UNIVERSITAT POLITÈCNICA
DE CATALUNYA

PhD Dissertation

Human Motion Capture with Scalable Body Models

Cristian Canton Ferrer

Thesis supervisors:
Dr. Josep Ramon Casas Pla
Dra. Montse Pardàs Feliu

Department of Signal Theory and Communications
Universitat Politècnica de Catalunya

Barcelona, June 2009

Abstract

Capturing and tracking human motion is becoming a hot research topic due to the number of applications that can be addressed using this information, ranging from action recognition, human-computer interfaces and biometrics. This PhD thesis addresses the problem of extracting the pose parameters of a human body in a multi-camera environment relying on Monte Carlo techniques.

Extracing the describing parameters (pose) of an articulated model of the human body from information provided by multiple cameras can be efficiently tackled using the standard Bayesian prediction and update formulation. However, due to the high dimensionality of the pose space, standard techniques based on linear and Gaussian assumptions are not suitable. Instead, Monte Carlo methods based on a sampled representation of the involved likelihood functions yield to a promising research direction. In this thesis, we present a number of contributions to this topic based on a coarse-to-fine analysis scheme. The input data to all presented algorithms will be a 3D reconstruction of the scene, described by colored voxels, thus combining the information provided by all camera views into a unified data representation.

In a first stage, subjects are coarsely approximated by an ellipsoid and their centroids are estimated and tracked. A novel approach achieving real-time performance is presented based a surface sampling of the objects in the scene: the Sparse Sampling algorithm. In this filtering scheme, an independent tracker is assigned to every target and an exclusion mechanism is defined to avoid interference among targets. Finally, the obtained centroid positions are employed afterwards to initialize a specific pose estimation algorithm.

Two pose estimation algorithms are presented based on the seminal principle of the annealead particle filter technique. The first one is a low cost approach to marker-based human motion capture and, the second, is a markerless technique relying on likelihood functions computed directly on the 3D voxel representation. In both approaches, kinematic constrains are employed to avoid unfeasible poses. Although these algorithms provide satisfactory results when dealing with accurate input data, they tend to loose track when processing noisy measurements and occluded body parts.

Scalability of the structure of the human body is exploited to define two robust alternatives to analyze faulty data. In the first case, the Scalable Human Body Model-Annealed Particle Filter, is presented as filtering approach adding an extra annealing level to the classical annealed particle filter approach: the body hierarchy annealing loop. In this way, a progressive fitting is performed in a coarse-to-fine manner thus yielding to both more efficient and accurate results. Another alternative is presented employing a human body model hierarchy where different limbs are added progressively to the model. This allows detecting those parts that are occluded (for instance, by furniture) and disregard them into the likelihood evaluation step of the filtering scheme.

Finally, in order to evaluate all the systems proposed in this thesis, a new methodology is presented. Existing methods based on computing the mean and variance of the committed estimation error tend to produce biased figures when a subset of the human body is not tracked properly. We proposed two alternative metrics that avoids this situations and therefore allow a fairer comparison among algorithms.

Resum

La captura i seguiment dels moviments executats per persones s'ha convertit en un tema de recerca important, donades les aplicaciones que es poden desenvolupar emprant aquesta informació: reconeixement d'accions, interfícies home-màquina i aplicacions biomètriques. En aquesta tesi doctoral s'adreça el problema de l'extracció de la postura del cos humà en un entorn amb múltiples càmeres usant tècniques de Monte Carlo.

L'extracció de la configuració d'un model articulat del cos humà basant-se en la informació obtinguda de múltiples càmeres pot ser adreçada de forma eficient a través de la formulació Bayesiana clàssica basada en predicció i correcció. En aquest cas, donada l'alta dimensionalitat de l'espai d'estat associat a la postura, les tècniques estàndard basades en suposicions Gaussianes i en relacions lineals no són adequades. Les tècniques de Monte Carlo basades en representacions mostrejades de les funcions de versemblança són més indicades per aquest tipus de problemes. En aquesta tesi es presenten contribucions en aquest camp basant-nos en una esquema d'anàlisi *coarse-to-fine*. Les dades d'entrada a tots els algorismes d'aquesta tesi són una reconstrucció 3D de l'escena en forma de voxels colorejats. D'aquesta manera es combina la informació generada per múltiples càmeres en una representació unificada.

En una primera etapa, el cos humà s'aproxima per un el·lipsoide i el seu centroide és estimat i seguit, en funció del temps. L'algorisme *Sparse Sampling* es presenta com una solució robusta i d'execució en temps real per al problema del seguiment basant-se en un mostreig de la superfície dels objectes a seguir. En aquest esquema, cada objecte seguit té un filtre independent assignat i un mecanisme d'exclusió entre objectes permet evitar la interferència entre ells. Finalment, els centroides obtinguts seran emprats per a la inicialització del subseqüents sistemes d'estimació de postura.

Dos algorismes d'estimació de postura basats en la tècnica del filtre de partícules amb *annealing*. El primer algorisme és una solució de baix cost al seguiment dels moviments del cos utilitzant marcadors, mentre que el segon és una proposta sense marcadors basada en l'avaluació de les funcions de versemblança del filtre de partícules directament en l'espai 3D voxelitzat. En totes dues aproximacions es tenen en compte les restriccions físiques de les articulacions del cos humà per evitar postures impossibles. Tot i que aquestes propostes donen bons resultats quan les dades d'entrada són de prou qualitat, hi ha una tendència a divergir quan aquestes dades són sorolloses o hi ha parts del cos que no són visibles.

L'escalabilitat de l'estructura del cos humà és explotada per a definir dues alternatives robustes per l'anàlisi de dades deficients. En el primer cas, el *Scalable Human Body Model-Annealed Particle Filter* es presenta com a un filtrat on s'ha afegit una capa extra d'*annealing*: el llaç d'*annealing* associat a l'escalabilitat del cos. D'aquesta manera, s'aconsegueix una estimació progressiva de la postura obtenint una solució eficient i robusta. Una segona alternativa es basa en l'explotació d'una jerarquia de cos humà on les extremitats són afegides de forma progressiva al model. Això permet detectar aquelles extremitats que estan ocultes (per exemple, degut al mobiliari de l'escenari) i ignorar-les en l'avaluació de la versemblança en el procés de filtrat.

Finalment, per tal de comparar els sistemes proposats en aquesta tesi, es presenta

una nova metodologia d'avaluació. Els sistemes existents d'avaluació basats en calcular la mitja i la variança de l'error comès tendeixen a generar resultats esbiaixats quan alguna extremitat no és seguida correctament. S'han proposat dues mètriques alternatives que eviten aquest efecte i permeten una comparació justa entre les diverses tècniques proposades.

Agraïments

En primer lloc m'agradaria agrair als meus dos tutors en aquesta experiència doctoral, al Josep R. Casas i a la Montse Pardàs, pel seu encertat criteri, els seus consells i, molt particularment, el seu suport. També, dins del grup d'imatge, vull agrair especialment els comentaris del Ferran Marqués, al llarg d'aquests anys.

De tots aquells companys i amics que he fet a l'universitat, destacar les seves suggerències i constructives discusions, el seu suport i els bons moments passats junts.

Finalment, a tota la gent que, des de fora de l'universitat, m'han donat tot el seu recolzament, família i amics, i que, sense ells, aquest treball no hagués estat possible.

Cristian Canton Ferrer
Barcelona, Juny 2009

Contents

List of Figures

LIST OF FIGURES

Notation

Boldface upper-case letters denote matrices, boldface lower-case letters denote vectors and lower-case italics denote scalars.

$\mathbb{R}, \mathbb{N}$	The set of real and natural numbers, respectively.	
$\mathbf{X}^\top$	Transpose of matrix $\mathbf{X}$.	
$\mathbf{X}^{-1}$	Inverse of matrix $\mathbf{X}$.	
$\mathbf{x}^\top$	Transpose of vector $\mathbf{x}$.	
$\mathbf{x}_{\{x,y,z\}}$	The x, y or z component of vector $\mathbf{x}$.	
$[\mathbf{x}]_{\{x,y,z\}}$	The x, y or z component of vector $\mathbf{x}$ (employed when using several sub-indices).	
$\|\mathbf{x}\|$	Euclidean norm of vector $\mathbf{x}$: $\|\mathbf{x}\| = \sqrt{\mathbf{x}^\top \mathbf{x}}$.	
$\mathcal{V}$	A voxel.	
$\mathcal{V}_\mathbf{x}$	The 3D-position associated to the voxel $\mathcal{V}$.	($\mathbb{R}^3$)
$\mathcal{V}_{\{x,y,z\}}$	The x, y or z position of the center of the voxel, respectively.	($\mathbb{R}$)
$\boldsymbol{\mathcal{V}}$	A collection of binary voxels describing a 3D reconstruction of a scene.	
$\boldsymbol{\mathcal{V}}^\mathrm{C}$	A collection of color voxels describing a 3D reconstruction of a scene.	
$\boldsymbol{\mathcal{V}}^\mathrm{S}$	A collection of binary surface voxels.	
$\|\boldsymbol{\mathcal{V}}\|$	Number of non-zero voxels of the set $\boldsymbol{\mathcal{V}}$.	($\mathbb{N}$)
$[\boldsymbol{\mathcal{V}}]$	The physical limits of the voxel set $\boldsymbol{\mathcal{V}}$ (i.e. the room)	
$\boldsymbol{\mathcal{V}}(\mathbf{x})$	The content (typically, a label) stored at position $\mathbf{x}$ from set $\boldsymbol{\mathcal{V}}$	
$s_\mathcal{V}$	Size of the side of voxel $\mathcal{V}$.	($\mathbb{R}$)
N_C	Number of cameras.	($\mathbb{N}$)
$\mathcal{X}$	A generic state space.	($\mathbb{R}^N$)
$\mathrm{E}[\cdot]$	Expectation operator.	
f_R	Sequence frame-rate.	($\mathbb{N}$)
$O(\cdot)$	Landau's notation for the limiting complexity of an algorithm	

1 Introduction

AUTOMATIC human motion capture and tracking is a difficult problem which has been an important challenge and a very active research field within the computer vision community during the last decade. However, despite this great deal of attention and a noticeable progress, the general problem of automatically obtaining the defining parameters of the human body pose remains unsolved. With this thesis, we intend to contribute to the state of the art in specific aspects of this problem in the context of a multi-camera scenario. We shall particularly focus our efforts on exploiting the underlying hierarchical structure of the human body for pose estimation by designing algorithms robust to noisy and heavily corrupted data, which is a common (and sometimes disregarded) scenario. In this introductory chapter, we present the motivations of our research, an outline of the contributions and the overall structure of the thesis.

1.1 Motivation

Human kinematic modeling, motion tracking and analysis are difficult problems due to the underlying multimodal and high dimensional estimation problem involved. Despite the complexity of this problem, this topic has received a great deal of attention mainly fostered by the numerous applications that benefit from the extracted information. One type of applications are those where the extracted body model parameters are directly used, for example to interact with a virtual world [RMG+02], drive an animated avatar in a video game [VU05] or for character animation computer graphics [HLGB03]. Another class of applications use extracted parameters to classify and recognize people, gestures or motions [CGPV05], surveillance systems [HHD00], intelligent environments [CHI07], content based indexing of sports video footage [YS05] or advanced user interfaces such as sign language translation [DM06]. Finally, the motion parameters can be used for motion analysis in applications such as personalized sports training [CPF03], choreography [OCFT+08a], or clinical studies of motion disorders [DBT03].

Human motion capture has been usually posed as a statistical estimation problem where the temporal evolution of the defining parameters of the human body are computed based on a series of noisy observations gathered by a number of sensors surrounding the person under study. Relating these observations with the set of parameters to be estimated involves dealing with multimodal probability distributions in high dimensional spaces rendering classical linear methods unsuitable. This thesis is focused towards this

challenging field. In particular, this thesis addresses the problem of human motion capture when analyzing a 3D colored voxel reconstruction of the scene. The input is obtained after a data fusion process from all input sensors (in our case, color cameras).

Real scenarios may involve input data of low quality containing large missing parts and heavy noise. This situation has been customarily skipped assuming that input feds were well conditioned. Therefore, most of existing human motion techniques are prone to fail when facing an adverse or harsh scenario. This situation poses an extra challenge that brings technology closer to real world conditions thus leading to the ultimate motivation of this thesis: exploiting the underlying hierarchical structure of the human body towards designing robust human motion capture algorithms.

Initialization of the human motion capture system is a delicate problem that has been usually solved by manual intervention or by forcing the performer to adopt a predefined initial pose. These start-up protocols greatly limit the applicability of the system when intended to operate in real world situations. In this thesis, we avoided such problem by designing a robust multi-person tracker that estimates the position of the person of interest in the analysis scenario and delivers this information to be used as initialization of the forthcoming human motion capture system.

Another motivation that has driven the spirit of this thesis is fairness in the comparison and evaluation of the proposed systems. Nowadays, there is still a number of scientific contributions that evaluate the accuracy and precision of their algorithms on a qualitative basis and with a limited comparison with already existing state of the art techniques. Recently, there is a growing interest within the research community to develop standard evaluation datasets and a set of beforehand agreed metrics towards allowing fair comparisons among systems based on quantitative criteria. In this thesis we followed this aim by employing standard evaluation datasets (CLEAR [CLE07] and HumanEva [SB06]) and fairly comparing the obtained results with other state of the art algorithms, using objectively defined metrics.

1.2 Summary of Contributions

The contributions of this thesis are found in the framework of multi-camera human motion analysis and can be summarized as follows:

- **3D processing**. The first step before addressing human motion capture is to detect and track people inside the analyzed region. In most of existing techniques, multiple video streams are processed separately and information is then fused at feature level. We explored the complementary approach, information fusion at data level, that is to generate a data representation aggregating information from all video sources into a 3D colored voxel reconstruction of the analyzed region before the tracking process starts, thus getting rid of occlusion and perspective issues. In this work, 3D information will be considered as the default input to all presented algorithms.

