HPF. Apart from reporting a better accuracy in terms of mean error, the standard deviation is consistently reduced, thus reflecting the stability gain and the increased robustness in front of tracking failures. Experimental results show the effectiveness of the proposed DD HPF in reducing the blindness of the prior. In Figure 2.12, we provide a visual example for a fast arm action (action 6) and we compare the outcomes of the best state-of-the-art PF, the HPF, with the DD HPF. As we can see, the HPF gets lost whereas the DD HPF perfectly tracks the arm. The experimental validation also shows the impact of the refinement layer, which is able to filter erroneous particles originated by weaknesses of the silhouette-based observation model and erroneous classifications of extremities. Even when using less particle evaluations, the proposed method outperforms the state-of-the-art approaches.

2.6 Conclusions

In this chapter we have presented a stochastic optimization framework applied to motion capture tasks. We have demonstrated its practical use for estimation of the anthropometric proportions of a subject and for tracking of body pose.

We have demonstrated with experimental results that the hierarchically layered particle filtering strategy outperforms both local optimization and annealed particle filtering in HumanEva and IXMAS datasets.

Moreover, we have studied the weighting functions and transition kernel functions and we have proposed two distinct approaches that improve the baseline approach.

On the one side, the APO approach presented in section 2.4 defines a new kind of weighting functions. These functions evaluate samples taking into account only the region of the observation space that is relevant to the subspace currently filtered. We demonstrated that APO outperforms the standard Silhouette XOR weighting function in Human Eva dataset. Specifically it improves the accuracy when tracking arms under self-occlusion

for certain views, which is the typical case where weighting functions that evaluate through the whole image cause failures. Also, the APO is effective in scenarios such as the one presented in the Tennis Sense Dataset. In this case, the subject being tracked is far from the cameras, thus its apparent size is small. The APO method is specially suited for such scenarios, as the particle evaluation dynamically adapts to viewpoint particularities exploiting actual tracking data.

The DD-HPF approach overcomes the blindness to data in the propagation of particles by proposing new transition kernels that take in to account end-effector positions. We demonstrated the effectiveness of the proposed solution in the IXMAS dataset, which is a dataset that comprises challenging arm gestures and whole body actions. By using an end-effector detector we obtain arms and head positions that can are used in a kernel transition function that favors particles with low distance to the end-effector. This technique in combination with the Silhouette XOR weighting function improves significantly the accuracy with respect to the use of a Gaussian kernel transition function. While the Layered Particle Filtering strategy is essentially a generative approach, with this solution we end up with an hybrid approach that takes advantage of a discriminative method for end-effector detection and fuses this detections to obtain the whole body pose.

CHAPTER 3

Interactive Machine Learning for Gesture Localization

Human motion analysis from visual data has a wide range of applications in the human computer interaction field. Among its applications, gesture detection or recognition is a main research area probably because gestures are particularly suited to use as inputs for an interactive system for several reasons. First, they are commonly used in natural conversation an non-verbal communication, thus, in general people feel comfortable when gesturing. Moreover, gestures form a vocabulary or language, so they can be used to convey specific information directly interpretable by the computer, for example, to activate distinct events in an interactive application. Also this kind of interaction would be naturally understood and easily remembered by the user, as again, it is our form of non-verbal communication and we feel familiar with its usage. For this reason, interfaces based on gestures can be more simple than other kinds of interaction where the user feels less comfortable and require a longer learning curve.

Recently, gesture recognition methods have become part of commercial systems such

as smart tv [103] or gaming applications [84]. One of the key elements towards their deployment in real systems is the irruption of low-cost or consumer depth cameras, which overcome critical illumination and depth perception problems of classical vision approaches.

As introduced by Wachs et. al. [124], the design of gesture-based interfaces must address two major issues: *intuitiveness* and *gesture spotting*.

By *intuitiveness*, one means that the types of gesture selected should have a clear cognitive association with the functions they perform. However, it should be taken into account that a gesture natural to one user may be unnatural to others, due to the strong associations with cultural background and experience. *Gesture spotting* [124] consists of distinguishing useful gestures from unintentional movement. This problem may be afforded by the recognition technique, by performing a temporal segmentation to determine where the gesture start and ends. However, this is a difficult problem and often the recognition methods assume temporally segmented actions [34]. Also the recognition may require spatial segmentation of the body parts (e.g hands) which may be also a task prone to errors.

To overcome gesture spotting difficulties, López-Méndez et. al [82] propose an interface based on still gestures. By relying on still gestures, the problem is then formulated as an object detection problem, thus the method learns the local appearance of gestures and neither temporal or spatial segmentation are required. The gesture localization method uses range data and it is based on class-specific (one-vs-all) random forests.

In this thesis, we extend the approach proposed in [82] to improve its performance in cluttered scenes by introducing the *Depth Clipping Test*. We explain this method in section 3.2 and several experiments and evaluation are described in section 3.2.6.

In this gesture localization application, the main difficulty is that the training data is highly unbalanced, the training set is composed by few positive examples and a huge amount of negative examples. While the approach from [82] relies on a boosted learning

technique to overcome this issue, and automatically choose the best training samples, the performance of the method is still highly influenced by the set of negative examples available in the training set. In case that the detector tests fail for certain samples, an experimental solution to this problem is to introduce more data based on the test failures, and train again the detector. This process can be tedious, as training can be slow, and once it finishes one would require to manually test and record new data, and then train again and so, in an iterative manner. Also the training set will grow and, as a consequence, the training process will be slower and the memory requirements will also grow. We propose an alternative using online learning, such that in this case the selection of the apropiate training data can be done during the training phase, according to the prediction provided by the random trees on-the-fly. In this case, recording data, annotation and training is fused in a single interactive application which allows to reduce the efforts required in its off-line counterpart. The interactive machine learning approach to gesture localization allows to reduce dramatically the time dedicated to recording, annotation and training. It is described in sections 3.3 and 3.4. Part of the work in section 3.2 has been published in [6]. The work in sections 3.3 and 3.4 has not been published.

3.1 Related Work

The recent commercialization of new game console controllers as Kinect or Wiimote, has been rapidly followed by the release of proprietary or third part drivers and SDKs suitable for implementing new forms of 3D user interfaces based on gestures [45]. On the one side, several authors propose gesture control interfaces based on accelerometers as the Wii controller [105][77]. On the other side, the Kinect depth sensor allows for device-less interfaces. Some solutions as ZigFu [131] or GesturePak [53] use the skeleton tracking SDKs [70][88] as input for gesture recognition based on skeletal poses. Thus, such approaches require a human pose estimation step, which is a complex task with high computational cost, often

prone to errors in presence of clutter. Alternatively, several approaches make use of raw Kinect depth data as input for hand pose and gesture recognition methods [97][121][95]. In contrast to 2D color video, the use of depth data makes such methods robust to illumination changes and suitable for dark environments. Moreover, depth data provides 3D information valuable to account for scale invariance of human body parts. In general, these features make depth based methods perform better than its color-based counter part.

3.1.1 Random Forests for Object Detection

Regarding object detection and localization methods, random forests and their variants have attracted the attention of the image processing and computer vision community [24, 109, 47] due to its excellent performance for classification and regression tasks. Demirdjian and Chenna [34] proposed a temporal extension of random forests for the task of gesture recognition and audio-visual speech recognition. However, as before, their gesture recognition approach strongly relies on spatial and temporal segmentation. In fact, this approach is not shown to work properly in real applications, since experiments are performed on temporally and spatially segmented sequences. Shotton et al. [108] use random forests on range data to detect body parts. Their ultimate goal is to estimate human poses, which at the same time may pave the way towards pose-based gesture recognition. However, their approach relies on a dense labeling of the human body in order to detect each part. Gall et al. [47] propose a Hough forest framework for dealing with different tasks, including object detection and localization. Although they overcome the problem of the dense labeling, their Hough forest framework is tested on standard datasets where objects of interest (*positive*) have a relatively large size compared to background (*negative*).

3.1.2 On-line Learning

For some applications, on-line learning has several advantages. Usually, off-line training methods require that the entire training data is given in advance, but in some cases this is not possible. For example in tracking applications, data arrives sequentially and predictions should be given as data is acquired, while the entire training set is still not available. Also on-line learning is also interesting in case that training data cannot be stored, for instance if a huge amount of data is required or data should be discarded just after the on-line data generation process. In general, on-line training is faster and the memory requirements are much lower.

Saffari et. al. [101] proposed an on-line learning algorithm for random forest and showed usage examples for tracking and for interactive image segmentation. Random forests on-line learning has been further studied by Denil et. al [35] where they introduce a theoretically consistent algorithm and it is evaluated for classification tasks in comparison with its off-line counterpart. An online algorithm for Hough forests has been introduced by Schulter et. al. [106] and they show how it outperforms off-line Hough Forests in detection tasks.

Yao et. al. [129] use On-line Hough Forests to help the annotation task in an interactive system. In such approach, the object detector is trained with recently annotated examples, and then the detector is used on-the-fly to propose the annotation to the user, thus reducing the annotation cost.

3.1.3 Interactive Machine Learning

Interactive Machine Learning (IML) is a research field that emerges at the intersection of machine learning and human-computer interaction disciplines. Although machine learning systems usually deal with human inputs such as labels, demonstrations or feedback, the

classical machine learning model does not consider usability issues and the user capabilities. In the classical setting, to train a classifier requires an expert user and long training time. This implies that the model is hard to use for example by developers that want to include machine learning systems in new machine interfaces. Early research has focused IML towards the development of machine learning systems which include human interaction in the training loop by means of graphical interfaces that show visual information about parameters, performance or system predictions [11][41][125]. Fails et. al. in [41] introduce an IML model for a visual image classifier generator to be used by designers of perceptual user interfaces that may not have expertise in machine learning. In this model the designer corrects and teaches the classifier by rapidly generating training data (manually classifying pixels), then examining the feedback received, to continue generating more training data and so on in a cycle, until the final classifier is created. IML systems can also be used by domain experts to get deeper insights of the method and improve the accuracy of the classifiers. Examples of interfaces intended for expert users, that can also facilitate the task to non experts, have been presented by Ware et. al [125] and Ankerst et. el. [11], where in this case, the user can construct decision trees graphically. Both systems present data visualization, trees visualization and allow to tune parameters and report results. The following table compares the main characteristics of the classical and the interactive machine learning models:

Classical ML	**Interactive ML**
One-pass process	Iterative process
No users feedback	Users control the behavior
Long time to train the model	Latency sensitive for training
Numerical evaluation	Friendly visualization evaluation

IML has been employed for diverse applications such as tools for image retrieval (CueFlik [44]), helping community content creation using information extraction [60], group creation in social networks [9] or visualization of confusion matrices to understand various classifiers (EnsembleMatrix [117]).

IML has also evolved in robotics research, where interactive systems are used to teach robots [31] [118], where the robots apart from learning on their own, they also learn interactively from people who is unfamiliar with machine learning. Several robotics IML applications are based on the *reinforcement learning* model [119] [72].