- **Person tracking**. The tracking loop is analyzed using a block diagram where we carefully review a usually neglected step that is the creation and deletion of new tracks. Impact of this block onto the overall system performance is emphasized and discussed. Two techniques based on the Monte Carlo principle are introduced to perform multi-person tracking: Particle Filtering and Sparse Sampling, being the latter a major contribution of this thesis, partly due to its notable computational complexity reduction.

- **Marker-based pose estimation**. An annealed particle filtering algorithm to perform human motion tracking from a set of markers placed on the body exploiting redundancy among cameras by means of the introduced generalized symmetric epipolar distance. This system is intended as an economic solution to professional motion capture systems built on dedicated and expensive hardware [Vic].

- **Markerless pose estimation**. The complementary markerless approach using an annealed particle filtering is also presented as a novelty. 3D information generated using multiple camera video streams are employed as the input for this algorithm together with some new considerations and likelihood features in the tracking algorithm.

- **Scalable human body model based pose estimation**. The concept of scalable human body model together with annealed particle filtering led to the adaptive filtering solutions that are the main contribution of this thesis. These algorithms are able to adapt the human body model used during the analysis to the quality of the input data, thus becoming highly robust to noisy and corrupted data.

- **Performance metrics**. Already existing metrics for human motion capture performance evaluation have been reviewed and we proved that they might produce biased results under some circumstances. A set of evaluation metrics for human motion capture have been proposed avoiding the aforementioned problem.

A complete list of the contributions of this thesis, as well as the references to journals, book chapters, conference proceedings, contributions to projects and publications in the course of preparation have been compiled in the last chapter.

1.3 Thesis Organization

The remainder of the thesis is organized as follows. In next chapter, we state the problem that we want to address, analyzing the pipeline going from the input devices (cameras) to the extraction of human body parameters. Together in this chapter, a review of the theoretical background aspects required to develop the algorithms presented in this thesis is given. This includes a brief but descriptive survey on the techniques employed to generate the input data to our algorithms (background/foreground image segmentation and 3D voxel reconstruction). State space theory and Monte Carlo techniques, specifically particle filtering and simulated annealing, are reviewed in Chapter 3.

In Chapter 4, we present our contribution to the field of multi-person tracking as a previous step to human motion capture since locating the position of the person(s) under study within the working area allows avoiding manual initialization of the forthcoming processing modules. Chapter 6 is devoted to our contributions in both marker and markerless human motion capture based on the seminal concept of the annealed particle filter. The limitations of these algorithms when dealing with faulty data are the motivation for the techniques presented in Chapter 7, where scalability and hierarchy concepts are introduced into the design of an adaptive filtering technique able to automatically modify the human body model employed to analyze the input data depending on its quality.

A thorough evaluation of all involved modules (multi-person tracking and human motion capture) is presented using standard datasets (CLEAR and HumanEva-I, respectively) allowing a fair comparison with already existing state of the art methods. Specific metrics have been designed for human motion capture evaluation as explained in Chapter 5. An overall discussion on the performance of the presented systems is presented in Chapter 8 and, finally, Chapter 9 draws some conclusions and future extensions of this work.

2
Problem Statement

EXTRACTING the pose of the human body based on a set of observations gathered from multiple synchronized video feeds entails a number of challenges ranging from generating informative image features to design efficient and robust tracking systems. Mainstream research in the topic of human motion capture assumes that input data are error-free, and therefore reliable to be used as input to analysis algorithms. However, actual data are quite commonly defective thus requiring techniques able to cope and adapt to the particularities of these inputs. Within this thesis, several contributions to the state of the art in various aspects of human motion capture are presented.

In this chapter, we give an overview of the problems tackled in this thesis and a review of the required background concepts. The problems and ideas exposed here are developed in the following chapters.

2.1 Thesis roadmap

Human motion capture using a set of multiple cameras can be approached from several perspectives. In this thesis we explore some of these approaches in order to offer a wide view of the problem and to be able to make a comparison among them. Let us review the flowchart depicted in Figure 2.1, that will be described later in §2.2. The first alternative is found when deciding how to process data provided by multiple cameras. Two options are available:

- First, by separately analyzing every image to extract some meaningful features and afterwards combine all them in what is denoted as a feature fusion approach. This approach is probably the most usual research path, and notable contributions have been made in the last years [GD96, KM00, DF01, CPF03, DR05].

- Second, by combining all information provided by every camera in a compact data representation and subsequently extracting features over this new dataset in what is known as a data fusion approach. Within the data fusion perspective, there is a recent growing interest [CKBH00, KG06] partly fueled by the improvement in the computers' capacity usually required to generate the fused dataset and by the ability of these methods to better handle perspective and occlusion issues (in comparison with feature fusion methods).

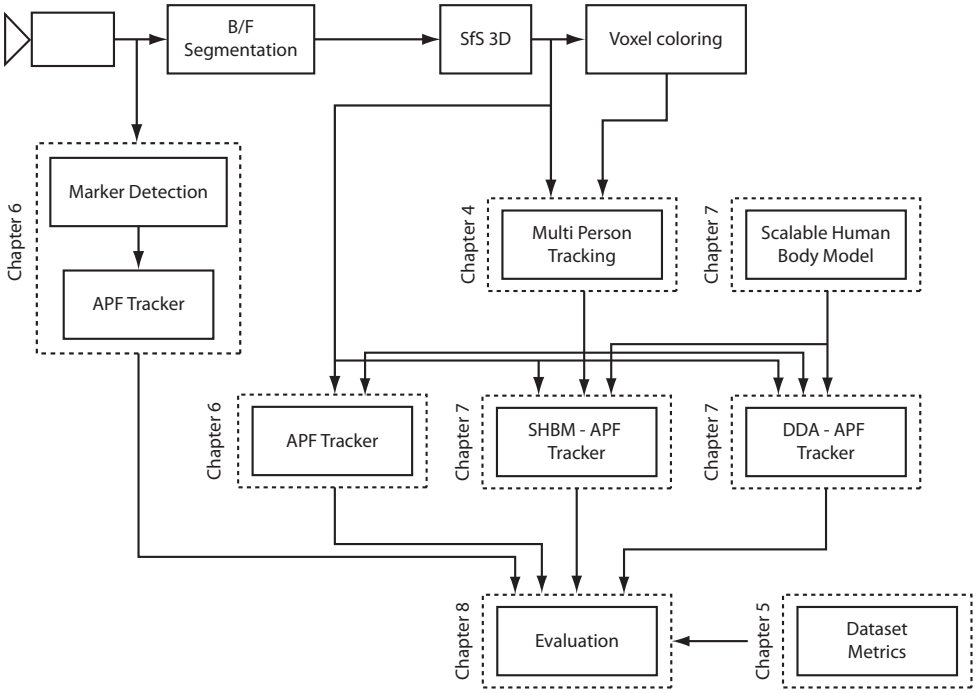


Figure 2.1: *Thesis roadmap.*

The data fusion technique employed in this thesis is the generation of a 3D reconstruction of the scene encoded with voxels. Although not being a problem, algorithms based on feature or data fusion will have to work with these data and therefore deal with their particularities. In this thesis, we present an algorithm based on feature fusion in Chapter 6 and several techniques working with data fusion in both Chapter 6 and 7.

The second alternative is found in the methodology employed to extract the pose of the subject under study. Two options are found: marker based, where a set of distinguishable makers are placed near the human body junctions, and markerless, where no artificial aid is provided. Obviously, the marker based approach will produce better results than the markerless one at a cost of being intrusive and limiting its applicability to some very specific goals (usually related with professional motion capture for cinema or medical applications). In the marker based approach, derived issues are found in managing the auto-occlusions of the human body and the problem of efficiently relating the detected markers in several views through projective geometry. They are tackled in Chapter 6. Markerless tracking is indeed a more complex problem and involves designing robust algorithms able to fit a highly articulated structure to the input data. Monte Carlo methods are the selected technique in this work to build the presented algorithms. A number of problems are derived such as multimodal likelihood function analysis that implies defining efficient ways to relate the observations with the defined state space and they are widely covered in Chapter 6.

Initialization of human motion capture systems has been usually performed by hand. Instead, we propose a multi-person tracking system based on the 3D voxel reconstruction

as the input of the system. Involved challenges include dealing with a noisy input dataset subject to light variations and shadows and keeping track of an unknown number of targets in a cluttered environment. Real-time performance is highly desirable. All these problems are handled using a novel Monte Carlo approach in Chapter 4.

Corrupted input data or data containing significant misses is a common scenario, though being commonly skipped in the literature. Indeed, most of the state of the art algorithms are prone to fail when facing corrupted data and this poses an interesting challenge. Monte Carlo theory, and specially particle filtering techniques, are adapted to deal with this type of data applying the newly introduced concept of hierarchical and scalable analysis models. This is the core of Chapter 7.

The final problem addressed in this thesis is to fairly compare all presented human motion capture methods. In this way, we designed a set of metrics overcoming some biasing problems that flawed the existing ones. These metrics presented in Chapter 5 are employed in Chapter 8 for an overall discussion.

2.2 Starting point: input data

Data fed to algorithm modules displayed in Figure 2.1 is a determining factor. Although this thesis is not devoted to background/foreground segmentation or 3D reconstructions, some indications are given to establish a starting point. In this way, the reader will understand the challenges associated to obtain a pose from these input data.

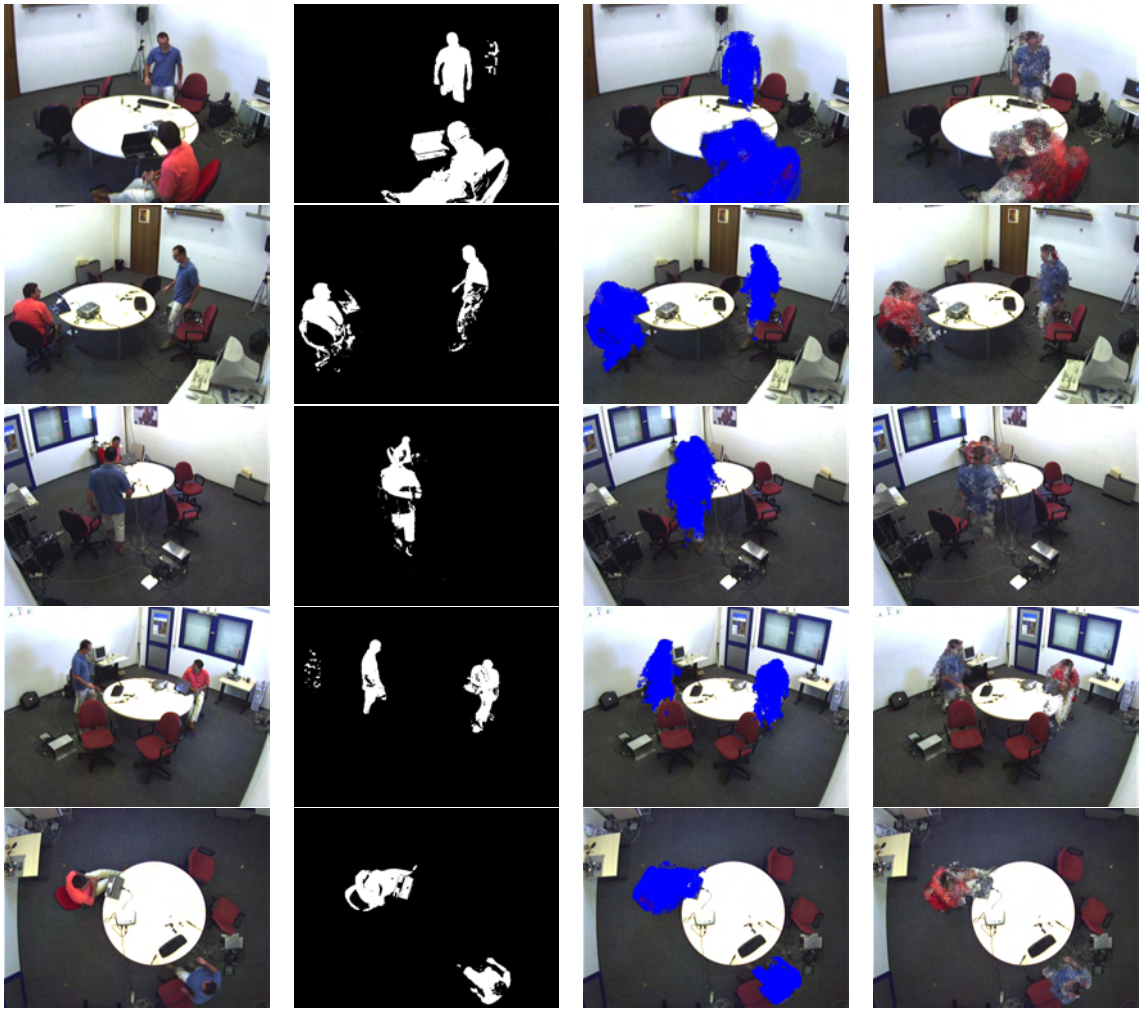
2.2.1 Multi-camera scenario

The continuous grow of computer's capacity, speed and storage has allowed to explore techniques involving more than one data stream with a quasi real-time operation. In our particular case, we deal with multiple synchronized video streams as input to our human motion capture algorithms. The cameras are placed in such a way that the area under analysis is covered from several perspectives (see [OM02] for a proposal on optimal camera planning). In our particular scenario, images will be similar to those shown in Figure 2.2(a) belonging to the CLEAR [CLE07] database. Data from the HumanEva-I [SB06] has been also thoroughly employed (see Figure 6.15 and Chapter 5 for a sample).

Epipolar geometry associated to multi-camera setups [HZ04] is exploited towards establishing correspondences across views in the feature fusion process and allows generating a 3D reconstruction of the space in the data fusion approach.

2.2.2 Pinhole Camera Model

Obtaining two-dimensional coordinates (pixel positions) of an image from a three dimensional magnitude (a 3D location) is a process where a dimension is lost. Formally, projection can be seen as a many-to-one morphism $\psi : \mathbb{R}^3 \rightarrow \mathbb{N}^2$ that transforms 3D Euclidean coordinates in the world reference frame into 2D coordinates in the camera reference frame. The usual mathematical way to model this process passes through projective geometry as an efficient description of the image formation process [FL01]. Essentially, a camera is regarded as a projective device where an image is the result of the central



(a) Raw images (b) Segmented images (c) Projection of the 3D reconstruction (d) 3D colored reconstruction

Figure 2.2: *Multi-camera input data using a voxel size of* 2 *cm.*

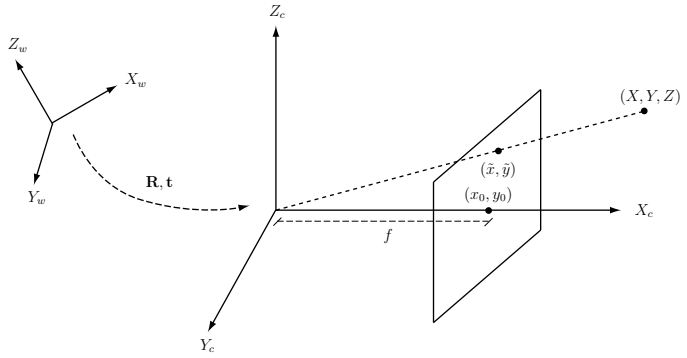


Figure 2.3: *Pinhole projection model. A point (X, Y, Z) in the real world coordinate system (X_w, Y_w, Z_w) is first referred to the camera coordinate system (X_c, Y_c, Z_c) and then projected into the image plane thus resulting in the $(\tilde{x}, \tilde{y})$ pixel coordinates. Focal length is noted as f.*

projection of 3D world points onto the image plane. In a simple model, the camera center is behind the image plane, and 3D points are mapped to 2D where the line joining the camera center and the 3D points meets with the image plane. This model, which is called the *pinhole camera model* [HZ04], is one of the most commonly employed models used with CCD cameras. Projective effects due to vanishing points can be easily modeled and understood if we take into consideration projective coordinate systems. Many authors take advantage from projective geometry and homogeneous coordinates when addressing computer vision problems [HZ04].

Projection operation can be fully described in homogeneous coordinates by the linear application $\mathbf{P} : \mathbb{P}^3 \rightarrow \mathbb{P}^2$ denoted as the projection matrix. So,

$$\mathbf{x} = \mathbf{P}\mathbf{X}, \qquad \mathbf{x} \in \mathbb{P}^2, \qquad \mathbf{X} \in \mathbb{P}^3. \tag{2.1}$$

It must be noted that projection is essentially a non-linear operation when defined by the application $\psi : \mathbb{R}^3 \rightarrow \mathbb{N}^2$. In fact, when adopting the pinhole camera model and the associated projective geometry model, the relation between the image coordinates $\tilde{\mathbf{x}} = [\tilde{x}\ \tilde{y}]^\top \in \mathbb{N}^2$ and the projected coordinates $\mathbf{x} = [x\ y\ z]^\top \in \mathbb{P}^2$ is stated as:

$$\tilde{x} = \left\lfloor \frac{x}{z} \right\rfloor, \qquad \tilde{y} = \left\lfloor \frac{y}{z} \right\rfloor. \tag{2.2}$$

From Figure 2.3, we can express the projection model in homogeneous coordinates in Eq.2.1 as:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.3}$$

where f stands for the focal length of the camera. Non-square pixels in CCD cameras and decentered image planes w.r.t. the intersection of the optical axis with the image

plane, force us to consider more general projective models:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \underbrace{\left(\begin{array}{ccc|c} fm_x & 0 & x_0 & 0 \\ 0 & fm_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right)}_{K} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.4}$$

where m_x and m_y are the scaling factors of the focal length in each dimension, and x_0 and y_0 are offsets in each dimension. The matrix containing all the information regarding the projective device (i.e. the camera) is usually denoted as the intrinsic parameters matrix $\mathbf{K}$.

Usually, the coordinate system of the real world (X_w, Y_w, Z_w) does not coincide with the coordinate system associated with the camera (X_c, Y_c, Z_c) thus an affine transformation relating this two systems is required:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \underbrace{\begin{pmatrix} fm_x & 0 & x_0 \\ 0 & fm_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} [\mathbf{R}|\mathbf{t}]}_{\mathbf{P}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.5}$$

where $\mathbf{R}$ and $\mathbf{t}$ are the 3×3 rotation matrix and 3×1 translation vector of the real world referred to the camera coordinate system. This rotation and translation are denoted as the extrinsic parameters of the camera. The projection matrix can be written more compactly as $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$.

Distortion introduced by lenses in the camera cannot be disregarded in the image formation process. Many authors [Zha02, HZ04, Gar04] agree that the most noticeable distortion effect in real cameras is the radial distortion, considering negligible decentering and other effects introduced by tangential distortion. The radial distortion model is expressed by the following equation:

$$\frac{r}{r_d} = \frac{\tilde{x}_d - x_0}{\tilde{x} - x_0} = \frac{\tilde{y}_d - y_0}{\tilde{y} - y_0}, \tag{2.6}$$

where $(\tilde{x}_d, \tilde{y}_d)$ are the distorted image point coordinates. The truncated Taylor expansion of Eq.2.6 w.r.t. variable r is $1 + k_1 r^2 + k_2 r^2$ thus the distortion process can be defined with the coefficients k_1 and k_2 solely. The pixel coordinates of the distorted image can be obtained as:

$$\tilde{x}_d = x_0 + L(r)(\tilde{x} - x_0), \tag{2.7}$$

$$\tilde{y}_d = y_0 + L(r)(\tilde{y} - y_0), \tag{2.8}$$

$$r = \sqrt{\left(\frac{\tilde{x} - x_0}{fm_x}\right)^2 + \left(\frac{\tilde{y} - y_0}{fm_y}\right)^2} \tag{2.9}$$

where r is the radius of distortion and $L(r) = 1 + k_1 r^2 + k_2 r^4$.

Finally, obtaining the 11 defining parameters of a camera is achieved by the calibration process. The reader is referred to [Bou04, Gar04] for more details on this process.

2.2.3 Background/Foreground segmentation

Segmenting an image is understood as the process where every pixel of it is labelled as belonging to zones or classes of the image. Background segmentation is a sub-type of general segmentation, where a binary classification is performed distinguishing between foreground (pixels that belong to the object) and background (the rest). Background segmentation is a very common pre-processing step in many image processing algorithms discarding the non-relevant information of the image and simplifying tracking tasks. Although there are a number of techniques for background/foreground segmentation we selected the family of those based on statistically modeling both the foreground and background as our starting point. For more information on these techniques, see [ZL00, Pic04, Lan07].

Most of background segmentation algorithms rely on building a statistical model of the background. In most cases, this implies capturing a training sequence with an empty scene and the impossibility to move the cameras during the recording process. Moreover, the background model may change along time due to illumination variations, hence the associated statistical model should be updated. Once this model is built up, images that we want to segment are compared with it and every pixel is labelled as belonging to the background model or being an exception to it, then being labelled as foreground. Although the algorithms presented in this thesis heavily rely on a good segmentation to ensure a proper operation, we have not explicitly conducted significant research on this topic relying on previously developed techniques. Basically, the background/foreground technique we have applied is that of Stauffer and Grimson [SG99] with some adjustments for real-time performance and shadow elimination introduced by Landabaso *et al.* [LP05]. For further details, the reader is referred to [Lan07]. Some examples of the technique employed in this thesis are depicted in Figure 2.2(b).

2.2.4 3D Data generation

Obtaining a 3D reconstruction of the analyzed scene from multiple images taken from different perspectives is itself a complex problem that has been actively researched in the recent years. The applications based on this information are numerous ranging from cinema applications and avatar animation [VU05], HCIs [CHI07] and human motion capture [Che03]. Although there are a number of available techniques to derive 3D information from a set of images [KS00, FK02, IS03, CS06], shape from silhouette [Che03] is among the most popular due to its conceptually simple methodology and fast execution time.

Shape from Silhouette

Let us consider a 3D space (i.e. a room) and a regular partition of this space into cubes of equal size denoted as *voxels* (in a volumetric analogy with the word *pixel*). For the sake of notation clarity, we will denote $\mathcal{V}$ as a voxel of this grid, $\mathcal{V}_{\mathbf{x}}$ as the discrete coordinates of voxel $\mathcal{V}$ and, more specifically, $\mathcal{V}_{\{x,y,z\}}$ to its x, y or z coordinates, respectively. Given a scene with several foreground objects, we would like to assess the occupancy or emptiness of every voxel and this goal can be achieved through the shape from silhou-

ette algorithm [CKBH00, Lan07]. The set containing all voxels labelled as belonging to a foreground object will be denoted as $\boldsymbol{\mathcal{V}}$ and its cardinality as $|\boldsymbol{\mathcal{V}}|$. As a final notation remark, we denoted $[\boldsymbol{\mathcal{V}}]$ as the physical limits of the analysis space, i.e. the size of the room. This notation will be thoroughly employed in Chapters 4 and 6.

Shape from silhouette methods are based on testing the occupancy of every voxel $\mathcal{V}$ in the analyzed space. In this work, the occupancy testing of $\mathcal{V}$ is achieved by projecting its center onto every camera image plane through projection operation described by Eq.2.1 and checking whether this pixel has been classified as foreground or background. Once this foreground test has been preformed for all N_{C} views, we may assess the occupancy of a given voxel. Ideally, assuming a perfect segmentation, the projections of a occupied voxel should fall into a foreground region for all N_{C} views. However, some tolerance is permitted to cope with faulty segmentations. Although this shape from silhouette is simple in comparison with more sophisticated schemes such as the SPOT algorithm [Che03] or the shape from inconsistent silhouette [LPC08], obtained results are of sufficient quality for our purposes (see an example in Figure 2.2(c)).

The main drawback of voxel based procedures is the computational cost of generating such reconstruction [Che03]. Each voxel has to projected into the image plane of each camera, leading to one matrix multiplication per voxel and per camera. Most implementations speed up this process by using an octree representation to compute the result from coarser to finer resolutions [Sze93] or exploiting hardware acceleration [HLGB03]. Usually, for a fixed 3D region to be reconstructed, the voxel size $s_{\mathcal{V}}$ will be the parameter determining the computational complexity of the algorithm: $O(s_{\mathcal{V}}^3)$. Moreover, the smaller the voxel size, the more precise the 3D reconstruction thus posing a tradeoff between speed and accuracy.

Voxel coloring

Color information is a useful cue to many tasks addressed in this thesis such as to distinguish among different subjects in the scene in a tracking problem. Obtaining color information from a RGB image is trivial but when dealing with a 3D voxel reconstruction of the scene, a process to assign a color to every voxel should be defined. There is a number of techniques that tackle this problem based on photoconsistency [SD99] or surface estimations [WC04]. In our particular work, we followed an approach close to [Bǒ3] leading to satisfactory results with an affordable complexity.

In the context of this thesis, fast performance was required thus avoiding methods based on computationally expensive algorithms. Two approaches have been devised:

- First, we projected the center of every voxel of the obtained volume $\boldsymbol{\mathcal{V}}$ onto every camera and averaged the colors found in these locations. This approach turned out to produce biased color assignations as shown in Figure 2.4(b). This effect was the consequence of not taking into account visibility of every voxel with respect to all cameras.

- Visibility of a voxel with respect to a given camera N_i can be assessed by first tracing the line joining the position of this voxel $\mathcal{V}_{\mathbf{x}}$ with the optical center of this

(a) Original Image (b) Color averaging (c) Color averaging with visibility consistency

Figure 2.4: *Voxel coloring examples showing the difference of color estimation when averaging the colors at the projection of every voxel in every camera (case (b)). When taking into account visibility, colors are not biased as shown in (c).*

camera. This line is drawn in the discrete space using Bresenham's line drawing algorithm [Bre65] in 3D.

The 3D positions within this line are tested in the volume $\boldsymbol{\mathcal{V}}$, and voxel $\mathcal{V}$ is considered to have direct vision with N_i if all elements of this line are empty. Finally, the average colors corresponding to cameras with direct vision of voxel $\mathcal{V}$ is assigned as the color of $\mathcal{V}$. Moreover, this assignation allows distinguishing between interior voxels, that is those with no direct vision with any camera, and surface voxels, the rest. The set of surface voxels will be denoted as $\boldsymbol{\mathcal{V}}^{\mathrm{S}}$ and the set of colored voxels as $\boldsymbol{\mathcal{V}}^{\mathrm{C}}$. Note that, every voxel in set $\boldsymbol{\mathcal{V}}^{\mathrm{C}}$ will have assigned three components (R, G and B) and that the cardinality of sets $\boldsymbol{\mathcal{V}}^{\mathrm{C}}$ and $\boldsymbol{\mathcal{V}}^{\mathrm{S}}$ will be the same. An example of this technique is shown in Figures 2.4(c) and 2.2(d).

2.3 Conclusions

This chapter has introduced the structure of this PhD thesis, presenting the several modules involved in the process that goes from the capture of multiple images to the extraction of the pose of an individual. There are a number of options to be considered in the design of such human motion capture algorithms, each of them entailing a number of challenges to be considered. These problems will be tackled in the following chapters.

Generation of the input data that will be fed to the processing chain is presented. These data are the 3D reconstruction of the analyzed space, obtained by means of a background/foreground segmentation followed by a shape-from-silhouette algorithm. Some brief remarks are given on the camera model and the algorithms involved in the generation of this 3D voxelized representation of the space.

3

Particle Filtering Background

HUMAN MOTION CAPTURE is a problem that has been usually addressed through an estimation and tracking perspective. Estimating the temporal evolution of the defining parameters of the human body (typically, the angles at the joints) involves dealing with high dimensional state spaces that are usually non-convex. This type of problems are well handled by Monte Carlo methods that obtain statistical measures through an efficient stochastic sampling of the involved state space. When extending the Monte Carlo theory to the filtering problem associated to human motion capture, we find the *particle filtering* techniques [AMGC02], that are the seminal concept applied to countless applications in the field of signal processing: digital communications [DKZ+03], multi-object tracking [Lan06], multimodal data fusion [CFSC+08], head orientation estimation [CFSC+07b], etc.