Another research field that has strong links with IML is *active learning* [107]. In active learning, the learning algorithm is able to interactively query the user or other information source to obtain the desired outputs at new data points. Active learning principles have been used throughout the whole range of machine learning applications [107]. Image retrieval is a research field that relies strongly on active learning concepts. For example, Tong and Chang [120] combine active learning with support vector machines in order to implement a relevance feedback algorithm, i.e to interactively determine a users desired output or query concept by asking the user whether certain proposed images are relevant or not. These kind of techniques are effective with image databases, where it is difficult to specify queries directly and explicitly.

3.2 Random Forests for Gesture Detection

In this section we first describe the method for gesture localization proposed in [82], that we will use as baseline method for further comparisons and evaluations. Then the depth clipped binary test that we propose to reduce potential clutter is introduced in section 3.2.5. Experimental results and comparisons with baseline method are presented in section 3.2.6.

We first review the main notions on random forests, for the sake of completeness.

3.2.1 Random Forests

Random forests [24] are an ensemble of m randomized trees, which are binary trees. Nodes n in each tree have a learned probability distribution $p_n(c|\mathbf{I}_t, \mathbf{x})$ that reflects how likely is a class c given a pixel $\mathbf{x}$ in the image $\mathbf{I}_t$. These probability distributions are learned by recursively branching left or right down the tree, according to some node-specific weak classifier, until some stopping criteria are met and thus a leaf node is reached. Weak classifiers associated to each node are binary functions of feature vectors obtained from images $\mathcal{I}$. The robustness of forests is based on the combination of several classification trees. Usually, one performs this combination by averaging the distributions over the leaf nodes $\{l_1, \cdots, l_M\}$ reached in all the M trees:

$$p(c|\mathbf{I}_t, \mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} p_{l_m}(c|\mathbf{I}_t, \mathbf{x}) \tag{3.1}$$

Each tree is trained separately with a small subset of the training data obtained by sampling with replacement. Learning is based on the recursive splitting of training data into left $\mathcal{L}$ and right $\mathcal{R}$ subsets, according to some binary test f and a threshold θ. The binary test is a function of the feature vector $\mathbf{v}$ obtained from each training example.

At each node, a test f and a threshold θ are randomly generated, and the one that maximizes some criteria is selected. We employ the information gain as a test selection criterion:

$$\Delta IG = -\frac{|\mathcal{L}|}{|\mathcal{L}| + |\mathcal{R}|} H(\mathcal{L}) - \frac{|\mathcal{R}|}{|\mathcal{L}| + |\mathcal{R}|} H(\mathcal{R}) \tag{3.2}$$

where $|\ |$ denotes the number of elements of the subset and $H()$ is the Shannon entropy of the classes in a subset. The process continues until a maximum depth D is reached or the information gain cannot be further maximized.

3.2.2 Depth Binary Tests

The binary tests used in [82] are based on the binary tests initially proposed in [108]. Specifically, for a given pixel $\mathbf{x}$ the test f has the following expression:

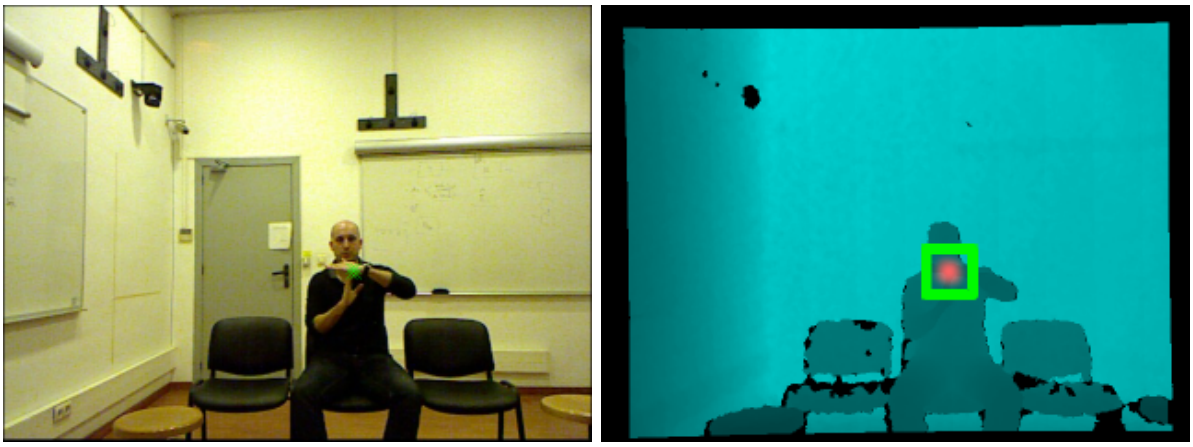
$$f_\theta(\mathbf{I}, \mathbf{x}) = d_\mathbf{I}\left(\mathbf{x} + \frac{\mathbf{u}}{d_\mathbf{I}(\mathbf{x})}\right) - d_\mathbf{I}\left(\mathbf{x} + \frac{\mathbf{v}}{d_\mathbf{I}(\mathbf{x})}\right) \tag{3.3}$$

where $d_\mathbf{I}$ is the depth map associated to image $\mathbf{I}$ and $\mathbf{u}$ and $\mathbf{v}$ are two randomly generated pixel displacements that fall within a patch size. Pixel displacements are normalized with the depth evaluated at pixel $\mathbf{x}$ in order to make the test features invariant to depth changes.

3.2.3 Boosted Learning of Random Forests

In a gesture localization problem, positive and negative classes are naturally unbalanced. On the one hand, in real applications users are not constantly performing gestures. On the other hand, the actual appearance of a gesture may be represented by a relatively low number of pixels. Summing up, the distribution of gesture classes (*positive*) with respect to non-gesture (*negative*) is biased towards the latter. This unbalance makes that low false positive rates constitute actually a large number of false positive votes. Taking into account this phenomenon is important during the training phase of a random forest since, under such unbalance, it will be difficult to optimize the information gain.

To address this problem, Shotton et. al. [109] introduce a weighted random forest scheme based on weighting the *positive* examples with the inverse class frequency. Alternatively, López-Méndez et. al. [82] propose to constrain the training data sampling such that each class obtains approximately the same number of training samples, i.e., the distribution at the root node is approximately uniform. In the gesture localization problem, balancing the number of training samples presents better performance than weighting since the unbalance is so large that weighted random forest overpowers the response on positive



(a) (b)

Figure 3.1: Gesture Localization with Random Forests (best viewed in color). (a) A number of votes (green dots) are casted for the target gesture (b) votes are aggregated to estimate a probability density (overlaid in red on the input depth map) and a localization is estimated (green square).

classes, thus increasing the false positive rate.

While balancing reduces the bias, one typically ends up sampling highly correlated *positive* samples while loosely sampling the *negative* class. This drawback has a greater impact on class-specific (or one-vs-all) learning schemes, where loose sampling has the effect of missing relevant samples of the *negative* class that may effectively improve the localization accuracy. In order to overcome this problem, a boosted learning scheme is introduced.

The method performs training of random forests as follows. The first tree is trained with a balanced set of samples from each class. Once the tree is trained, it is evaluated against the out-of-bag set [24]. The wrongly classified samples are added to the training set of the second tree (up to a maximum number of training samples). This new training set is completed by sampling with replacement from the full training set until balance is achieved. The second tree is then trained with this training subset and the process is repeated until the forest is fully trained.

In the proposed approach, there is no attempt to balance the wrongly classified samples. This is because the majority of wrongly classified samples are false positives that one want to incorporate into training subsets, in order to have a more relevant set of *negative* examples.

A boosted learning scheme for random forests has been presented also in [47]: in [47] base learners are *subforests*, while in the approach in [82] presented above base learners are decision trees. Another difference with respect to [47] is that Lopez-Mendez et. al. [82] exclusively use the out-of-bag set. This increases the efficiency (we evaluate less samples) and, together with the tree-wise approach, produces less correlated trees, which helps in reducing the generalization error [24].

Patch collection For each annotated image a set of patches are collected for training. Collecting patches for all the pixels in the image would require greater run-time memory in the training process and many of them would be redundant. For this reason, only a small subset of patches are sampled from the annotated image. In a positive example, a fixed number of positive samples N_+ are obtained by randomly sampling within a bounding box centered in the annotated point. Negative samples are obtained randomly choosing pixels from outside such bounding box. In a negative example, N_- sample patches are obtained randomly from the whole image.

3.2.4 Gesture localization

For gesture detection and localization, a set of patches are provided to the detection forest, which casts a vote whenever a *positive* class has more probability than the *negative* class and other *positive* classes. Figure 3.1 illustrates the casted votes for a positive class in a class-specific learning example. To detect a gesture, we first estimate a probability density using the votes within a frame and we take into account temporal consistency by

recursively updating this distribution with votes aggregated from past time instants. In order to construct the probability density, we use a Parzen estimator with Gaussian kernel. In order to account for the time component of the approximated density, we sequentially update such density $p(c|\mathbf{I}_t)$ as follows:

$$p'(c|\mathbf{I}_t) = \alpha p(c|\mathbf{I}_t) + (1-\alpha)p'(c|\mathbf{I}_{t-1}) \tag{3.4}$$

This is a simple yet effective method to keep temporal consistency of the casted votes, as it requires storing a single probability map. An adaptation rate $\alpha = 0.8$ works well in practice, as it prevents several false positives while avoiding a delayed response.

Finally, we compute the pixel location $\mathbf{g}_c$ of a gesture class $c > 0$ as the pixel location with maximum probability. We ensure that such a maximum represents a target gesture by thresholding the probability volume V computed by locally integrating the estimated pseudo-probability measure :

$$V = \sum_{\mathbf{x} \in \mathcal{S}} p'(c|\mathbf{I}_t(\mathbf{x})) \tag{3.5}$$

where $\mathcal{S}$ is a circular surface element of radius inversely proportional to the depth, and centered at the global maximum. In this way, the localization is depth-invariant.

3.2.5 Depth Clipped Binary Tests

Although the proposed boosted learning yields an improved sampling of the *negative* class, modeling the appearance of clutter, distractors or accidental gestures from a limited set of training data is an ill-posed problem. Typically, one may resort to spatial segmentation [55]. However, segmentation on depth data usually assumes a certain distance with respect to several background objects, as well as prominent motion to accurately obtain the silhouette of humans in a scenario. Furthermore, the segmentation step can be computationally costly.

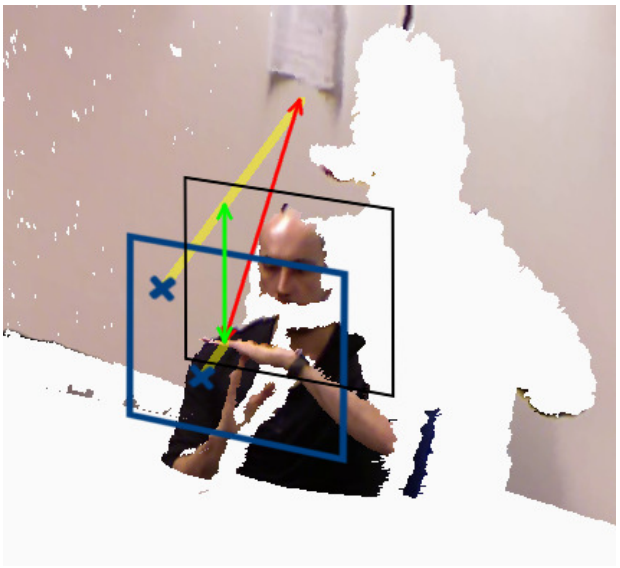


Figure 3.2: Three-dimensional representation of binary tests (best viewed in color). Without clipping, the algorithm is prone to learning binary decision tests that consider the actual depth of the training data (red arrow). Clipping avoids this problem by producing an artificial depth value that can be reproduced in test time (green arrow).