All through this thesis, particle filtering will be the basic technique where the presented algorithms and systems are built on, therefore we considered opportune a brief explanatory review in order to introduce the required concepts. Moreover, we tried to focus our explanations on human motion capture. In this way, the reader will acquire the required background to fully grasp the ideas introduced in Chapter 4 and 6. The concepts introduced in this chapter will be specially useful when reaching Chapter 7 where particle filtering theory is extended to variable dimension state spaces, in what is a novelty presented in this thesis.

First, an introduction to state space problems within a Bayesian framework is presented and particle filtering is introduced as an efficient technique to deal with multimodal high dimensional state spaces. The four steps of this algorithm are reviewed: resampling, propagation, evaluation and estimation. Afterwards, *simulated annealing* is presented as an efficient technique to deal with minimization problems on multimodal functions. Its combination with particle filtering leads to the *annealed particle filter* that has been found particularly useful for human motion capture. Regarding this last filtering technique, an issue that has not been previously reported in the literature is noticed and analyzed: the over-annealing effect, that renders the annealed particle filtering technique inefficient under certain circumstances. Examples on all these concepts are given for the sake of understandability.

3.1 Bayesian Framework and Monte-Carlo Filtering

3.1.1 Bayesian Framework

Let us define the problem of tracking as the estimation of a time varying state vector $\mathbf{y}_t$, $t \in \mathbb{N}$, belonging to a given state space $\mathcal{X} \subset \mathbb{R}^D$. The evolution of the state vector is modelled as a discrete process:

$$\mathbf{y}_t = f_t\left(\mathbf{y}_{t-1}, \mathbf{v}_{t-1}\right), \tag{3.1}$$

where $f_t(\cdot)$ is a possibly non-linear function describing the evolution of the model at time t, and $\mathbf{v}_t$ is the process noise. The model process function $f_t(\cdot)$ is unknown, and can not be observed directly. The objective of tracking is to recursively estimate $\mathbf{y}_t$ from a series of observations $\mathbf{z}_t \in \mathbb{R}^M$ derived from the measurement equation:

$$\mathbf{z}_t = h_t\left(\mathbf{y}_t, \mathbf{w}_t\right), \tag{3.2}$$

where $h_t(\cdot)$ is a possible non-linear function modelling the relation between the hidden state $\mathbf{y}_t$ and its measurable magnitude $\mathbf{z}_t$, and $\mathbf{w}_t$ is the measurement noise. Noise components $\mathbf{v}_t$ and $\mathbf{w}_t$ are assumed to be independent stochastic processes with a given distribution. Note that state space dimension, D, and dimension of the measure space, M, do not necessarily coincide. In the case of human motion capture, the state space will be selected as the pose defining parameters of the human body (typically, the angles at the joints) and typically rises up to $D \sim 25$ whereas measurement space for this task will be the 3D voxel reconstruction of the space.

From a Bayesian perspective, the estimation and tracking problem is to recursively estimate a certain degree of belief in the state vector $\mathbf{y}_t$ at time t, given the set of all available measurements $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t\}$. Thus, it is required to calculate the *posterior* probability density function $p(\mathbf{y}_t|\mathbf{z}_{1:t})$, and this can be done recursively in two steps, namely prediction and update [WH97]. Assuming that Eq.3.1 involves a first order Markov process (that is $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{z}_{1:t-1}) = p(\mathbf{y}_t|\mathbf{y}_{t-1})$), the prediction step obtains the prior *pdf* by means of the Chapman-Kolmogorov integral:

$$p\left(\mathbf{y}_t|\mathbf{z}_{1:t-1}\right) = \int p\left(\mathbf{y}_t|\mathbf{y}_{t-1}\right) p\left(\mathbf{y}_{t-1}|\mathbf{z}_{1:t-1}\right) d\mathbf{y}_{t-1}, \tag{3.3}$$

with $p\left(\mathbf{y}_{t-1}|\mathbf{z}_{1:t-1}\right)$ known from the previous iteration and $p\left(\mathbf{y}_t|\mathbf{y}_{t-1}\right)$ determined by Eq.3.1. When a measurement $\mathbf{z}_t$ becomes available, it may be used to update the prior *pdf* via Bayes' rule:

$$p\left(\mathbf{y}_t|\mathbf{z}_{1:t}\right) = \frac{p\left(\mathbf{z}_t|\mathbf{y}_t\right) p\left(\mathbf{y}_t|\mathbf{z}_{1:t-1}\right)}{p\left(\mathbf{z}_t|\mathbf{z}_{1:t-1}\right)}, \tag{3.4}$$

where the normalizing constant

$$p\left(\mathbf{z}_t|\mathbf{z}_{1:t-1}\right) = \int p\left(\mathbf{z}_t|\mathbf{y}_t\right) p\left(\mathbf{y}_t|\mathbf{z}_{1:t-1}\right) d\mathbf{y}_t, \tag{3.5}$$

depends on the likelihood function $p\left(\mathbf{z}_t|\mathbf{y}_t\right)$ derived from Eq.3.2. Combining Eqs.3.3 and 3.4, a more descriptive formulation of the posterior can be obtained:

$$\underbrace{p\left(\mathbf{y}_t|\mathbf{z}_{1:t}\right)}_{\text{Posterior}} = \kappa \; \underbrace{p\left(\mathbf{z}_t|\mathbf{y}_t\right)}_{\text{Likelihood}} \int \underbrace{p\left(\mathbf{y}_t|\mathbf{y}_{t-1}\right)}_{\text{Motion prior}} \; \underbrace{p\left(\mathbf{y}_{t-1}|\mathbf{z}_{1:t-1}\right)}_{\text{Previous posterior}} \; d\mathbf{y}_{t-1}, \tag{3.6}$$

with the normalization constant $\kappa = p\left(\mathbf{z}_t|\mathbf{z}_{1:t-1}\right)$. From this equation, it may be appreciated that the necessary elements that must be provided to a tracking system in order to follow the described Bayesian framework are the dynamic model and the likelihood function. At initialization ($t = 0$), no observation is available so the initial distribution of the posterior is set to be the initial distribution of the state vector, $p\left(\mathbf{y}_0|\mathbf{z}_0\right) \equiv p\left(\mathbf{y}_0\right)$, also known as the *prior* and, typically, this initial prior is set to be a wide distribution.

The posterior *pdf* $p\left(\mathbf{y}_t|\mathbf{z}_{1:t}\right)$ in Eq.3.4 may be peaky and far from being convex. Hence, it can not be computed analytically unless linear-Gaussian models are adopted. In this case, the Kalman filter provides the optimal solution [WB95]. Otherwise, sub-optimal approaches like the extended Kalman filter or the unscented Kalman filter tackle this problem using algebraic methods and linearization approaches [Ord05]. However, these approaches tend to collapse when facing multimodal distributions involving a high dimensionality of the state space. In the field of human motion capture, some early approaches were reported to use the Kalman filter [Mik03] together with some improvements to avoid the high dimensionality of the problem.

3.1.2 Monte Carlo approach

Particle Filtering (PF) algorithms are methods based on point mass (or "particle") representations of probability densities. These techniques are employed to tackle estimation and tracking problems where the involved variables do not hold Gaussianity uncertainty models and linear dynamics. PF belongs to the more general class of sequential Monte Carlo methods [DFG01] that are computational algorithms that rely on repeated random sampling to compute their results. Several names have been given to denote the PF algorithm such as sequential Monte Carlo, bootstrap filtering, CONDENSATION [IB98] or survival of the fittest. The reader is referred to [AMGC02, Che05] for a comprehensive review of all these techniques.

PF expresses the belief about the system at time t by approximating the posterior probability distribution $p(\mathbf{y}_t|\mathbf{z}_{1:t})$. This distribution is represented by a *weighted particle set* $\{(\mathbf{y}_t^j, \pi_t^j)\}_{j=1}^{N_\mathrm{p}}$, which can be interpreted as a sum of Dirac functions centered at $\mathbf{y}_t^j$ with their associated real, non-negative weights π_t^j:

$$p\left(\mathbf{y}_t|\mathbf{z}_{1:t}\right) \approx \sum_{j=1}^{N_\mathrm{p}} \pi_t^j \delta(\mathbf{y}_t - \mathbf{y}_t^j). \tag{3.7}$$

In order to ensure convergence, weights must fulfill the normalization condition $\sum_j \pi_t^j = 1$. Each particle represents a possible instance of the state space $\mathcal{X}$ hence, PF can be regarded as a tracking scheme where multiple hypothesis are propagated along time. The approximation from Eq.3.7 approaches the true posterior as $N_\mathrm{p} \to \infty$.

Weights are computed by means of an importance sampling procedure where a sampling distribution $q(\cdot)$ is employed; taking some widely accepted assumptions [AMGC02], it leads to the following recursive expression to compute the weights:

$$\pi_t^j \propto \pi_{t-1}^j \frac{p(\mathbf{z}_t|\mathbf{y}_t^j)p(\mathbf{y}_t^j|\mathbf{y}_{t-1}^j)}{q(\mathbf{y}_t^j|\mathbf{y}_{t-1}^j, \mathbf{z}_t)}. \tag{3.8}$$

Algorithm 1: Generic Sample Importance Resampling Particle Filter

repeat

▷ Resampling (according to §3.1.2.1)

Draw $\tilde{\mathbf{y}}_t^j \sim p(\mathbf{y}_{t-1}^j|\mathbf{z}_{t-1})$

$\pi_t^j = N_{\mathrm{p}}^{-1}$

▷ Propagation (according to §3.1.2.2)

$\mathbf{y}_t^j = \tilde{\mathbf{y}}_t^j + \mathcal{N}(\mathbf{0}, \mathbf{P})$

▷ Evaluation (according to §3.1.2.3)

$\pi_t^j = w(\mathbf{z}_t|\mathbf{y}_t^j)$

$\pi_t^j = \frac{\pi_t^j}{\sum_{i=1}^{N_{\mathrm{p}}} \pi_t^i}$

▷ Estimation (according to §3.1.2.4)

$t = t + 1$

until end

It is often convenient to choose the importance density to be the prior, $q(\mathbf{y}_t^j|\mathbf{y}_{t-1}^j, \mathbf{z}_t) = p(\mathbf{y}_t|\mathbf{y}_{t-1}^j)$ yielding to $\pi_t^j = \pi_{t-1}^j p(\mathbf{z}_t|\mathbf{y}_t^j)$. A common approach to drive particles along time is the Sampling Importance Re-sampling (SIR) strategy [GSS93], where $\pi_{t-1}^j = N_{\mathrm{p}}^{-1}$ and, therefore, weight computation is reduced to:

$$\pi_t^j \propto p(\mathbf{z}_t|\mathbf{y}_t^j). \tag{3.9}$$

Other techniques (in other domains) aim at constructing efficient sampling procedures through Markov Chain Monte Carlo methods [GRS95] or exploiting independence among variables in the state space through Rao-Blackwellization [DFG01]. Although there is a large number of methods to compute the associated particle weights[1], the presented procedure is the most largely accepted and the one that better suited the type of problems tackled in this thesis.

Four steps are involved in the SIR PF operation: resampling, propagation, evaluation and estimation. This algorithm is described in Algorithm 1 and depicted in Figure 3.1. In the following subsection, we review each step of the filtering loop.

3.1.2.1 Resampling

A common problem with the PF is the degeneracy phenomenon, where after a few iterations, all but one particle will have negligible weight. This effect implies that a large computational effort is devoted to updating particles whose contribution to the approximation of $p(\mathbf{y}_t|\mathbf{z}_{1:t})$ is almost null. A measure on the degeneracy of the particle set is the *effective sample size* proposed in [LC98] as

$$\widehat{N_{\mathrm{eff}}} = \left(\sum_{j=1}^{N_{\mathrm{p}}} (\pi_t^j)^2 \right)^{-1}. \tag{3.10}$$

[1] The reader is referred to [Che05] for a comprehensive review of most of these methods.

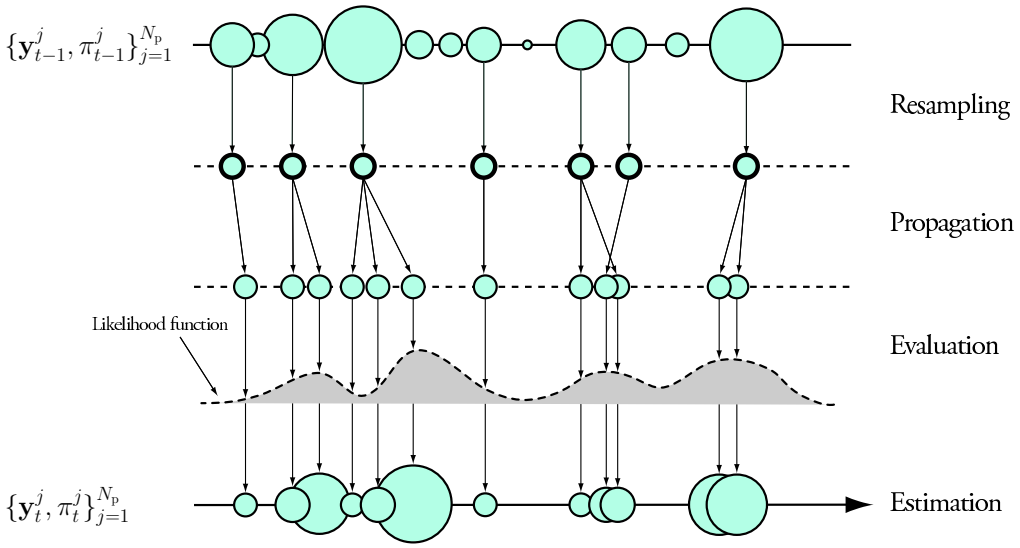


Figure 3.1: *Particle filtering scheme. Particles at time $t-1$, $\{(\mathbf{y}_{t-1}^j, \pi_{t-1}^j)\}_{j=1}^{N_p}$, are updated to $\{(\mathbf{y}_t^j, \pi_t^j)\}_{j=1}^{N_p}$ after completing the loop of resampling, propagation, evaluation and, eventually, estimation.*

It can be seen that small values of $\widehat{N_\mathrm{eff}}$ indicates a severe degeneracy. Another measure that evaluates the efficiency of a PF system is the *particle survival rate* proposed by [MI00]:

$$\alpha = \frac{\widehat{N_\mathrm{eff}}}{N_\mathrm{p}}. \tag{3.11}$$

A strategy to circumvent the particle degeneracy problem is re-sampling with replacement when $\widehat{N_\mathrm{eff}}$ is below a threshold, that is to dismiss the particles with lower weights and proportionally replicate those with higher ones. The underlying idea behind resampling is to concentrate particles in the regions of the state space where there is a higher likelihood to find a correct configuration of the estimated variables. However, the SIR implementation performs the resampling process at each time instant hence these two measures may appear unnecessary but still useful to assess the performance of the system[2].