In this work, we advocate the avoidance of a segmentation step. Instead, we introduce an auxiliary parameter that clips the depth of the available training examples.

If we represent the binary tests (Equation 3.3) in the 3D space (see Fig. 3.2), we see that displacements may point to *background* pixels. This implies that the binary tests used during training may yield forests that *detect gestures with the backgrounds observed in the training data*. To avoid this, the proposed auxiliary clipping parameter defines a maximum displacement in the Z axis, i.e., depth values. Specifically, the clipping parameter is a value that represents the maximum and minimum relative depth with respect to the depth value of the center pixel. Formally, let κ denote the clipping parameter. Then, the binary tests are re-formulated as follows:

$$
\begin{aligned}
&f_\theta(\mathbf{I}, \mathbf{x}) = \\
&\max\left(\min\left(d_\mathbf{I}\left(\mathbf{x} + \frac{\mathbf{u}}{d_\mathbf{I}(\mathbf{x})}\right), d_\mathbf{I}(\mathbf{x}) + \kappa\right), d_\mathbf{I}(\mathbf{x}) - \kappa\right) \\
&-\max\left(\min\left(d_\mathbf{I}\left(\mathbf{x} + \frac{\mathbf{v}}{d_\mathbf{I}(\mathbf{x})}\right), d_\mathbf{I}(\mathbf{x}) + \kappa\right), d_\mathbf{I}(\mathbf{x}) - \kappa\right)
\end{aligned}
\tag{3.6}
$$

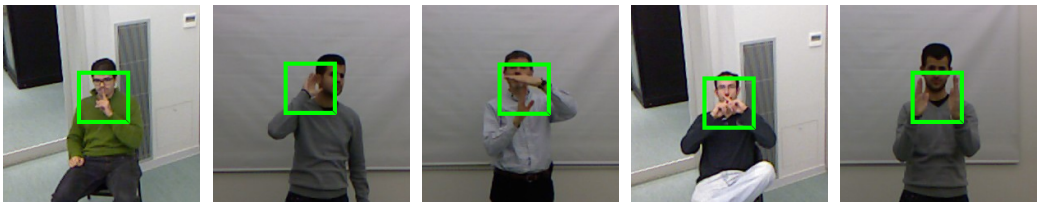


Figure 3.3: Examples of successful localization results using our approach ($\kappa = 15$ cm) *Silence*, *Audio*, *Tee*, *Mute* and *Pause*.

Table 3.1: Average Area Under the Curve (AUC) for different learning approaches.

Method	Silence	Audio	Tee	Mute	Pause	Average
Boosted [47]	0.17	0.10	0.31	0.14	0.02	0.15
Method in [82]	0.17	0.10	0.31	0.20	**0.25**	0.21
Ours + $\kappa = 15$cm	0.58	**0.53**	**0.81**	**0.47**	0.11	**0.50**
Ours + $\kappa = 30$cm	**0.69**	0.50	0.54	0.27	0.14	0.42
Ours + $\kappa = 50$cm	0.59	0.46	0.61	0.24	0.18	0.41

Similarly to the depth invariant displacements, the auxiliary clipping parameter renders tests that are more robust to changes in the background. This can be also regarded as an improved characterization of training examples by its local appearance, where local is understood in the 3-dimensional world, and not only in the image domain.

3.2.6 Experimental results

We conduct experiments of the gesture localization method to provide a quantitative performance of the approach and determine optimal clipping parameters. We focus on the gestures defined in Figure 3.3. We record 5 training sequences where 5 different actors perform a set of gestures and actions, including the 5 target gesture classes. Additionally, we record 6 test sequences with 4 additional actors that were not included in the training set.

The employed detection forests have 15 trees with maximum depth 20, and each tree is trained with approximately 20000 examples per class.

In all cases, we employ squared patches of size 85x85 pixels. We train class-specific forests with the boosted learning method. Additionally, in order to compare the learning method, we implement a boosted approach based on [47].

To measure the accuracy of the proposed methods, we consider a correct localization if the estimated gesture and the actual gesture belong to the same class and if the estimated location is within a radius of 10 pixels. We compute the curves representing 1-precision vs recall to then compute the Area Under the Curve (AUC). Average AUC's per gesture class are shown in Table 3.1.

The comparison of different training approaches shows that the proposed clipping depth test improves the results with respect to [82]. These improvements are specially significant for the gesture classes that can be potentially more affected by background clutter, i.e the gestures that are not performed in front of the chest. For example, for *silence* and *audio* gestures. In *pause* gesture we do not appreciate improvement. This is due to the computation of the reference depth for such gesture. As the reference depth is obtained from the center of the patch, in this gesture such point results not precise enough to perform proper clipping.

3.3 Online Random Forests for Gesture Localization

Still gesture localization using a random forest classifier based on range data has good accuracy and is computationally efficient, which enables the technology to be used for human computer interaction applications. Even though, a cumbersome aspect of the method, as in general happens in supervised learning methods, is that it requires a relatively large amount of annotated examples. Moreover, in order to keep low false positive rates, a set of negative examples should be carefully chosen for each specific gesture in order to keep the method robust in real life scenarios. In practice, this means more recording and annotation which is a slow process that requires a lot of human effort. Also, it is not clear in advance

which set of negative examples would be relevant in order to improve the classifier accuracy, so in general, one would train the classifier with an initial dataset, then test it, and realize that it fails with certain examples. So more instances of those examples would be recorded and annotated and included in the dataset. In this thesis, we advocate for the use of an interactive machine learning approach to ease the recording, annotation and training procedure of the gesture classifier. So in a single process the user would record examples, the classifier would be trained, and then by testing on-the-fly, relevant negative examples would be added to the training set. In order to automatically annotate the examples, the scenario will be subject to certain constraints to facilitate the task. This process requires a classifier suited for online learning or incremental learning. As random forests has demonstrated accurate results for the gesture localization task with rely on online random forest learning. We base our method on the algorithm proposed by Saffari et. al. [101].

3.3.1 Online Random Forests

Random forests online learning has relevant differences with its offline counterpart. On the one side, bagging should be performed differently. In the offline case, given a standard training set D of size n, bagging generates m new training sets, each of size $n' < n$, by sampling from D uniformly and with replacement. Trees are trained with these sets, in order to improve accuracy and to help avoiding over-fitting. In the online case, the whole training set is not available. For such case, Oza et. al. [2] proposed a method that is proved to converge to off-line bagging. Following such method, each tree is updated on each sample k times in a row where k is a random number generated by a Poisson distribution.

On the other side, the tests assigned at each node of the tree also could not be computed in the same manner that for the offline case. In the offline case, initially a set of random tests is generated for each node. During training, the best tests are selected such that they maximize the information gain. As the whole training dataset is available, such statistics

can be robustly estimated.

In the online mode, trees are grown upon arrival of new data. A tree starts with only one root node with a set of randomly selected tests. A node is split, so the best test according to the information gain is chosen, when two conditions apply: 1) a minimum number of samples α has been already seen by the node, 2) the split achieves a minimum information gain β. Such process is applied to the right and left newly generated leaf nodes, and so until the tree has grown to the required depth. At each split, the statistics for each class label of the parent node are propagated to children such that leaf nodes can perform classification on-the-fly, even before observing new samples. In Algorithm 6 we present the pseudo-code of the forest updating method which is executed for each sample x with label y.

Algorithm 6: Online RF proposed by Saffari et. al.[101]

```
1  for t from 1 to T do
2      k ← Poisson(λ);
3      if k > 0 then
4          for u from 1 to k do
5              j =findLeaf(x);
6              updateNode(j, {x, y});
7              if shouldSplit(j) then
8                  findBestTest(j);
9                  createLeftChild();
10                 createRightChild();
```

In the following the functions in the algorithm are further explained:

- **findLeaf(x)** recursively traverse the tree starting at root node until a leaf is found,

it applies best selected test at each node.

- **updateNode**$(j, \{x, y\})$ updates the following statistics at node j. Statistics of class labels $\mathbf{p}_j = [p_1^j, \ldots, p_K^j]$ where p_i^j is the label density of class i and K is the number of classes. For each random test compute also the statistics of class labels of samples falling at left or right partitions according to the test. These statistics will be used to select the best test.

- **shouldSplit(j)** check if node j should be split. Denote $\mathcal{S}_j$ the set of samples that arrive at node j. The node is split when $|\mathcal{S}_j| > \alpha$ and exists a test $t \in T$ such that the information gain with respect to t $\Delta I(S_j) > \beta$.

Note that, once the decision for a split is performed and the test selected it cannot be further corrected. Thus, parameter α should be properly adjusted to maintain the tree growing process while new training data is still arriving, and not performing splitting decisions too early. If alpha is too low, trees will grow to the maximum depth early, and maybe relevant data arriving later will not have influence in training and the classifier would be over-fitted to a small part of the training set. In the opposite case, if α is too large the training process may end with the trees not grown to its maximum depth, which will affect to the detector accuracy and precision.

3.3.2 Hard negative mining using on-the-fly detection

The set of negative samples used for training is highly relevant for the performance of a detector. From the annotated images, collecting all the patches of the whole image is not practical, so methods rely on randomly sampling patches, as it is not clear in advance which patches from this images would be more useful as training samples. A common approach usually employed in order to improve the performance is to train the detector with a randomly selected set of examples, and then use the detector for mining hard negative

examples from the training images, i.e collecting the examples where the detector fails. This procedure is performed iteratively until some criterion is met. The boosted learning approach described above (sec. 3.2.3) follows this principle, where decision trees are used as base detectors.

Alternatively, in this thesis we propose a method to collect the negative patches from the training images using the prediction of the online forest during the training phase. In this manner, the training process is done in a single iteration and the set of examples used to update the trees is reduced, so that redundant or non informative patches are not used. The procedure is applied during training for each negative training image I_{neg} as follows:

1. The probability for the positive class c for each pixel x in I_{neg}, $p(c|x)$ is computed on-the-fly using the statistics collected at the current leaf nodes.

2. A pseudo-probability value for each pixel is computed using a Parzen estimator with a Gaussian kernel. Then we obtain the location with maximal probability m_c. We denote $maxp$ the probability at m_c.

3. A set of N_{neg} patches are collected within a neighborhood centered at m_c. The number of patches collected is proportional to $maxp$, so in this manner the worse is the failure of the detector, more negative samples which produce the failure are used for training.