There are a number of algorithms implementing the resampling procedure such as the systematic, stratified or residual resampling algorithms [AMGC02]. Systematic resampling [Kit96] algorithm is the scheme preferred in this work since it is simple to implement and achieves a linear complexity in the number of particles, $O(N_\mathrm{p})$. Its operation is described in Algorithm 2 and shown in Figure 3.1.2.2. The resulting set of particles is a random sample from the discrete approximation of the posterior, which explains that all weights are reset to N_p^{-1}. Once the algorithm has been introduced, $\widehat{N_\mathrm{eff}}$ can be intuitively regarded as the number of particles which would survive the resampling operation and α the fraction of "efficient" particles.

[2]Annealing strategies presented further in this chapter will make use of this measure.

Algorithm 2: Systematic Resampling Algorithm

$c_1 = \pi_t^1$
for $j = 2$ **to** N_p **do**
  $c_j = c_{j-1} + \pi_t^j$
end
Draw a starting point $u_1 \sim \mathbf{U}[0, N_{\mathrm{p}}^{-1}]$
$j = 1$
for $i = 1$ **to** N_p **do**
  $u_i = u_1 + N_{\mathrm{p}}^{-1}(i-1)$
  while $u_i > c_j$ **do**
    $j = j + 1$
  end
  $\{\mathbf{y}_t^i, \pi_t^i\} = \{\mathbf{y}_t^j, N_{\mathrm{p}}^{-1}\}$
end

Although the resampling step reduces the effects of the degeneracy problem, it introduces a sample impoverishment effect. Particles that have high weights are statistically selected many times leading to a loss of diversity among new particles. However, this problem is harmless when the process noise introduced in the propagation step allows differentiating particles from each other.

3.1.2.2 Propagation

Propagation of particles after resampling is made using temporal dynamics encoded in Eq.3.1. Typically, it is accomplished by:

$$\mathbf{y}_t^j = f(\tilde{\mathbf{y}}_{t-1}^j) + \mathcal{N}(\mathbf{0}, \mathbf{P}), \tag{3.12}$$

where $f(\cdot)$ represents the dynamical model and $\mathcal{N}$ a Gaussian multi-variate drift with zero mean and a diagonal co-variance matrix $\mathbf{P}$. When particularizing the propagation step in PF to our field of interest, that is motion tracking, some remarks can be drawn.

A dynamic model is beneficial when tracking motion in a "steady state", for instance a repetitive motion pattern (i.e. walking or running) or when tracking beforehand known actions. This model yields to a more efficient usage of particles but it often requires annotated data to train it. Gaussian processes have been employed in [RRR08] to model a set of repetitive actions to efficiently drive particles through a reduced state space thus noticeably improving the efficiency of their PF algorithm. Another example is provided in [CGH05] where Markov models are used to model motion patterns. Although these methods provide good results when tracking actions captured by theirs dynamic models, they fail when tracking unseen/unmodelled agile motions. Moreover, these dynamic models need to be selected beforehand according to the action to be tracked thus becoming a limited analysis tool.

When aiming at tracking any type of motion, no dynamic model is adopted [DR05] ($\mathbf{y}_t^j = \mathbf{y}_{t-1}^j + \mathcal{N}(\mathbf{0}, \mathbf{P})$) or, instead, a smooth motion assumption is taken [BSB05] ($\mathbf{y}_t^j =$

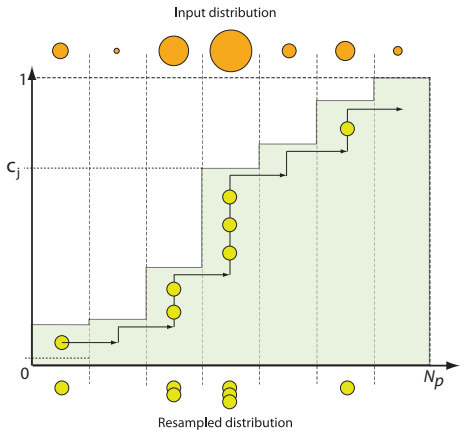


Figure 3.2: *Systematic resampling algorithm. In the top of the figure, initial weighted particles are resampled into the equally weighted particles at the bottom. The cumulative function c_j is employed to decide the amount of resampled particles derived from a weighted particle.*

$2\mathbf{y}_{t-1}^j - \mathbf{y}_{t-2}^j + \mathcal{N}(\mathbf{0}, \mathbf{P})$). The price we pay for this is a less economical use of particles than would be ideal, and the potential for jittery trajectories. The latter could be addressed by smoothing the recovered pose trajectories. By defining an appropriate drift in the propagation one may address problems such as inter-penetrating limbs or angles exceeding the anatomical joint limits, as will be addressed further in this thesis.

3.1.2.3 Evaluation

A problem arising when applying PF techniques to computer vision problems is to derive a valid observation model $p(\mathbf{z}_t|\mathbf{y}_t^j)$ relating the input data $\mathbf{z}_t$ with a given particle state $\mathbf{y}_t^j$. Nevertheless, even if such likelihood model can be defined, its evaluation may be very computationally inefficient. Instead of that, a fitness function $w(\mathbf{z}_t, \mathbf{y}_t^j) : \mathcal{X} \rightarrow [0, 1]$ can be constructed according to the likelihood function, such that it provides a good approximation of $p(\mathbf{z}_t|\mathbf{y}_t^j)$ but is also relatively easy to calculate.

Let us define a generic cost function $C(\mathbf{z}_t, \mathbf{y}_t^j)$ that measures the match between the observation that would be produced by the particle state $\mathbf{y}_t^j$ and the input data $\mathbf{z}_t$. This function may take into account many criteria and is a design parameter related with the structure of the problem that we want to solve. There is a number of image based features and criteria employed in the literature to construct the cost function: silhouette overlap [DR05], edge distance [IB98], color similarity [CGH05], etc. Constructing a joint cost function taking into account some features, C_k, can be achieved through a linear combination:

$$C(\mathbf{z}_t, \mathbf{y}_t^j) = \sum_{k=1}^{N_{\text{features}}} \lambda_k C_k(\mathbf{z}_t, \mathbf{y}_t^j), \qquad \sum_{k=1}^{N_{\text{features}}} \lambda_k = 1, \tag{3.13}$$

where coefficients λ_k weight the influence of each feature on the global cost function (assuming $C_k(\mathbf{z}_t, \mathbf{y}_t^j) \in [0, 1]$). Usually, these coefficients are set empirically [DR05, RRR08], although some research towards automatic adjustment has been presented [Mit03].

Assuming that the involved errors follow a Gaussian distribution, it has been proved in [LRH04] that an accurate way to define the fitness function is:

$$w(\mathbf{z}_t, \mathbf{y}_t^j) \propto \exp\left(-\frac{C(\mathbf{z}_t, \mathbf{y}_t^j)^2}{2\sigma^2}\right). \tag{3.14}$$

This choice has the advantage that even weak hypotheses have finite probability of being preserved, which is desirable in the case of very sparse samples.

The linear combination expressed in Eq.3.13 does not model the interactions among the involved scores and may lead to poor performance in presence of noisy observations. Typically, this occurs when an inaccurate score masks the others. Within the framework of this thesis we have experimented with a more descriptive fitness function to avoid this problem:

$$w(\mathbf{z}_t, \mathbf{y}_t^j) \propto \exp\left(-\frac{1}{2}\left[C_1\ C_2\ \cdots\ C_{N_{\text{features}}}\right]^\top \mathbf{H}^{-1}\left[C_1\ C_2\ \cdots\ C_{N_{\text{features}}}\right]\right). \tag{3.15}$$

Whit this proposal, using a multi-variate Gaussian function, it allows us to control the influence of each parameter individually and their cross-dependencies through the structure of covariance matrix $\mathbf{H}$.

3.1.2.4 Estimation

The best state at time t, $\mathcal{Y}_t$, is derived based on the discrete approximation of Eq.3.7. The most common solution is the Monte Carlo approximation of the expectation

$$\mathcal{Y}_t = \mathrm{E}[\mathbf{y}_t] = \int_{\mathbf{y}_t \in \mathcal{X}} \mathbf{y}_t\ p(\mathbf{y}_t|\mathbf{z}_{1:t}) d\mathbf{y}_t \approx \sum_{j=1}^{N_\mathrm{p}} \pi_t^j \mathbf{y}_t^j. \tag{3.16}$$

It must be noted that in the case were there are several peaks in the likelihood function, the output of Eq.3.16 can be incorrect since the average may fall far of the maximum. Some approaches to PF circumvent this problem by selecting the particle with highest or the particle corresponding to the median of weights. Nonetheless, the strategies presented in this thesis rely on more elaborated PF algorithms that do not incur in such pitfalls.

3.2 Simulated Annealing

As its name implies, the simulated annealing exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system. The algorithm is based upon that of Metropolis *et al.* [MRR+53], which was originally proposed as a means of finding the equilibrium configuration of a collection of atoms at a given temperature. The connection between this algorithm and mathematical minimization was first noted by Pincus [Pin70], but it was Kirkpatrick *et al.* [KGV83] who proposed it as the basis of an optimization technique for combinatorial (and other) problems.

Optimizing a multi-modal objective function $U(\mathbf{y})$ can be done through Markov chain theory. It proceeds by defining a distribution over the function values as

$$P(\mathbf{y}) \propto e^{-\lambda U(\mathbf{y})}. \tag{3.17}$$

The aim is then to generate samples $\mathbf{y}_i$ from this distribution, in the knowledge that as $\lambda \to \infty$, the probability mass concentrates on the minimum of $U(\mathbf{y})$, and hence the samples $\mathbf{y}_i$ will cluster around the minimum value state. Simulated annealing's major advantage over other methods is an ability to avoid becoming trapped at local minima.

Samples from the distribution in Eq.3.17 are generated using the Metropolis-Hastings algorithm[3] [MRR+53] which generates a Markov sequence of points whose distribution will converge to $P(\mathbf{y})$. In this context, generating a sequence starting at $\mathbf{y}_0$ with a large value of λ yields to poor results if $U(\mathbf{y})$ has isolated minima since the sequence can easily get trapped in a local mode of $P(\mathbf{y})$ (typically, the closest to $\mathbf{y}_0$).

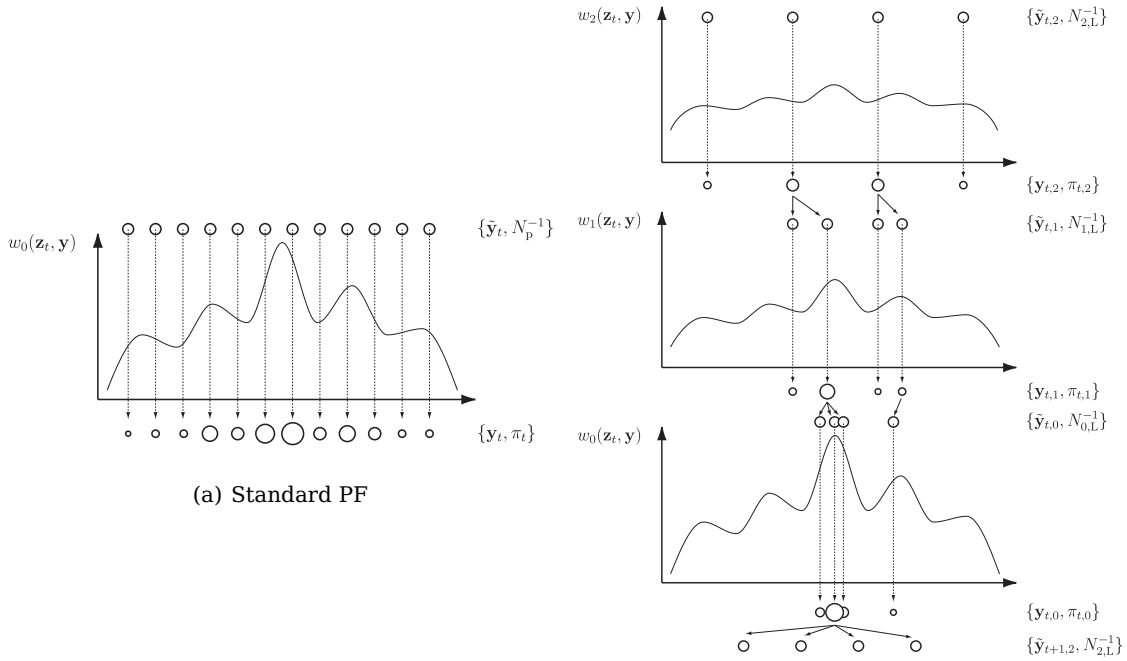
The annealing process is a heuristic technique to avoid such situations by selecting different values of λ through the process. Initially, λ is set to be small thus smoothing the probability function $P(\mathbf{y})$ and allowing a broad exploration of the search space. Samples are generated from this distribution, and then the value of λ is increased. Then, new samples are generated from the new distribution starting from the final state of the previous sequence, and so on. Each increase of λ successively discards (in a probabilistic sense) regions that contain little of the probability mass of the distribution.

The set of values for $\lambda = \{\lambda_L, \cdots, \lambda_1\}$ is referred as the annealing schedule. These values are to be assigned as a compromise between speed and efficacy: slow annealing is more likely to find a globally optimal solution, but might be computationally prohibitive. This method can be successfully applied to particle filtering when we view this process as generating samples from a sequence of distributions, $\{P_{\lambda_L}, \cdots, P_{\lambda_1}\}$, with $P_{\lambda_m} \propto P_{\lambda_1}^{\beta_m}$, for $1 = \beta_1 > \beta_2 > \cdots > \beta_L$, and where $\beta_m = \lambda_m/\lambda_1$. The overall behavior of the algorithm when applied to particle filtering will tend to concentrate particles around the global maxima of the weighting function $w(\mathbf{z}_t, \mathbf{y}_t)$ by moving particles through a set of progressively smoothed versions of $w(\mathbf{z}_t, \mathbf{y}_t)$. This combination will allow avoiding getting misguided by local maxima as seen in Figure 3.3(a). The idea of annealed particle filtering was first introduced by Deutscher *et al.* [DR05] in the context of articulated motion tracking.