In section 3.3.3 we show the improvement in average precision using this approach in comparison with random sampling of negative examples.

3.3.3 Experiments: Offline vs Online Comparison

The purpose of the following experiments is to validate the online random forest method for the gesture localization task, to analyze some of its parameters and to compare its performance with the offline counterpart.

Figure 3.4: Gestures performed by distinct subjects of the dataset. Gestures are respectively *Audio*, *Cross*, *Ok*, *Pause*, *Silence* and *Tee*.

We recorded several sequences of 8 actors performing 6 distinct gestures and also negative examples. We used the sequences of 4 actors for training, and the sequences of the other 4 were used for testing. Figure 3.4 shows example images of the dataset where the subjects are performing a gesture. The dataset recorded has a significant variability of viewpoints and scenarios.

We noticed an improve of performance of the online learning method when iterating the training process several times on the same dataset. Thus we iterate 5 times the method over the same dataset (The same procedure is employed in [101] iterating 10 times for evaluation on standard machine learning datasets). Summing up the 5 iterations, we end up using about 240K samples.

First we evaluated the influence of the parameter α for such sequences, i.e. the minimal number of samples visiting a node required to perform a split of the node. We found that a value of $\alpha = 300$ was the best performance. Although there were not big differences in average precision it was the best when considering also that memory usage was greater when using lower values of α. In Figure 3.5 we show the precision-recall curves, together with average precision values for the gesture *pause* and a range of α values.

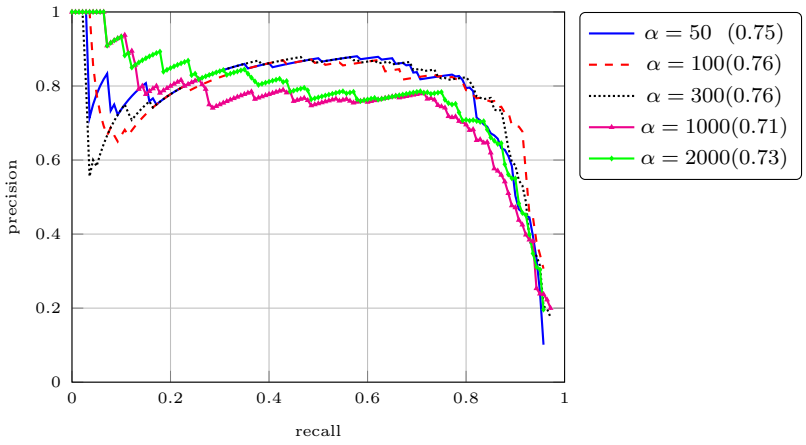


Figure 3.5: Performance curves for the *Pause* gesture for different values of parameter α. Average precision in parenthesis.

Method	Tee	Pause	Silence	Vol up	Cross	Ok
Offline Boosted	0.66	0.71	0.23	0.26	0.75	0.27
Online random	0.80	**0.81**	0.27	0.21	0.56	0.35
Online hard mining	**0.88**	0.74	**0.58**	**0.56**	**0.76**	**0.59**

Table 3.2: Average precision obtained for each of the gestures.

We also analyzed the influence of parameter β, i.e. the minimum information gain required to make a node split. We noticed that this condition did not improve the performance of the detector. This fact has also been reported by Schulter et.al. in [106] for a similar method for online-hough forests. We show the performance curves of gesture *pause* for several values of β in Figure 3.6. Note that, while the performance is not significantly different, the average precision is slightly better for $\beta = 0$.

We conducted further experiments to evaluate the performance of the online detector in comparison with the offline detector, both trained with the same dataset. Table 3.2 shows the average precision results obtained for the test sequences of each of the gestures. The performance of the online detector is better for most of the gestures evaluated or at least equivalent. Figure 3.7 show examples of successful detections using the online detector.

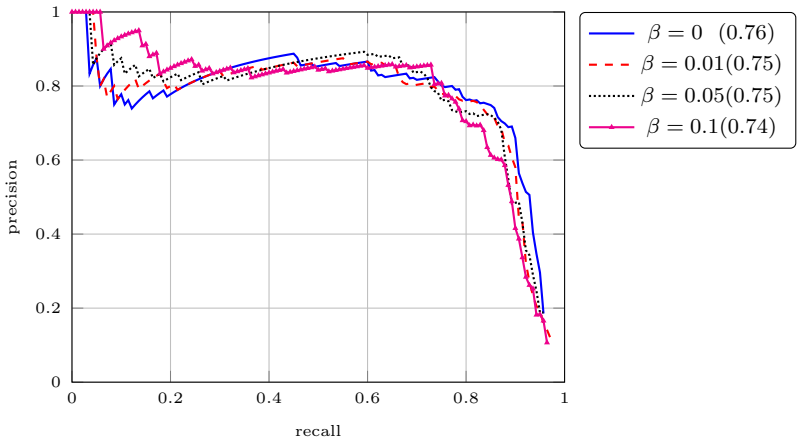


Figure 3.6: Performance curves for the *Pause* gesture for different values of parameter β. Average precision in parenthesis.

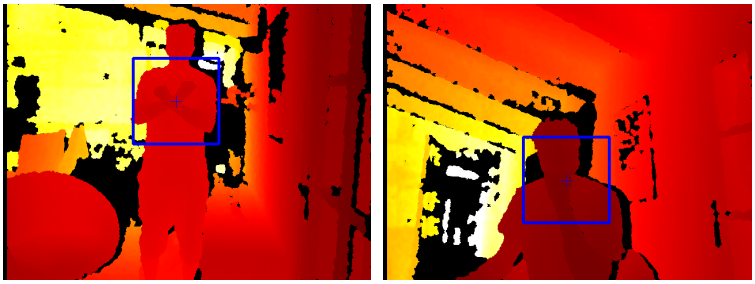


Figure 3.7: Successful detections for gestures *Cross* and *Silence* using the online detector, represented on the depth map.

Moreover, the proposed solution for hard negatives mining using on-the-fly detection improves significantly the results obtained. The improvement is more relevant for the gesture classes which are distinguishable by subtle or small shape differences, such as the *ok* or the *silence* gesture, where its main feature is the finger pose. In such cases, selectively collecting hard negative samples allows to train the detector more effectively with less amount of data, when compared with randomly sampling negative samples.

We have also analyzed the topology of the trees obtained. While it is not clear the influence of the topology of the trees in the accuracy of the detector, it has relevant impact in computational performance and memory usage. We have observed substantial

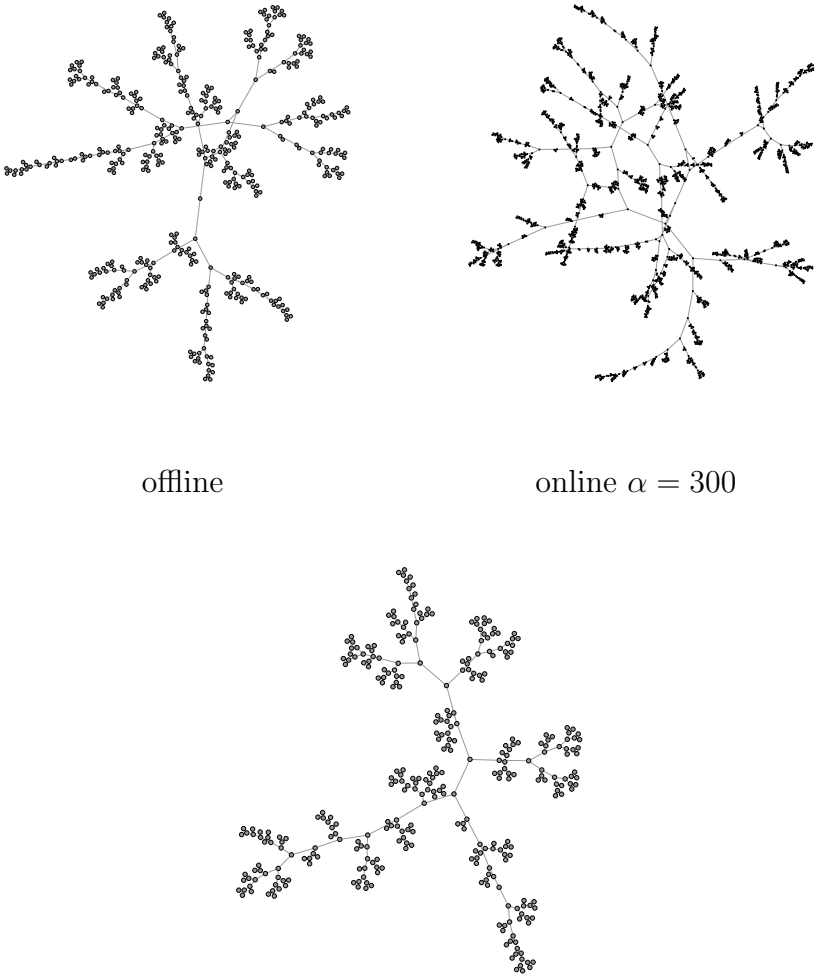


Figure 3.8: Graphical representation of a tree of the random forests trained offline and online.

differences between the topologies of the trees obtained when using the offline training approach compared to the trees obtained with online learning.

On the one side, as one could expect, the number of nodes of the trees increases as the parameter α decreases, i.e. when the number of required samples to make a split is lower. This implies more memory usage which is relevant in training time, because the number of

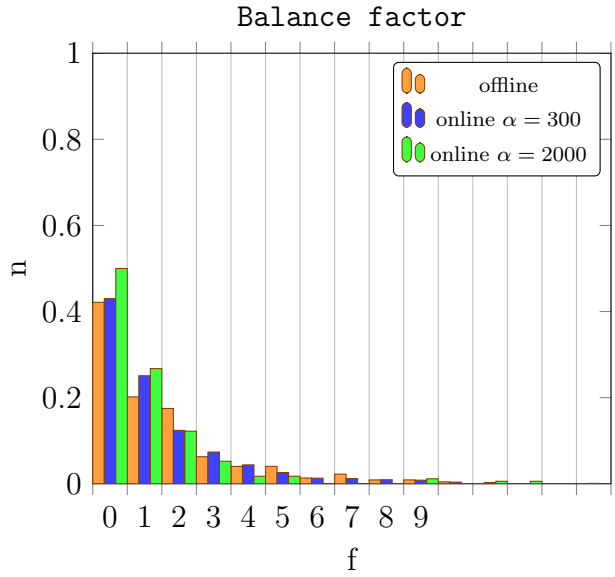


Figure 3.9: Histograms of the balance factor of the nodes of a tree for different training approaches.

bytes required per node is high, as we should maintain the statistics of the samples visiting the node while the split is not made. In test time the memory usage becomes relevant depending of the platform where the system would be deployed, and also when several classes should be detected.

Figure 3.8 shows a graphical representation of trees of the random forests trained online or offline and with distinct α values. Besides from the differences in the number of nodes, one can appreciate from the figures that trees resulting from the online learning approach are more balanced than the trees obtained offline, i.e. for each node the height of the left brach is similar or equal to the height of the right branch. Balanced trees are computationally more efficient. In general, when comparing trees with the same number of nodes, if the tree is balanced, less nodes should be visited to reach a leaf in comparison with its unbalanced equivalent.

To measure the balance of the trees we have computed the *balance factor* f of example

trees trained with distinct parameters. The balance factor accounts for each node, the difference between the height of the left branch and the height of the right branch. So, if the balance factor is 0, such node is balanced. In Figure 3.9 we show normalized histograms of the balance factor per node for three example trees trained in distinct manner. One can notice that the online approach result in more balanced trees, and such balance increases with α. Thus, greater *alpha* parameter favor computational efficiency and memory usage, but, as described above, greater α value reduces the precision and recall of the detector, which at the end compromises the final choice.

3.4 Interactive Machine Learning method for Gesture Localization Training

In this section we introduce an interactive machine learning (IML) method that allows to record and annotate data, and train the detector in a single step. The method focuses two main goals. On the one side, it reduces the effort and time usually required to record data, annotate and train an offline gesture detector. On the other side, it enables new related applications such as gesture customization, i.e. to enable the end-user of a gesture controlled application to define his own gestures to use them within the application. Such gesture customization can have a more generic usage such as online training of a custom object detector.

3.4.1 Training loop

The interactive gesture learning method consists of a main loop that captures depth frames from the camera, and it process such data to update the classifier and display some information depending on the state of the internal state machine. The user visualizes the depth map and some information overlaid such as text or bounding boxes. We have defined the

following set of states, with transitions described in Figure 3.10:

1. **do gesture**. The user is asked to perform the gesture to train centered in the bounding box suggested. The display shows a bounding box with a cross in the center.

2. **update positive**. The user is asked to keep the gesture centered in the bounding box while performing minor pose variations to introduce variability. During this phase the display shows the bounding box centered. Positive patches are extracted in the neighborhood of the bounding box center, and they are used to update the random forest with positive samples as explained in section 3.3.1.

3. **release**. The user is asked to release the gesture, and wait for more instructions.

4. **do negative examples**. The user is asked to perform other gestures that should be distinguished by the system. This phase can also be used to capture backgrounds and other kinds of negative examples. Negative patches are extracted using on-the-fly detection as described in section 3.3.2 and the random forest is updated with the corresponding negative samples. These patches are shown to user by means of a bounding box.

This sequence of states is executed several iterations to capture more variability of the gesture, such as distinct contexts, backgrounds, distances or slightly modified poses. Also more negative examples can be added at each iteration, which also improves generalization of the detector. At the end of a training session, the application launches a testing mode that allows to check the performance of the current detector and realize its main failures. The random forest also can be stored in disk, such that the application can be stopped, and the user can change the settings, the scenario, and other users can continue the training. The stored random forest is loaded and can be updated again with new data, so the

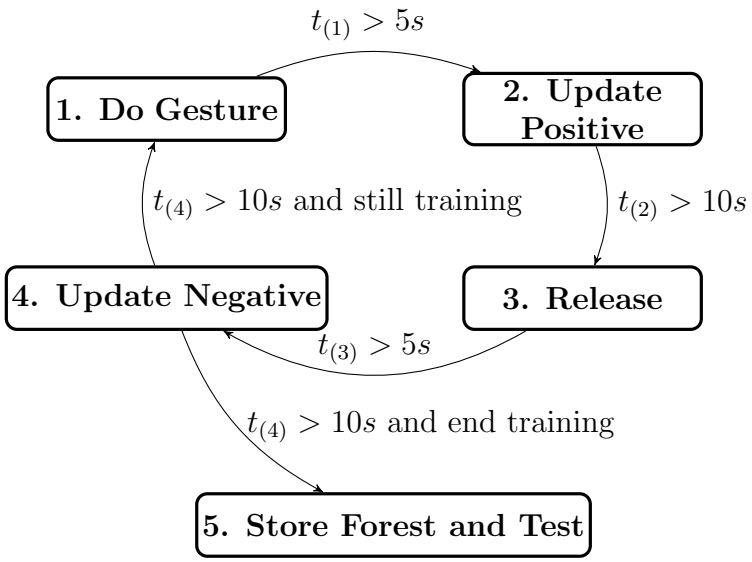
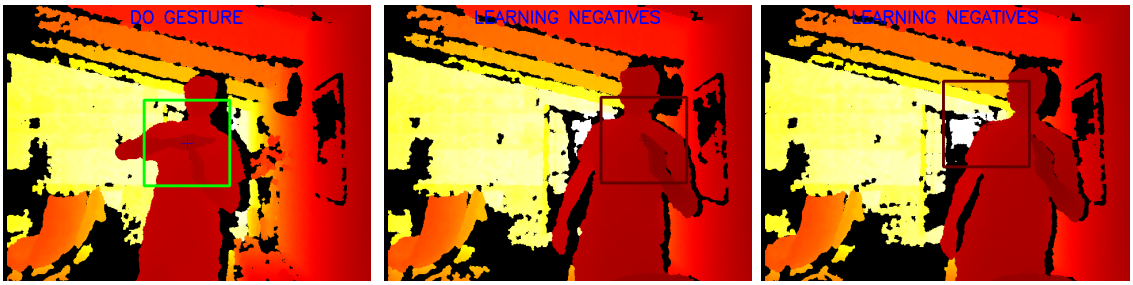


Figure 3.10: State diagram

trees continue growing from the point they were left in the previous iteration. This is a clear advantage of the online approach, so its incremental nature permits these pauses within the training period, that allow to test the current detector, and to continue training specifically with the settings and examples that induced more failures. Figure 3.11 shows example frames of the feedback displayed by the system. Note that in Fig. 3.11.b the online detector is detecting a maximum on the hand of the user, so the patch is used to update the forest with a negative sample. 6 frames after (Fig. 3.11.c) the detection is not on the hand of the user, thus the detector has been updated and this hand pose is not anymore a relevant patch. Such feedback is useful to the user as in real-time suggests which kind of poses are more informative for training the detector.

Obtaining the threshold For certain applications it might be convenient to automatically obtain a detection threshold, such that the user can use the detector immediately after training without manual adjustments. In this case, we can rely on the training phase to compute a threshold. Towards this goal, the detector is evaluated in the *update positive*



(a) (b) (c)

Figure 3.11: Images of the feedback shown to the user by the IML interface. The user is training the *Tee* gesture. (a) Feedback shown during *update positive* state. (b) and (c) Feedback shown during *do negative* state, where (c) is 6 frames after (b).

state, and the maximum pseudo-probability value obtained for each frame i is stored as th_i^+. Also, the detector is evaluated for each frame during the *update negative* state, and the maximum pseudo-probability value is stored as th_i^-. We compute at each frame the exponential moving average of both thresholds, such that

$$\hat{th}_i^+ = \sigma th_i^+ + (1-\sigma)\hat{th}_{i-1}^+$$

$$\hat{th}_i^- = \sigma th_i^- + (1-\sigma)\hat{th}_{i-1}^-$$

with $\sigma = 0.9$ for fast adaptation. Then the detection threshold is computed as a weighted mean of both $th = \rho\hat{th}^+ + (1-\rho)\hat{th}^-$ where ρ can be used to tune the sensitivity of the detector (e.g. $\rho = 0.8$ for low false positive rate).

3.4.2 Experiments

In order to evaluate the interactive machine learning method we have conducted a series of experiments to compare the presented method with the offline approach. On the one side we have evaluated the performance of the trained detector and on the other side we have measured the time required to train a detector for a single gesture in both cases.

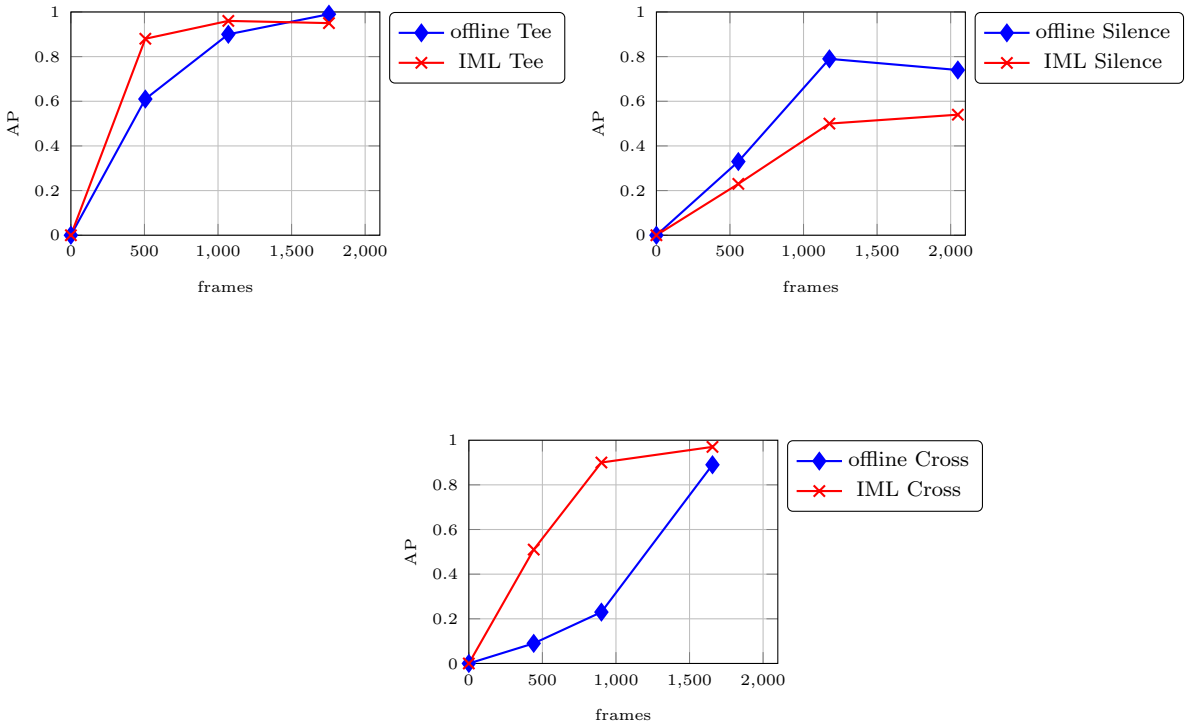


Figure 3.12: Average precision of the detector for gestures *Tee*, *Silence* and *Cross*. Detectors are trained from data captured on three consecutive sessions of about 500-700 frames per session. Figures show the precision obtained with the detector trained after each session, using offline training or the interactive machine learning (IML) approach.

To compare the performance of the method with the offline approach we have configured the interactive machined learning loop to record the captured frames and corresponding annotations. In this manner, the offline method can be trained with exactly the same images than the online approach for a fair comparison.