3.2.1 Annealed Particle Filter

The main idea in the annealed particle filter (APF) is to use a set of weighting functions instead of a single one during the filtering loop. A series of functions are used, $\{w_m(\mathbf{z}_t, \mathbf{y}_t^j)\}_{m=1}^L$, where w_{m+1} slightly differs from w_m and represents a smoothed version of it. In the standard PF algorithm, samples should be drawn from the w_1 function, which might be peaky, and therefore a large number of particles should be needed in order to find the global maxima. In the APF approach, function w_L is designed to be

[3] In mathematics and physics, the Metropolis-Hastings algorithm is a method for creating a Markov chain that can be used to generate a sequence of samples from a probability distribution that is difficult to sample from directly.



(a) Standard PF

(b) Annealed PF

Figure 3.3: *Comparison between PF and APF. In (a), PF is employed to explore the state space containing multiple maxima and therefore misguiding the final estimation. In (b), APF produced a more efficient placing of particles through 3 layers of annealing. Note that the number of (efficient) particles is the same in both cases.*

very broad, representing the overall trend of the search space while w_1 should be peaky, emphasizing local features. The usual method to achieve this effect is by setting:

$$w_m(\mathbf{z}_t, \mathbf{y}_t^j) = w(\mathbf{z}_t, \mathbf{y}_t^j)^{\beta_m}, \tag{3.18}$$

for $\beta_1 > \beta_2 > \cdots > \beta_L$, where w_1 is the original weighting function. Since we are estimating the location of the global maxima of $w(\mathbf{z}_t, \mathbf{y}_t^j)$, it is not necessary to impose $\beta_1 = 1$. Factor β_m will define the shape of w_m: large values of β_m will produce peaky w_m functions while small values will result in broad and flat functions w_m. The sequence of values β_m will determine the behavior of particles within the state space and its influence is discussed in §3.2.2.

As it is shown in Algorithm 3 and Figure 3.3(b), the APF consists in exploring a set of L progressively smoothed versions of the original weighting function w towards an efficient placement of the particle set around the global maxima of this function. Two examples of APF applied to human motion capture are displayed in Figure 3.4. Within the context of APF, we will denote $N_{\mathrm{p,L}}$ as the number of particles per layer and $N_{\mathrm{p}} = L \cdot N_{\mathrm{p,L}}$ as the *equivalent* number of particles (to be able to compare with the standard PF algorithm). The filtering process is described as:

Algorithm 3: Annealed Particle Filtering

repeat
  $\pi^j_{t,M} = \pi^j_{t-1,1}$
  for $m = M$ **to** 1 **do**
    ▷ Resample (as done in §3.1.2.1)
    $\mathbf{y}^j_{t,m} \rightarrow \tilde{\mathbf{y}}^j_{t,m-1}$
    $\pi^j_{t,m} = N^{-1}_{\mathrm{p,L}}$
    ▷ Propagation (as done in §3.1.2.2)
    $\mathbf{y}^j_{t,m} = \tilde{\mathbf{y}}^j_{t,m} + \mathcal{N}(\mathbf{0}, \mathbf{P}_m)$
    ▷ Evaluation
    $\pi^j_{t,m} = w_m(\mathbf{z}_t, \mathbf{y}^j_t)$
  end
  ▷ Estimation
  $t = t + 1$
until end

1. Starting at $m = L$, the particle set $\{(\mathbf{y}^j_{t,L}, \pi^j_{t,L}\}^{N_{\mathrm{p,L}}}_{j=1}$ is initialized as the particle set obtained in the last annealing layer at $t-1$, that is $\{(\mathbf{y}^j_{t-1,1}, \pi^j_{t-1,1}\}^{N_{\mathrm{p,L}}}_{j=1}$.

2. Particle set is resampled with replacement and the associated weights are reset to $N^{-1}_{\mathrm{p,L}}$, generating the set $\{(\tilde{\mathbf{y}}^j_{t,m}, N^{-1}_{\mathrm{p,L}}\}^{N_{\mathrm{p,L}}}_{j=1}$.

3. Propagation is applied to every particle state as: $\mathbf{y}^j_{t,m} = \tilde{\mathbf{y}}^j_{t,m} + \mathcal{N}(\mathbf{0}, \mathbf{P}_m)$, where $\mathcal{N}(\mathbf{0}, \mathbf{P}_m)$ is a Gaussian multi-variate random variable with zero mean and covariance matrix $\mathbf{P}_m$.

4. Weight associated to each particle is computed as $\pi^j_{t,m} = w_m(\mathbf{z}_t, \mathbf{y}^j_{t,m})$. Particle weights are normalized so that $\sum \pi^j_{t,m} = 1$.

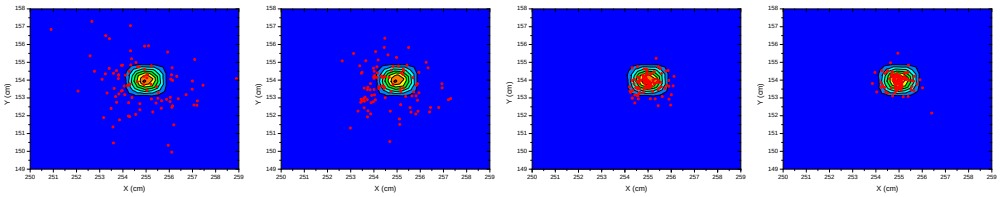
5. Particle set $\{(\mathbf{y}^j_{t,m}, \pi^j_{t,m}\}^{N_{\mathrm{p,L}}}_{j=1}$ is employed to initialize the next layer, $m-1$. Steps 2-5 are repeated until reaching $m = 1$.

6. The set $\{(\mathbf{y}^j_{t,1}, \pi^j_{t,1}\}^{N_{\mathrm{p,L}}}_{j=1}$ is employed to estimate the optimal state configuration as:

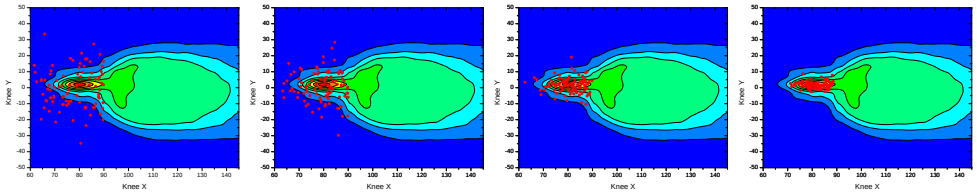
$$\mathcal{Y}_t = \sum_{j=1}^{N_{\mathrm{p,L}}} \mathbf{y}^j_{t,1} \pi^j_{t,1}. \tag{3.19}$$

   This set is used to initialize the particle set corresponding to $m = L$ at the next time instant $t+1$, $\{(\mathbf{y}^j_{t+1,L}, \pi^j_{t+1,L}\}^{N_{\mathrm{p,L}}}_{j=1}$.

Although this is an introduction to annealed particle filters from the engineering point of view, a more in-depth mathematical description of this algorithm can be found in [GPS+07].



(a) Centroid XY



(b) Hip angles θ_x,θ_y

Figure 3.4: *APF operation on real data. Two examples of particle evolution using 4 annealing layers in a real case of human motion capture. Red dots depict particles on the likelihood (cost) function.*

3.2.2 Filter settings

Parameters employed by the APF algorithm are several and most of them do not have an analytic procedure to be set, hence empirical values are usually selected. The number of annealing layers, L, and the number of particles per layer, $N_{\mathrm{p,L}}$, are selected tacking into account the associated complexity proportional to the equivalent number of particles $N_{\mathrm{p}} = L \cdot N_{\mathrm{p,L}}$. In our framework, several configurations of L and $N_{\mathrm{p,L}}$ are tested and the overall behavior is analyzed towards selecting the optimal pair. Usually, an exploratory dataset ($\sim 10\%$ of the total analysis data) is employed for such experiments due to the overall computational complexity of the problem.

Design of the annealing schedule, β_m, is crucial for this algorithm since it determines the behavior and scatter of particles at each layer. Our choice follows the criteria stated by Deutscher *et al.* [DR05] using the analogy between the physics problem and particle filtering where temperature is equivalent to the particle survival rate α defined in Eq.3.11. Factor α for a given annealing coefficient β_m can be written as:

$$\alpha(\beta_m) = \frac{\widehat{N_{\mathrm{eff}}}(\beta_m)}{N_{\mathrm{p,L}}} = \frac{1}{\left(\sum_{j=1}^{N_{\mathrm{p,L}}} (\pi_{t,m}^{j})^2\right) N_{\mathrm{p,L}}}. \tag{3.20}$$

As it has been stated in Eq.3.18, weights $\pi_{t,m}^{j}$ can be referred to the original weighting function w thus leading to the following expression of $\alpha(\beta_m)$:

$$\alpha(\beta_m) = \frac{1}{\left(\sum_{j=1}^{N_{\mathrm{p,L}}} w(\mathbf{z}_t, \mathbf{y}_t^j)^{2\beta_m}\right) N_{\mathrm{p,L}}}, \tag{3.21}$$

with the normalization restriction $\sum_j w(\mathbf{z}_t, \mathbf{y}_t^j)^{\beta_m} = 1$. A high survival rate corresponds to an even spread probability mass, while a low one suggests the mass is concentrated in a

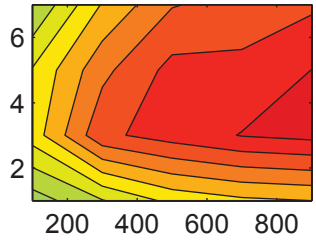


Figure 3.5: *Over-annealing effect. Performance plot for several layers (y axis) and particles per layer (x axis).*

few particles. Moreover, this factor α can be understood as the fraction of particles that will survive the resampling step. By properly selecting values of β_m, we can control the amount of particles that survive at each annealing layer that corresponds to the particles with higher weights. In this way, particles are progressively concentrated in the global maxima of the likelihood function w.

Let us define α_m as the desired particle survival rate at layer m. Therefore, it is required to estimate the value of β_m, and this can be easily achieved by searching over β_m on Eq.3.21, that is a monotonically decreasing function with β_m. In our case, the well known *regula falsi* root finding method is applied to solve this equation and find β_m. It must be taken into account that weights $w(\mathbf{z}_t, \mathbf{y}_t^j)$ employed in Eq.3.21 can be stored into memory to speed up the estimation process of β_m.

Regarding the propagation noise, diagonal elements of covariance matrix $\mathbf{P}_0$ are set to half of the maximum expected variation of each variable of the state space over one time step. The amount of diffusion added to each successive annealing layer should decrease at the same rate as the resolution of the particle set at layer m increases. It is proposed to set:

$$\mathbf{P}_m = \alpha_L \cdots \alpha_{m-1} \cdot \mathbf{P}_0. \tag{3.22}$$

Finally, annealing rates α_m are influenced by the number of layers L. When using a large number of layers, a lower rate of annealing can be used to more accurately explore the several maxima. Nonetheless, an empirical criteria is taken and we set $\alpha_m = 0.5, \forall m$, providing satisfactory results.

3.2.3 Over-annealing effects

Although annealed PF is a good strategy to deal with multimodal likelihood functions, it is not free from some common issues inherent to PF algorithms. Let us consider the situation where, for a fixed number of particles per layer $N_{\text{p,L}}$ and an annealing rate α_m, we set the number of layers L to be a large number. Intuitively, it can be assumed that the larger the number of layers, the better the obtained approximation. However, when examining Figure 3.3(b), it may be seen that, by adding more annealing layers, particles will tend to concentrate in the main mode of the likelihood function thus not properly representing the overall structure of it. This particle impoverishment effect has been already introduced in §3.1.2.1. In the standard PF algorithm, this effect is circumvented

by adding a sufficient amount of propagation or process noise to the state of the particles. In the annealed PF case, this process noise is decreased by a factor α at every layer, then decreasing the original process noise described by covariance matrix $\mathbf{P}$ (see Eq.3.22) with a geometric proportion.

An example of over-annealing is depicted in Figure 3.5 where a performance score is shown in a color scale for several combinations of number of layers and particles per layer. This effect is shown by the fact that, for $L > 4$, the performance decreases gradually, meaning that the particle set does not longer properly represent the likelihood function. In this thesis, when applying an annealed PF, the optimal values for L and $N_{\mathrm{p,L}}$ will be empirically estimated using a fraction of the analysis datasets. Although, an automatic selection of these values would be desirable, no standard solution is yet reported in the literature.

3.3 Conclusion

Monte Carlo based techniques have been found to be an efficient estimation and tracking tool when dealing with problems involving large dimensional state space with a multi-modal profile. Particularly, simulated annealing combined with particle filtering is the state of the art in human motion capture algorithms.

In this chapter we have introduced the elemental concepts of Monte Carlo theory to better understand the forthcoming algorithms that will unfold along this thesis. Some examples have been presented to better illustrate the relationship between particle filters and human motion capture.

4

Multi-person voxel based tracking

TRACKING multiple objects and keeping record of their identities along time in a cluttered dynamic scene is a major research topic in image processing and a number of applications may be derived from tracking information such as visual surveillance, automatic scene classification or supporting HCI tasks. In the context of our research, the person position inside our analysis scenario can be used to place and constrain the location of the human body model during the initialization phase, as will be seen in Chapter 6.