We have recorded test sequences of the same subject performing the selected gestures. Note that testing with the same subject than the training subject would require less generalization of the detector, thus less training data and training time. In this manner we can validate the approach for its usage in a custom gesture learning application. In the custom gesture learning use case, the focus is an application where the user would introduce a new

Training Session	Avg Prec	Record and Auto-Annotation	Training	Total Time
Offline (*short session*)	0.34	3:00	1:51	4:51
Offline (*medium session*)	0.64	6:00	4:16	10:16
Offline (*long session*)	0.87	9:00	7:59	16:59
IML (*short session*)	0.54	3:00	-	3:00
IML (*medium session*)	0.78	6:00	-	6:00
IML (*long session*)	0.82	9:00	-	9:00

Table 3.3: Average Precision and time employed for each of the learning tasks (min:sec). Mean average precision and times for gestures *Tee*, *Silence* and *Cross*.

gesture in the gesture-based interface, and in this case the requirements usually would be a short training period and robustness for a single subject.

The interactive method has been used to train a a gesture during three sessions of 3 min each. At the end of a session the random forest is stored and can be used for testing. The following session loads the stored random forest and continues the online training. Figure 3.12 shows the average precision of the detector trained using the interactive machine learning (IML) method, and the precision obtained when training the detector offline using the same frames and annotations. The figure shows results obtained after each of the three sessions for three distinct gestures, *Tee*, *Silence* and *Cross*.

In average, the results with respect to accuracy are comparable between the online and offline training approach, but the time spend for the whole training process is considerably shorter. In table 3.4.2 the average precision results are summarized for the gestures evaluated, together with the training time employed. Note that we consider automatic annotation for both offline and IML. Such automatic annotation simply consists in providing the user a bounding box overlayed on the camera view, and asking the user to perform the gesture centered in the bounding box. Then the frame captured is stored together with the bounding box position. This strategy substantially reduces the annotation effort usually required for object detection datasets, for instance, first recording and then marking

the gesture positions by hand on the stored frames. Also note that the fact that in the IML approach all tasks are performed at once, the time devoted to the task is constant, and would be increased always by the same constant (3min) to include more data with each extra session. On the other side, in the offline case one will be switching tasks, from recording, waiting the training process to finish, testing and recording more data if needed, which in general is more tedious and require a more experienced user.

3.4.3 Conclusions

In this chapter we have proposed solutions to the problem of gesture localization on depth data. Mainly we focused on reducing the effort required to train the detectors. At the end, solving this issue can habilitate the development of new applications such as custom gesture training by non experienced users.

First of all, we have analyzed the solution for gesture localization proposed in [82] and we have introduced depth clipping in the random forests tests. Experimental results demonstrate an improvement in precision, mainly due to an increase of robustness in presence of clutter or people in background.

Towards the goal of an interactive machine learning approach to gesture detector training, we have studied an online random forest learning approach. Within such online learning framework, we have proposed a method for hard negative mining using the detector on-the-fly while training. On-the-fly hard negative mining have demonstrated superior precision than random sampling negative samples. Also, the results obtained in the recorded datasets show better precision of the online approach than using offline learning.

We have proposed an interactive machine learning (IML) method that allows to reduce the time and effort required to train a new gesture detector. Experimental results towards custom gesture training for a single user demonstrate that the IML approach have almost equivalent precision than the offline training approach, and it contrast requires less time

and can be employed by a non experienced user.

CHAPTER 4

Conclusions

In this thesis we have studied the problem of human motion analysis from visual data, where the main goal has been the obtention of the body pose. In the first part, the human pose has been described by a full body skeleton. In the second part, we have focused the problem to the detection of specific poses, mainly involving hand and arm pose, also seen as gestures.

4.1 Contributions

1. **Marker-less Human Motion Capture** Human pose estimation is a complex problem that has been studied from the whole range of pattern analysis and machine learning perspectives. We have made contributions to the field focusing on specific parts of the whole problem. The contributions are summarized as follows:

    - We propose a generic framework for hierarchically layered particle filtering

(HPF) specially suited for motion capture tasks. Human motion capture problem generally involve tracking or optimization of high-dimensional state vectors where also one have to deal with multi-modal *pdf*s. HPF allow to overcome the problem by means of multiple passes through substate space variables. Such framework provides a guide for a practical implementation which can be configured to resolve distinct tasks such as pose tracking or anthropometry estimation. It is formulated taking into account the flexibility to exchange its main components, the transition kernel and the weighting function, in order to adapt them to new features or cues related to specific scenarios.

- Based on the HPF framework, we have proposed a method to estimate the anthropometry of the subject, which at the end allows to obtain a human body model adjusted to the subject. Such body model will be used in subsequent motion capture tasks. The method optimize the pose and anthropometric parameters of the subject using silhouettes from multiple views.

- We propose a new weighting function strategy for approximate partitioning of observations which is integrated within the HPF framework. The APO-HPF allows to partition the observation space for specific body parts, which has demonstrated an improvement in accuracy because it ends up with more robustness to self-occlusions of parts and presence of background clutter.

- Alternatively, we propose the DD-HPF, which employ body part detections to improve particle propagation and weight evaluation. This technique has demonstrated better accuracy in tasks involving tracking of complex arms movements, where the baseline method using propagation based on Gaussian transition kernel failed to track the motions.

Publications The publications related to the marker-less motion capture contributions are:

- M. Alcoverro, J. Casas, and M. Pardàs, Skeleton and shape adjustment and tracking in multicamera environments, in *6th Int. Conf. AMDO 2010, 2010, pp. 88-97.*
- M. Alcoverro, A. López-Méndez, J. Casas, and M. Pardàs, A real-time body tracking system for smart rooms, in *ICME - 2011 IEEE International Conference on Multimedia and Expo, 2011, pp. 1-6.*
- A. López-Méndez, M. Alcoverro, M. Pardàs, and J. Casas, Approximate partitioning of observations in hierarchical particle filter body tracking, in *2011 IEEE Conf. on Computer Vision and Pattern Recognition Workshops, 2011, pp. 19-24.*
- A. López-Méndez, M. Alcoverro, M. Pardàs, and J. Casas, Real-time upper body tracking with online initialization using a range sensor, in 2011 IEEE International Conference on Computer VIsion Workshops (ICCV Workshops), 2011, pp. 391398.
- S. Navarro, A. López-Méndez, M. Alcoverro, and J. Casas, Multi-view Body Tracking with a Detector-Driven Hierarchical Particle Filter, in 7th International Conference AMDO 2012, Port d'Andratx, Mallorca, 2012.

2. **Interactive Machine Learning for Gesture Localization** The second part of this thesis is centered in the detection of specific poses, mainly gestures, and we have focused the problem of reducing annotation and training efforts required to train a specific gesture. The gesture localization solution proposed by López-Méndez and J. Casas [82] based on random forests learning from depth data is the starting point of our proposed methods. The contributions are summarized as follows:

   - First, we propose the depth clipping test. Clipping depth in the sample patch has demonstrated improvement in accuracy in presence of background clutter. Given that the body parts conforming a gesture can be delimited by a bounding

box, by clipping depth according to such bounding box parameters the information obtained by a sample is more relevant to the gesture.

- In order to reduce the efforts required to train a gesture detector, we propose a solution based on online random forests that allows training in real-time, while receiving new data in sequence. The main aspect that makes the solution effective is the method we propose to collect the hard negatives examples while training the forests. The method uses the detector trained up to the current frame to test on that frame, and then collects samples based on the response of the detector such that they will be more relevant for training. In this manner, training is more effective in terms of the number of annotated frames required. We have demonstrated better accuracy on comparison with the offline random forest approach when using the same dataset.
- We propose a method for training a gesture interactively. The method defines an application loop that allows to annotate and train the detector at the same time. Such, approach is specially suited for gesture customization within gesture based interfaces, and can be used by a non experienced user. In single user tests, we have demonstrated that training time and effort is substantially reduced in comparison with and offline approach, and the accuracy is equivalent.

Publications The following publication propose an interface using the gesture detector based on depth-clipping tests:

- M. Alcoverro, X. Suau, J. R. Morros, A. López-Méndez, A. Gil-Moreno, J. Ruiz-Hidalgo, and J. Casas, Gesture Control Interface for immersive panoramic displays, *Multimedia Tools and Applications, pp. 1-27, 2013.*

4.2 Side Contributions

During the development of this thesis we have contributed with other methods in the human motion analysis field that have been reported in the following publications:

- **Voxel occupancy with viewing line inconsistency analysis and spatial regularization** Shape reconstruction from multiple cameras enables further analysis of the 3D scene. We proposed a method for volume reconstruction from silhouettes where we use a new viewing line based inconsistency analysis within a probabilistic framework. Our method adds robustness to errors by projecting back to the views the volume occupancy obtained from 2D foreground detections intersection, and then analysing this projection.

  - M.Alcoverro and M. Pardàs. Voxel Occupancy with Viewing Line Inconsistency Analysis and Spatial Regularization. in *VISAPP 2009 - Fourth International Conference on Computer Vision Theory and Applications, Lisboa, Portugal, February 5-8, 2009.*

- **Connected Operators on 3D data for human body analysis** We have studied the problem of detection of salient points in 3D data. The proposed algorithm consists in processing the geodesic distances on a 3D surface representing the human body in order to find prominent maxima representing salient points of the human body. We introduce a 3D surface graph representation and filtering strategies to enhance robustness to noise and artifacts present in this kind of data. Such approach has been successfully employed for end-effector detection within the HPF method proposed in [80].

  - M. Alcoverro, A. López-Méndez, M. Pardàs, and J. Casas, Connected Operators on 3D data for human body analysis, in *2011 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2011, pp. 9-14.*

- **Fingertip Localization and Hand Gesture Classification** In addition to the gesture localization approach presented in this thesis, we have also proposed another approach related to gesture classification using depth data, in the context of the thesis of X. Suau. In this case, we focus the problem of fingertip localization. The method obtains the most probable fingertip locations conditioned on the obtained hand gesture by means of KNN search.

  - X. Suau, M. Alcoverro, A. López-Méndez, J. Ruiz-Hidalgo, J. Casas, Real-time Fingertip Localization Conditioned on Hand Gesture Classification, *Image and Vision Computing, vol. 32, no. 8, 2014.*

4.3 Future Work

In this thesis we have proposed methods that provided good results for a set of human motion analysis problems. Even though, there is still problems that require more research in order to find good solutions, where our work can serve as basis.

Markerless motion capture in controlled scenarios with an accurate video capturing infrastructure is a solved problem, at least with respect to the capture of the pose [78],[116]. Research in this field is moving towards accurate capture of shape, clothing movements and illumination [127]. However, pose estimation under general conditions, where background is not controlled and few views are available, is still a challenging problem. These would be the requirements of human motion analysis systems in outdoor scenarios or in presence of occluders. Solving pose estimation in such conditions would enable the application of human motion analysis techniques in fields such as sports, safety in industrial facilities, video surveillance or elderly and handicapped assistance. In such scenarios, the combination of stochastic optimization and body part detectors seems a promising line of research. The DD-HPF method presented in this thesis can be used combined with

the state-of-the-art body part detectors that have been recently proposed for single view pose estimation [89]. Such combination of discriminative and generative approaches fusing the information of distinct viewpoints can add robustness to occlusions and background clutter. Also it could be an step forward into a lesser dependency on silhouettes, which are not reliable in presence of background clutter.

Current depth sensors facilitate the task of pose estimation in close range scenarios where the user is facing the camera. This fact is enabling new kinds of interfaces with computers or other devices based on hand gestures. The approach for interactive learning of gestures presented in this thesis could be used to bring gesture localization solutions closer to interface designers. Towards this goal, a line of research would be required to fully automate the annotation process such that the user would not require to perform the gesture within the bounding box presented in screen. That would allow a generalization of the approach to learn other kind of shapes or events, as for example objects, or combinations of a pose and an object, not being constrained to perform the action, or place the object in a specific point. A method would be required to extract the relevant positive patches during the training phase. Using other cues such as skeleton pose or motion flow could bring insights on the location of the positive patch, in combination of data previously learned.

Moreover, extending the gesture localization framework to dynamic gestures would be also an interesting line of research. Usually our natural forms of gesture communication involve a dynamic component, so including such gestures in gesture-based interfaces could make them more intuitive.

Bibliography

[1] "Actibio (unobtrusive authentication using activity related and soft biometrics)," http://www.actibio.eu/, sTREP 215372, Call: FP7-ICT-2007-1.

[2] "Fascinate (format-agnostic script-based interactive experience)," http://www.fascinate-project.eu/, ref.: IP 248138, Call: FP7-ICP-2009-1.5 Period: 1.2.2010 - 31.7.2013.

[3] "Organic motion," http://www.organicmotion.com.

[4] "Vicon," http://www.vicon.com.

[5] "Vision (comunicaciones de vídeo de nueva generación)," (CENIT 2007-1007) project of the Spanish Ministry of Industry. Ingenio 2010.

[6] M. Alcoverro, X. Suau, J. Morros, A. López-Méndez, A. Gil-Moreno, J. Ruiz-Hidalgo, and J. Casas, "Gesture control interface for immersive panoramic displays," *Multimedia Tools and Applications*, pp. 1–27, 07/2013 2013.

[7] M. Alcoverro, "Models and estimators for markerless human motion tracking," Master's thesis, Universitat Politècnica de Catalunya, June 2009. [Online]. Available: http://hdl.handle.net/2099.1/7191

[8] M. Alcoverro, J. R. Casas, and M. Pardàs, "Skeleton and Shape Adjustment and Tracking in Multicamera Environments," in *AMDO*, ser. Lecture Notes in Computer Science, F. J. P. López and R. B. Fisher, Eds., vol. 6169. Springer, 2010, pp. 88–97.

[9] S. Amershi, J. Fogarty, and D. Weld, "Regroup: Interactive machine learning for on-demand group creation in social networks," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*. ACM, 2012, pp. 21–30.

[10] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "SCAPE: shape completion and animation of people," *SIGGRAPH '05*, vol. 24, no. 3, pp. 408–416, 2005.

[11] M. Ankerst, C. Elsen, M. Ester, and H.-P. Kriegel, "Visual classification: an interactive approach to decision tree construction," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 392–396.

[12] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and S. Adelaide, "A tutorial on particle filters for online nonlinear/non-GaussianBayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[13] A. Balan, L. Sigal, M. Black, J. Davis, and H. Haussecker, "Detailed Human Shape and Pose from Images," in *IEEE CVPR'07*, 2007, pp. 1–8.

[14] L. Ballan and G. M. Cortelazzo, "Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes," in *3DPVT*, Atlanta, GA, USA, June 2008.

[15] J. Bandouch, F. Engstler, and M. Beetz, "Accurate Human Motion Capture Using an Ergonomics-Based Anthropometric Human Model," in *AMDO'08*. Springer, 2008, p. 248.

[16] J. Bandouch and M. Beetz, "Tracking humans interacting with the environment using efficient hierarchical sampling and layered observation models," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, Sep. 2009, pp. 2040–2047.

[17] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," in *SIGGRAPH '07*. ACM, 2007, p. 72.

[18] M. Bergtholdt, J. Kappes, and S. Schmidt, "A Study of Parts-Based Object Class Detection Using Complete Graphs," *International Journal of Computer Vision*, pp. 93–117, 2010.

[19] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

[20] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[21] J. Bouguet, "Camera calibration toolbox for matlab," 2004.

[22] R. Boulic, J. Varona, L. Unzueta, M. Peinado, A. Suescun, and F. Perales, "Evaluation of on-line analytic and numeric inverse kinematics approaches driven by partial vision input," *Virtual Reality*, vol. 10, no. 1, pp. 48–61, 2006.

[23] C. Bregler, J. Malik, and K. Pullen, "Twist based acquisition and tracking of animal and human kinematics," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 179–194, 2004.

[24] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[25] C. Canton-Ferrer, J. R. Casas, and M. Pardàs, "Exploiting structural hierarchy in articulated objects towards robust motion capture," in *AMDO'08*. Springer, 2008, pp. 82–91.

[26] J. Carranza, C. Theobalt, M. Magnor, and H. Seidel, "Free-viewpoint video of human actors," *ACM TOG*, vol. 22, no. 3, pp. 569–577, 2003.

[27] C. O. Conaire, P. Kelly, D. Connaghan, and N. E. O'Connor., "TennisSense: A Platform for Extracting Semantic Information from Multi-camera Tennis Data," in *DSP 2009*, 2009, pp. 1062–1067.

[28] S. Corazza, L. Mündermann, A. Chaudhari, T. Demattio, C. Cobelli, and T. Andriacchi, "A markerless motion capture system to study musculoskeletal biomechanics: Visual hull and simulated annealing approach," *Annals of biomedical engineering*, vol. 34, no. 6, pp. 1019–1029, 2006.

[29] P. Correa Hernandez, J. Czyz, F. Marques, T. Umeda, X. Marichal, and B. Macq, "Bayesian Approach for Morphology-Based 2-D Human Motion Capture," *Multimedia, IEEE Transactions on*, vol. 9, no. 4, pp. 754–765, 2007.

[30] N. Courty and E. Arnaud, "Inverse Kinematics Using Sequential Monte Carlo Methods," in *Articulated Motion and Deformable Objects*, ser. Lecture Notes in Computer Science, F. Perales and R. Fisher, Eds. Springer Berlin / Heidelberg, 2008, vol. 5098, pp. 1–10.

[31] H. Cuayáhuitl, M. van Otterlo, N. Dethlefs, and L. Frommberger, "Machine learning for interactive systems and robots: a brief introduction," in *Proceedings of the 2nd Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Perception, Action and Communication*. ACM, 2013, pp. 19–28.

[32] Cyberware, "www.cyberware.com." [Online]. Available: www.cyberware.com

[33] Q. Delamarre and O. Faugeras, "3D articulated models and multiview tracking with physical forces," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 328–357, 2001.

[34] D. Demirdjian and C. Varri, "Recognizing events with temporal random forests," in *ICMI-MLMI*. New York, NY, USA: ACM, 2009, pp. 293–296.

[35] M. Denil, D. Matheson, and N. de Freitas, "Consistency of online random forests," *arXiv preprint arXiv:1302.4853*, 2013.

[36] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *IEEE CVPR'00*, vol. 2, 2000, pp. 126–133.

[37] J. Deutscher and I. Reid, "Articulated body motion capture by stochastic search," *International Journal of Computer Vision*, vol. 61, no. 2, pp. 185–205, 2005.

[38] DINBelg 2005. Body dimensions of the Belgian population., "http://www.dinbelg.be/anthropometry.htm."

[39] S. Duffner, J.-M. Odobez, and E. Ricci, "Dynamic partitioned sampling for tracking with discriminative features," in *Proceedings of the British Maschine Vision Conference*, no. LIDIAP-CONF-2009-010, 2009.

[40] H. Egashira, A. Shimada, D. Arita, and R. Taniguchi, "Vision-based motion capture of interacting multiple people," in *Image Analysis and Processing ICIAP 2009*, ser.

Lecture Notes in Computer Science, P. Foggia, C. Sansone, and M. Vento, Eds. Springer Berlin / Heidelberg, 2009, vol. 5716, pp. 451–460.

[41] J. A. Fails and D. R. Olsen Jr, "Interactive machine learning," in *Proceedings of the 8th international conference on Intelligent user interfaces*. ACM, 2003, pp. 39–45.

[42] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial Structures for Object Recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[43] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Pose search: Retrieving people using their pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[44] J. Fogarty, D. Tan, A. Kapoor, and S. Winder, "Cueflik: interactive concept learning in image search," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 29–38.

[45] R. Francese, I. Passero, and G. Tortora, "Wiimote and kinect: gestural user interfaces add a natural third dimension to hci," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ser. AVI '12. New York, NY, USA: ACM, 2012, pp. 116–123.

[46] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, pp. 209–226, September 1977.

[47] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *TPAMI*, vol. 33, no. 11, pp. 2188 –2202, 2011.

[48] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel, "Optimization and Filtering for Human Motion Capture," *International Journal of Computer Vision*, vol. 87, no. 1-2, pp. 75–92, 2008.

[49] J. Gall, J. Potthoff, C. Schnörr, B. Rosenhahn, and H.-P. Seidel, "Interacting and annealing particle filters: Mathematics and a recipe for applications," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 1, pp. 1–18, 2007.

[50] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 755–762.

[51] D. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[52] D. Gavrila and L. Davis, "3-D model-based tracking of humans in action: a multi-view approach," in *1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96*, 1996, pp. 73–80.

[53] "Gesturepak: Gesture recording and recognition toolkit," http://www.franklins.net/gesturepak.aspx, accessed: 20/02/2013.

[54] L. Guan, J.-S. Franco, and M. Pollefeys, "3D Occlusion Inference from Silhouette Cues," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1–8.

[55] V. Gulshan, V. Lempitsky, and A. Zisserman, "Humanising grabcut: Learning to segment humans using the kinect," in *ICCV-CDC4CV,*, 2011.

[56] G. Haro and M. Pardàs, "Shape from incomplete silhouettes based on the reprojection error," *IMAGE AND VISION COMPUTING*, pp. 1–15, 2010.

[57] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H.-P. Seidel, "Markerless Motion Capture with unsynchronized moving cameras," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2009, pp. 224–231.

[58] S. r. Hauberg, J. Lapuyade, M. Engell-Nø rregård, K. Erleben, and K. Steenstrup Pedersen, "Three Dimensional Monocular Human Motion Analysis in End-Effector Space," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, ser. Lecture Notes in Computer Science, D. Cremers, Y. Boykov, A. Blake, and F. Schmidt, Eds. Springer Berlin / Heidelberg, 2009, vol. 5681, pp. 235–248.

[59] S. Hauberg and K. S. Pedersen, "Predicting articulated human motion from spatial processes," *IJCV*, vol. 94, no. 3, pp. 317–334, 2011.

[60] R. Hoffmann, S. Amershi, K. Patel, F. Wu, J. Fogarty, and D. S. Weld, "Amplifying community content creation with mixed initiative information extraction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 1849–1858.

[61] J. Hu and J. Hua, "Salient spectral geometric features for shape matching and retrieval," *The Visual Computer*, vol. 25, no. 5, pp. 667–675, 2009.