We first introduce a methodology to multi-person tracking based on a colored voxel representation of the scene as the start of the processing chain. The contribution of this chapter is twofold. First, we emphasize the importance of the creation and deletion of tracks, usually neglected in most of tracking algorithms, that has indeed an impact on the performance of the overall system. A general creation/deletion of tracks technique is presented. The second contribution is the filtering step where two techniques are introduced. The first technique is to apply a particle filter to the input voxels to estimate the centroid of the tracked targets. However, we realized that this process was far from real-time performance and we proposed an alternative: Sparse Sampling. This method aims at decreasing computation time by means of a novel tracking technique based on the seminal particle filtering principle. Particles no longer sample the state space but instead a magnitude whose expectancy produces the centroid of the tracked person: the surface voxels. The likelihood evaluation relying on occupancy and color information is computed on local neighborhoods thus dramatically decreasing the computation load of the overall algorithm.

Effectiveness of the proposed techniques is assessed by means of objective metrics defined in the framework of the CLEAR [CLE07] multi-target tracking dataset. Computational performance is thoroughly reviewed towards proving the real-time operation of the SS algorithm. Fair comparisons with state-of-the-art methods evaluated using the same dataset are also presented and discussed.

The following articles have been published in this field: [CFCP05b, ACFS$^+$06, LCFC07, CFSC07a, CFSC$^+$08, CFSCP08, BTNCF08a, BTNCF08b, CFCPM09a, NPS$^+$09].

4.1 Introduction

A number of methods for camera based multi-person 3D tracking have been proposed in the literature [BES06, CFSC07a, KBD03, LCB07]. A common goal in these systems is robustness under occlusions created by the multiple objects cluttering the scene when estimating the position of a target. Single camera approaches (see [YJS06] for a survey), have been widely employed but they are vulnerable to occlusions, rotation and scale changes of the target. In order to avoid these drawbacks, multi-camera tracking techniques exploit spatial redundancy among different views and provide 3D information as well. Integration of data extracted from multiple cameras has been proposed in terms of a fusion at feature level as image correspondences [CFCP05b] or multi-view histograms [Lan06] among others. Information fusion at data or raw level has been achieved by means of voxel reconstructions [CKBH00], polygon meshes [IS03], etc.

Most of the multi-camera approaches to this task rely on a separate analysis of each camera view to extract some features followed by a fusion of these features to finally generate an output. Exploiting the underlying epipolar geometry of a multi-camera setup towards finding the most coherent feature correspondence among views was first tackled in [MSJ00] using algebraic methods together with a Kalman filter and further developed in [FS02]. Epipolar consistency within a robust Bayesian framework was presented in [CFCP05b]. Other systems rely on detecting semantically relevant patterns among multiple cameras to feed the tracking algorithm as done in [KTPP07] by detecting faces. Particle filtering (PF) has been a commonly employed algorithm due to its ability to deal with problems involving multimodal distributions and non-linearities. In [Lan06], the authors propose a multi-camera appearance based PF tracker exploiting foreground and color information and several contributions have also employed the same input data together with an adapted PF algorithm: [BES06, LCB07]. Occlusions, being a common problem in feature fusion methods, have been addressed in [LH08] using HMM to model the temporal evolution of occlusions within a PF algorithm. Information about the tracking scenario can also be exploited towards detecting and managing occlusions as done in [OWS+07] by modeling the occluding elements such as furniture in a training phase before tracking. It must be noted that, in our framework, it is assumed that all employed cameras will be covering the area under study. However, other approaches to multi-camera/multi-person tracking do not require this condition to be fulfilled leading to the non-overlapped multi-camera tracking algorithms [BER02].

Multi-camera/multi-person tracking algorithms based on a data fusion before doing any analysis was pioneered in [LCFC07] by using a voxel reconstruction of the scene and this idea was further developed in [CFSC07a, CFSCP08]. Up to our knowledge, this is the first attempt to multi-person tracking employing a data fusion from multiple cameras as the input of the algorithms.

Applications based on the obtained tracking information are numerous: multi-person tracking has been found useful for automatic scene analysis [PT08], human computer interfaces [CHI07] and detection of unusual behaviors in security applications [HHD00]. Integration of tracking outputs with audio information leaded to multimodal approaches to group action analysis [MGPB+05] or focus of attention estimation [CFSC+08].

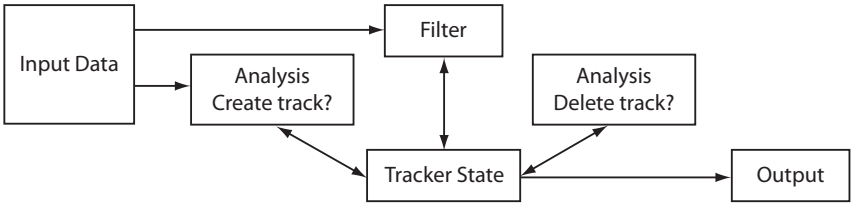


Figure 4.1: *Multi-person tracking scheme.*

4.2 Tracker design methodology

Designing a multi-target tracking system involves making a series of assumptions and strategies taking into account constraints regarding system complexity or other constraints derived from the particularities of the problem. Specifically, we are aiming at obtaining near real-time algorithms. Typically, a multi-target tracking system can be depicted as in Figure 4.1 and comprises a number of elemental modules. Although most papers present techniques that contribute to the filtering module, the overall architecture is rarely addressed assuming that some blocks are already available. In this section, this scheme will be analyzed and some proposals for each module will be presented. The filtering step, being our major contribution, will be addressed in a separate section.

4.2.1 Input and Output data

When addressing the problem of multi-person tracking within a multi-camera environment, a strategy about how to process this information is needed. As it has been already presented, many approaches perform an analysis of the images separately and combine the results by using some geometric constraints [Lan06]. We may define this approach as an information combination by fusion of decisions. However, a major issue in this procedure is dealing with occlusion and perspective effects. It has been suggested that a more efficient way to combine information is by data fusion [HM04]. In our case, that would correspond to combine information from all images to build up a new data representation and apply the algorithms directly on these data. Several data representations aggregating the information of multiple views have been proposed in the literature such as voxel reconstructions [CKBH00, KS00], level sets [FK02], polygon meshes [IS03], conexels [CS06], depth maps [KZ04], etc. In our research, we opted for a colored voxel representation, as previously introduced in §2.2.4. Scene analysis using other multi-camera aggregated data representations is still a field where computational load is an issue.

The output of this algorithm will be a number of hypotheses for the centroid position of each of the targets present in the scene.

4.2.2 Tracker state

One of the major challenges in multi-target tracking is the estimation of the number of targets and their positions in the scene, based on a set of uncertain observations. The definition of the tracker state, that is, how a number of tracks are managed along

time, will be a constraint in the design of the filtering step and affects the computational complexity, as well. This issue can be addressed from two perspectives. First, extending the theory of single-target algorithms to multiple targets. This approach defines the working state space $\mathcal{X}$ as the concatenation of the positions of all N_{T} targets as $\mathcal{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_{N_{\mathrm{T}}}]$. One of the main difficulties is the fact that the dimensionality of this space is time variant. A solution proposes predefining a maximum number of targets and then declare a group of them to be hidden [Sto01]. Monte Carlo approaches, and specifically PF approaches, to this problem have to face the exponential dependency between the number of particles required by the filter and the dimension of $\mathcal{X}$ turning it to be computationally infeasible. Recently, a solution based on random finite sets achieving linear complexity has been presented [MPRC07].

Multi-target tracking can be also tackled by tracking each target independently, that is to maintain N_{T} trackers with a state space $\mathcal{X}_i = \mathbf{x}_i$. In this case, the system attains a linear complexity with the number of targets, thus allowing feasible implementations [Cox93]. However, interactions among targets must be modeled in order to ensure the most independent set of tracks. That is to define an interaction model and this will be particularly defined for each filtering scheme. This approach to multi-person tracking will be adopted in our research and further reviewed in §4.3.

4.2.3 Track creation/deletion

A crucial factor in the performance of a tracking system is the module that addresses the creation and deletion of filters. The creation of a new tracker is independent of the employed filtering technique and only relies on the input data and the current state (position) of the tracks in the scene. On the other hand, the deletion of a filter is driven by the performance of the tracker.

The initialization of a new filter is determined by the correct detection of a person in the analyzed scene. This process is crucial when tracking, and its correct operation will drive the overall system's accuracy. However, despite the importance of this step, little attention is paid to it in the design of multi-object trackers in the literature. Only few papers explicitly mention this process such as [TPC08] that employs a face detector to detect a person or [BGS07] that uses scout particle filters to explore the 3D space for new targets. Moreover, it is assumed that all targets in the scene are of interest, i.e. people, not accounting for spurious objects, i.e. furniture, shadows, etc. In this section we introduce a method to properly handle the creation and deletion of filters from a Bayesian perspective.

Track creation criteria

The 3D input data $\mathcal{V}$ fed to the tracking system generated using the procedure descripted in §2.2.4 is usually corrupted and presents a number of inaccuracies such as objects not reconstructed, mergings among adjacent blobs, spurious blobs, etc. Hence, defining a track initialization criterium based solely on the presence of a blob might lead to poor performance of the system. For instance, objects such as furniture might be reconstructed and tracked. Instead, a classification of the blobs based on a probabilistic

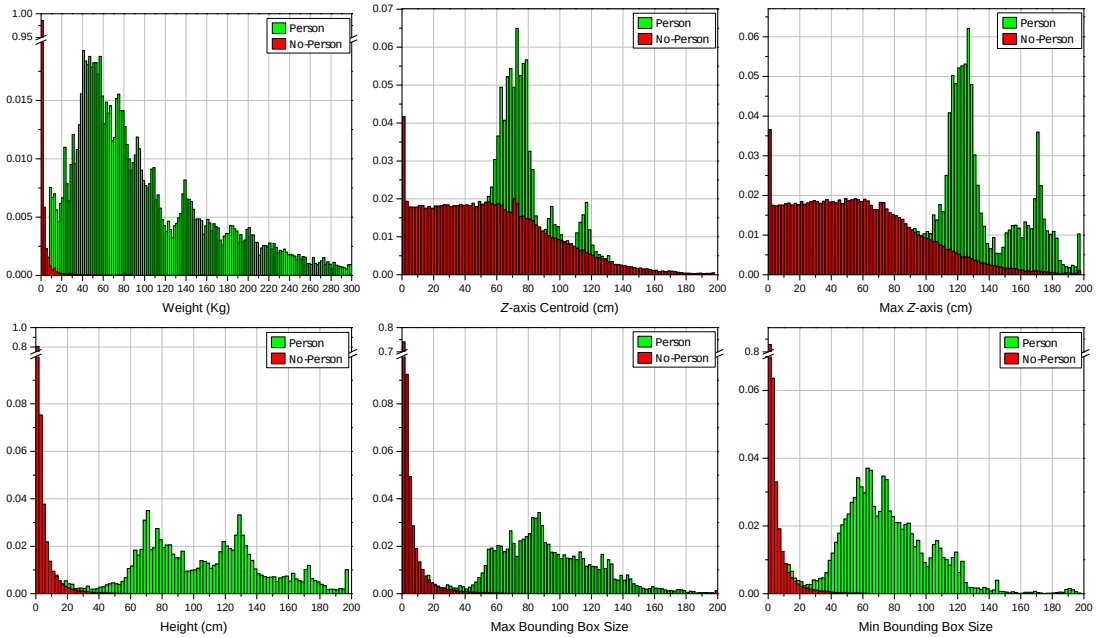


Figure 4.2: *Normalized histograms of the variables conforming the feature vector employed by the person/non-person classifier.*

criteria can be applied during this initialization process towards a more robust operation. Training of this classifier is based on the development set of the used database together with the available ground truth describing the position of the tracked objects.

Let $\mathbf{X}^{\mathrm{GT}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_{\mathrm{GT}}}\}$ be the ground truth positions of the N_{GT} targets present in the scene at a given instant. Once the reconstruction $\boldsymbol{\mathcal{V}}$ is available, a connected component analysis (*CCA*) is performed over this data thus obtaining a set of K disjoint components, $\boldsymbol{\mathcal{C}}_i$, fulfilling:

$$\boldsymbol{\mathcal{V}} = \bigcup_{i=1}^{K} \boldsymbol{\mathcal{C}}_i. \tag{4.1}$$

We will consider the region of influence of a target with centroid $\mathbf{x}$ as the ellipsoid $\mathcal{E}$ with axis size $\mathbf{s} = (s_x, s_y, s_z)$ centered at $\mathbf{c} = \mathbf{x} = (c_x, c_y, c_z)$ described by the equation:

$$\mathcal{E}(\mathbf{x}, \mathbf{s}) : \left\{ \left(\frac{x - c_x}{s_x}\right)^2 + \left(\frac{y - c_y}{s_y}\right)^2 + \left(\frac{z - c_z}{s_z}\right)^2 \leq 1 \right\}. \tag{4.2}$$

A mapping is defined such that for every $\mathbf{x}_j \in \mathbf{X}^{\mathrm{GT}}$ a component $\boldsymbol{\mathcal{C}}_i$ is assigned. This process is defined as follows: first, a region of influence $\mathcal{E}(\mathbf{x}_j, \mathbf{s})$ with size $\mathbf{s} = (s_x, s_y, [\mathbf{x}_j]_z)$ centered at $\mathbf{c} = \mathbf{x}_j$ is placed in the 3D space. The radii s_x and s_y are chosen to contain an average person, $s_x = s_y = 30$ cm. Then, the assignation is defined as:

$$\mathbf{x}_j \rightarrow \arg\max_i \left| \mathcal{E}(\mathbf{x}_j, \mathbf{s}) \cap \boldsymbol{\mathcal{C}}_i \right|, \tag{4.3}$$

that is to assign $\mathbf{x}_j$ to the component with the largest volume enclosed in the region of influence. It must be noted that some $\mathbf{x}_j$ might have not any $\boldsymbol{\mathcal{C}}_i$ associated due to a

Feature	Expression
Weight	$\lvert\boldsymbol{\mathcal{C}}_i\rvert s_{\mathcal{V}}^3 \rho \qquad \rho = 1.1$ [gr/cm^3]
Centroid (z-axis)	$\lvert\boldsymbol{\mathcal{C}}_i\rvert^{-1} \sum_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_z$
Top	$\max_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_z$
Height	$\max_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_z - \min_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_z$
Bounding Box	$\max \left\{ \max_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_x - \min_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_x, \max_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_y - \min_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_y \right\}$ $\min \left\{ \max_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_x - \min_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_x, \max_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_y - \min_{\mathcal{V}\in\boldsymbol{\mathcal{C}}_i} \mathcal{V}_y \right\}$

Table 4.1: *Features employed by the person/no-person classifier.*

wrong segmentation or faulty reconstruction of the target. Moreover, the set of components not associated to any ground truth position can be identified as spurious objects, reconstructed shadows, etc.