[62] M. Isard and A. Blake, "Condensation: conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[63] D. Jacka, A. Reid, B. Merry, and J. Gain, "A comparison of linear skinning techniques for character animation," in *Proceedings of the 5th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*. ACM New York, NY, USA, 2007, pp. 177–186.

[64] A. Jaume-i Capó, J. Varona, M. González-Hidalgo, and F. J. Perales, "Adding image constraints to inverse kinematics for human motion capture," *EURASIP J. Adv. Signal Process*, vol. 2010, pp. 1–13, 2010.

[65] M. Jones and J. Rehg, "Statistical color models with application to skin detection," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 1. IEEE, 1999.

[66] M. Kallmann, "Analytical inverse kinematics with body posture control," *Comput. Animat. Virtual Worlds*, vol. 19, no. 2, pp. 79–91, May 2008.

[67] S. Katz, G. Leifman, and A. Tal, "Mesh segmentation using feature point and core extraction," *The Visual Computer*, vol. 21, no. 8-10, pp. 649–658, Sep. 2005.

[68] M. Keck and J. W. Davis, "3D occlusion recovery using few cameras," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2008, pp. 1–8.

[69] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4. IEEE, 1995, pp. 1942–1948.

[70] "Kinect for windows sdk," http://www.microsoft.com/en-us/kinectforwindows/develop/, accessed: 20/02/2013.

[71] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi *et al.*, "Optimization by simmulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[72] W. B. Knox and P. Stone, "Reinforcement learning from simultaneous human and mdp reward," in *Proceedings of the 11th International Conference on Autonomous*

Agents and Multiagent Systems-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 475–482.

[73] P. Kohli, J. Rihan, M. Bray, and P. Torr, "Simultaneous segmentation and pose estimation of humans using dynamic graph cuts," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 285–298, 2008.

[74] J.-L. Landabaso, M. Pardás, and J. R. Casas, "Shape from inconsistent silhouette," *Computer Vision and Image Understanding*, vol. 112, no. 2, pp. 210–224, 2008.

[75] A. Laurentini, "The Visual Hull Concept for Silhouette-Based Image Understanding," *IEEE Trans. PAMI*, vol. 16, no. 2, pp. 150–162, 1994.

[76] J. Lewis, M. Cordner, and N. Fong, "Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation," in *SIGGRAPH '00.* ACM New York, NY, USA, 2000, pp. 165–172.

[77] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657 – 675, 2009, ¡ce:title¿PerCom 2009¡/ce:title¿.

[78] Y. Liu, C. Stoll, J. Gall, H.-P. Seidel, and C. Theobalt, "Markerless motion capture of interacting characters using multi-view image segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* IEEE, 2011, pp. 1249–1256.

[79] A. López and J. R. Casas, "Feature-based annealing particle filter for robust body pose estimation," in *VISAPP 2009 - Proceedings of the Fourth International Conference on Computer Vision Theory and Applications, Lisboa, Portugal*, A. Ranchordas and H. Araújo, Eds., vol. 2. INSTICC Press, 2009, pp. 438–443.

[80] A. López-Méndez, M. Alcoverro, M. Pardàs, and J. Casas, "Real-time upper body tracking with online initialization using a range sensor," in *2011 IEEE International Conference on Computer VIsion Workshops (ICCV Workshops)*, 2011, p. 391–398.

[81] A. Lopez-Mendez, M. Alcoverro, M. Pardas, and J. Casas, "Approximate partitioning of observations in hierarchical particle filter body tracking," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, June 2011, pp. 19–24.

[82] A. López-Méndez and J. R. Casas, "Can our tv robustly understand human gestures?: real-time gesture localization in range data," in *Proceedings of the 9th European Conference on Visual Media Production*, ser. CVMP '12. New York, NY, USA: ACM, 2012, pp. 18–25.

[83] J. MacCormick and M. Isard, "Partitioned Sampling, Articulated Objects, and Interface-Quality Hand Tracking," in *ECCV'00*. Springer-Verlag London, UK, 2000, pp. 3–19.

[84] Microsoft Kinect for the Xbox, "http://www.xbox.com/kinect."

[85] I. Mikić, M. Trivedi, E. Hunter, and P. Cosman, "Human body model acquisition and tracking using voxel data," *International Journal of Computer Vision*, vol. 53, no. 3, pp. 199–223, 2003.

[86] T. Moeslund and E. Granum, "A Survey of Computer Vision-Based Human Motion Capture," *Computer Vision and Image Understanding*, vol. 81, pp. 231–268, 2001.

[87] S. Navarro, A. López-Méndez, M. Alcoverro, and J. Casas, *Multi-view Body Tracking with a Detector-Driven Hierarchical Particle Filter*, ser. Lecture Notes in Computer Science. Berlin / Heidelberg: Springer, 2012, vol. 7378, pp. 82–91.

[88] "Openni sdk," http://www.openni.org/openni-sdk/, accessed: 20/02/2013.

[89] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Poselet conditioned pictorial structures," in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Portland, Oregon: IEEE, 2013, pp. 1–8.

[90] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time Identification and Localization of Body Parts from Depth Images," in *IEEE Int. Conference on Robotics and Automation (ICRA), Anchorage, Alaska, USA*, 2010.

[91] R. Plankers and P. Fua, "Tracking and modeling people in video sequences," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 285–302, 2001.

[92] R. Poppe, "Vision-based human motion analysis: An overview," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 4–18, 2007.

[93] ——, "Vision-based human motion analysis: An overview," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 4–18, 2007.

[94] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical recipes in C: the art of scientific computing*, 1992.

[95] N. Pugeault and R. Bowden, "Spelling It Out: Real-Time ASL Fingerspelling Recognition," in *ICCV-CDC4CV*, 2011.

[96] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Strike a pose: tracking people by finding stylized poses," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 2005, pp. 271 – 278 vol. 1.

[97] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *ACM MM*. New York, NY, USA: ACM, 2011, pp. 1093–1096.

[98] J. Rodgers, D. Anguelov, H.-C. Pang, and D. Koller, "Object Pose Detection in Range Scan Data," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 2006, pp. 2445–2452.

[99] B. Rosenhahn, T. Brox, and J. Weickert, "Three-Dimensional Shape Knowledge for Joint Image Segmentation and Pose Tracking," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 243–262, Sep. 2006.

[100] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[101] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1393–1400.

[102] J. Salvador, X. Suau, and J. R. Casas, "From silhouettes to 3d points to mesh: towards free viewpoint video," in *Proceedings of the 1st international workshop on 3D video processing*, ser. 3DVP '10, 2010, pp. 19–24.

[103] "Samsung smart tv interaction," http://www.youtube.com/watch?v=FRk5bMDs0Iw&feature=youtu.be, accessed: 23/02/2013.

[104] A. Schick and R. Stiefelhagen, "Real-Time GPU-Based Voxel Carving with Systematic Occlusion Handling," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, J. Denzler, G. Notni, and H. Süß e, Eds. Springer Berlin / Heidelberg, 2009, vol. 5748, pp. 372–381.

[105] T. Schlömer, B. Poppinga, N. Henze, and S. Boll, "Gesture recognition with a wii controller," in *Proceedings of the 2nd international conference on Tangible and embedded interaction*, ser. TEI '08. New York, NY, USA: ACM, 2008, pp. 11–14.

[106] S. Schulter, C. Leistner, P. M. Roth, H. Bischof, and L. J. Van Gool, "On-line hough forests." in *BMVC*, 2011, pp. 1–11.

[107] B. Settles, "Active learning literature survey," University of Wisconsin, Madison, Tech. Rep.

[108] 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," in *CVPR*, 2011, pp. 1297 –1304.

[109] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *CVPR*, 2008, pp. 1 –8.

[110] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard, "Tracking loose-limbed people," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* IEEE, 2004, pp. 421–428.

[111] L. Sigal, A. Balan, and M. Black, "HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion," *International Journal of Computer Vision*, vol. 87, no. 1, pp. 4–27, 2010.

[112] L. Sigal and M. Black, "Guest Editorial: State of the Art in Image- and Video-Based Human Pose and Motion Estimation," *International Journal of Computer Vision*, vol. 87, no. 1, pp. 1–3, 2010.

[113] V. K. Singh and R. Nevatia, "Monocular human pose tracking using multi frame part dynamics," in *2009 Workshop on Motion and Video Computing (WMVC).* IEEE, Dec. 2009, pp. 1–8.

[114] P. Sloan, C. Rose III, and M. Cohen, "Shape by example," in *Proceedings of the 2001 symposium on Interactive 3D graphics*. ACM New York, NY, USA, 2001, pp. 135–143.

[115] R. C. H. Solutions, "www.ramsis.de." [Online]. Available: www.ramsis.de

[116] C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt, "Fast articulated motion tracking using a sums of gaussians body model," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 951–958.

[117] J. Talbot, B. Lee, A. Kapoor, and D. Tan, "Ensemblematrix: Interactive visualization to support machine learning with multiple classifiers," in *Proceedings of the 27th international conference on Human factors in computing systems*. ACM, 2009, pp. 1283–1292.

[118] A. L. Thomaz and M. Cakmak, "Learning about objects with human teachers," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*. ACM, 2009, pp. 15–22.

[119] A. L. Thomaz and C. Breazeal, "Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance," in *AAAI*, vol. 6, 2006, pp. 1000–1005.

[120] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the ninth ACM international conference on Multimedia*. ACM, 2001, pp. 107–118.

[121] D. Uebersax, J. Gall, M. Van den Bergh, and L. Van Gool, "Real-time Sign Language Letter and Word Recognition from Depth Data," in *ICCV-HCI*, 2011, pp. 1–8.

[122] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511 – I–518 vol.1.

[123] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," *ACM TOG*, vol. 27, no. 3, pp. 1–9, 2008.

[124] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Commun. ACM*, vol. 54, no. 2, pp. 60–71, Feb. 2011.

[125] M. Ware, E. Frank, G. Holmes, M. Hall, and I. H. Witten, "Interactive machine learning: letting users build classifiers," *International Journal of Human-Computer Studies*, vol. 55, no. 3, pp. 281–292, 2001.

[126] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 249–257, 2006.

[127] C. Wu, K. Varanasi, Y. Liu, H.-P. Seidel, and C. Theobalt, "Shading-based dynamic shape refinement from multi-view video under general illumination," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1108–1115.

[128] L. Xu, J. Landabaso, and M. Pardas, "Shadow removal with blob-based morphological reconstruction for error correction," in *IEEE ICASSP'05*, vol. 2, 2005.

[129] A. Yao, J. Gall, C. Leistner, and L. Van Gool, "Interactive object detection," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3242–3249.

[130] X. Zhang, W. Hu, S. Maybank, X. Li, and M. Zhu, "Sequential particle swarm optimization for visual tracking," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.

[131] "Zigfu. motion controlled web," http://zigfu.com, accessed: 20/02/2013.

[132] G. Zou, J. Hua, M. Dong, and H. Qin, "Surface matching with salient keypoints in geodesic scale space," *Computer Animation and Virtual Worlds*, vol. 19, no. 3-4, pp. 399–410, 2008.