Finally, we have grouped the set of connected components $\boldsymbol{\mathcal{C}}_i$ in two categories: person and non-person. A set of features are extracted from each of these components thus conforming the characteristics that will be used to train a person/no-person binary classifier. This set of extracted features is described in Table 4.1.

Input data is defined by a voxel set and the size of the side of each voxel, $s_{\mathcal{V}}$. As it has been mentioned in §2.2.4, the reconstruction of the scene may vary depending on the voxel size and the employed shape-from-silhouette method. In order to ensure the independence of the person/no-person classifier in relation with $s_{\mathcal{V}}$, the following experiment has been devised. First, several 3D reconstructions of the same time instant are generated using a number of different voxel sizes (that is: $s_{\mathcal{V}} = \{2, 5, 10, 15\}$ cm). Second, proposed features are extracted for all these reconstructions and, finally, histograms of these features are built up. These histograms are compared through the Bhattacharya distance always obtaining a distance over 0.95 proving that the obtained classifier will not significantly depend on the input voxel size $s_{\mathcal{V}}$. However, in order to have the more accurate data to train our classifier, the smallest voxel size was used.

In order to characterize the objects to be tracked and to decide the best classifier system, we have done an exploratory data analysis [Tuc77], which will allow us to contrast the underlying hypotheses of the classifiers with the actual data. Histograms of these features are computed as shown in Figure 4.2 and scatter plots depicting the cross dependencies among all features are displayed in Figure 4.3. Some remarks can be drawn from the study of these figures:

- The histograms of the features related with the person class properly describe a human body. For instance, the top (or maximum z-axis magnitude) presents a bi-modal distribution with centers located in $z = 120$ and $z = 170$ cm, clearly representing the sitting and standing position of the people inside the room. Also, height and z-centroid maxima are placed nearly in the same locations.

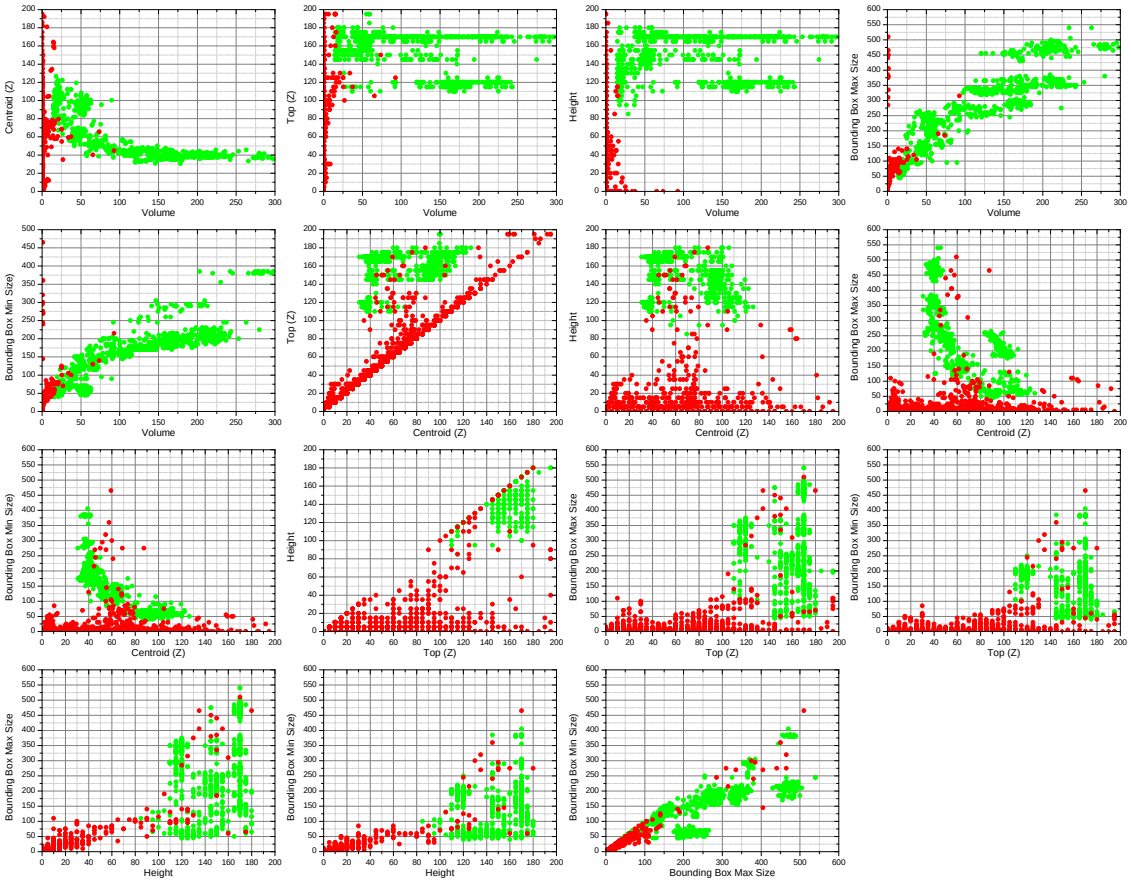


Figure 4.3: *Scatter plots of the combination of the variables involved in the creation/deletion module. Green markers stand for instances of the person class while red markers are assigned to the non-person class.*

- The histograms related with the non-person class tend to attain low values for most of the features thus indicating that most of these components are small in size. Centroid in the z-axis presents a flat distribution showing no spatial preference in the manifestation of these spurious blobs.
- In most cases, a parametric, i.e. Gaussian, distribution can not be assumed and multimodal and non-symmetric distributions are present.

Observing Figure 4.2, we see that some variables are easily separable, i.e. weight, height and bounding box, as well as having a low cross-dependency with other features. Figure 4.3 shows that a pairwise comparison of the features reduces the overlap between classes, and also shows that the boundaries between classes are spread along the space, and the points do not cluster. Therefore, a good classifier would be one that partitions the space by means of hyperplanes. In contrast, classifiers based on parametric approximations of the *pdf* or clustering will suffer of a much higher variability. Thus, it will be expected that classifiers such as Gaussian, K-Means or Parzen will have worst

Method	Precision	Recall	F-Measure
Gaussian	0.0179	0.9011	0.0351
Neural Network	0.0179	0.9013	0.0351
K-Means	0.2429	0.8992	0.3825
PCA	0.2884	0.9090	0.4379
Parzen	0.9568	0.8137	0.8795
MoG (12 Gaussians)	0.9267	0.8697	0.8973
Decision Trees	**0.9844**	**0.9920**	**0.9882**

Table 4.2: *Results for the tested person/no-person classifiers.*

performance than classifiers based on planes such as MLP or classification trees.

A number of standard binary classifiers were tested and their performances are reported in Table 4.2, namely Gaussian, Mixture of Gaussians, Neural Networks, K-Means, PCA, Parzen and Decision Trees [BFOS93, DHS00]. Due to the aforementioned properties of the statistic distributions of the features, some classifiers are unable to obtain a good performance, i.e. Gaussian, PCA, etc. Other classifiers require a large number of characterizing elements, such as K-Means, MoG or Parzen. It must be noted that applying some transformation to the input variables, the performance of some of these classifiers might be improved. However, these transformations, sometimes involving non-linearities, may have some undesirable effects such as instabilities when the input is not in the proper range [DHS00].

Decision trees [BFOS93] have reported the best results. The rules of the tree consist of a disjunction of conjunctions [Mit97]. This technique generates a binary decision tree that aims at obtaining the maximum class similarity on each of its leafs, that is the minimum entropy among the elements contained in the leaf. This technique has proved effective in classification problems where the involved *pdf*s present an heterogeneous distribution. Separable variables such as height, weight and bounding box size are automatically selected to build up the decision tree as seen in Figure 4.4. Note that, geometrically, the decision tree cuts the feature space with planes parallel to the axis. The criterion for stopping the growth of the tree was cross-validation, in order to assure a good performance on unseen data. Also, the cross-validation criterion solves the problem of the instability of the classifier training [BFOS93]. Instability means that changing a small fraction of the training data, the structure of the tree changes significantly. The effect of cross-validation is to prune the tree to a number of levels that maximize the performance on unseen data and, at the same time, eliminates the lower parts of the tree that suffer of instability.

Another complementary criterium employed in the creation of new track is based on the current state of the tracker. It will not be allowed to create a new track if its distance to the closest target is below a threshold.

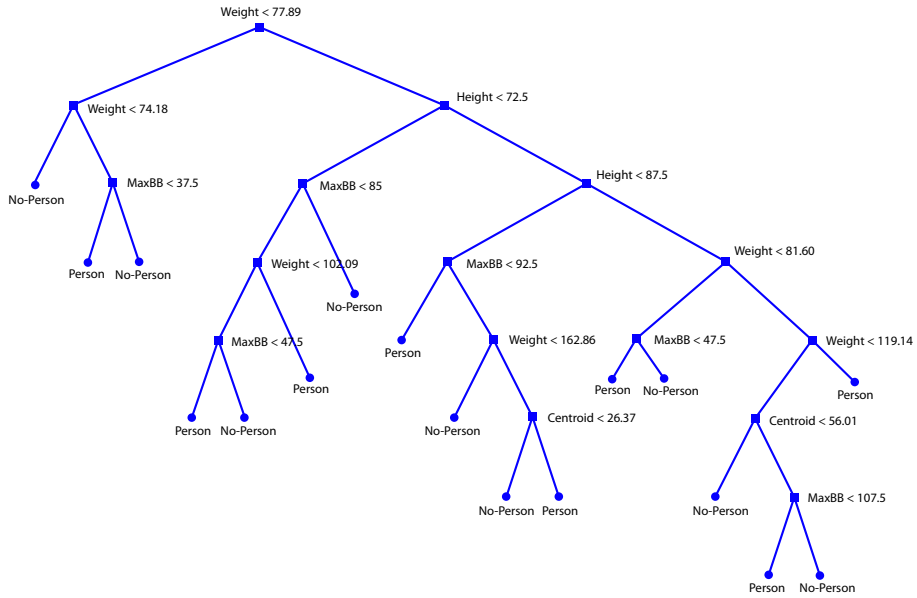


Figure 4.4: *Decision tree employed in the track creation and deletion modules.*

Track deletion criteria

A target will be deleted if one of the following conditions are fulfilled:

- If two or more tracks fall too close to one another, they might be tracking the same target, hence only one will be kept alive while the rest will be removed.

- If tracker's efficiency becomes very low (see §3.1.2.1) it might indicate that the target has disappeared and should be removed.

- The person/no-person classifier is applied to the set of features extracted from the voxels assigned to a target. If the classifier outputs a no-person verdict for a number of frames, the target will be considered as lost.

4.3 Voxel based solutions

The filtering step shown in Figure 4.1 will address the problem of keeping consistent trajectories of the tracked objects, resolving crossings among targets, mergings with spurious objects (i.e. shadows) and producing an accurate estimation of the centroid of the target based on the input voxel information. Although there is a number of papers addressing the problem of multi-camera/multi-person tracking, none has attempted to do it based on a voxel reconstruction.

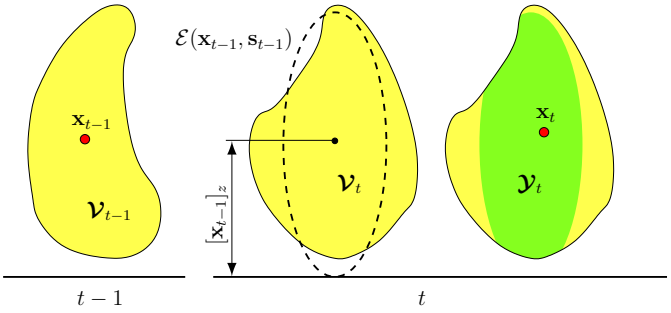


Figure 4.5: *Naïve tracking scheme. For a given object at time t, the region of analysis $\mathcal{Y}_t$ is computed as the intersection of the ellipsoid $\mathcal{E}(\mathbf{x}_{t-1}^k, \mathbf{s}_{t-1})$ with the input data $\mathcal{V}_t$. The centroid at time t is computed as the centroid of the elements of the set $\mathcal{Y}_t$.*

4.3.1 Naïve Tracking

In order to determine a performance baseline in the multi-person tracking task, a naïve approach is devised. Given an estimation of the position of the k-th target of interest at time $t-1$, $\mathbf{x}_{t-1}^k$, it is desired to update this estimation as the new observation $\mathbf{z}_t$ becomes available. For this simple tracking scheme, only the foreground information $\mathcal{V}_t$ will be employed, $\mathbf{z}_t = \mathcal{V}_t$. The region of analysis is defined as the voxel set $\mathcal{Y}_t^k$:

$$\mathcal{Y}_t^k = \mathcal{E}\left(\mathbf{x}_{t-1}^k, \mathbf{s}_{t-1}\right) \cap \mathcal{V}_t, \tag{4.4}$$

that is the portion of foreground voxels enclosed in the region of influence $\mathcal{E}\left(\mathbf{x}_{t-1}, \mathbf{s}_{t-1}\right)$ defined by Eq.4.2. Then, the centroid at time t, $\mathbf{x}_t^k$, is computed as the centroid of the data contained in $\mathcal{Y}_t^k$:

$$\mathbf{x}_t = \frac{1}{|\mathcal{Y}_t^k|} \sum_{\mathcal{V} \in \mathcal{Y}_t^k} \mathcal{V}_{\mathbf{x}}. \tag{4.5}$$

A planar example of this tracking process is depicted in Figure 4.5.

This simple scheme is governed by the parameter $\mathbf{s}_t$ that defines the size of the ellipsoid in Eq.4.4 and is set to be $\mathbf{s}_t = (\alpha s_x, \alpha s_y, \mathbf{x}_{t-1,z})$. Size of axes x and y are set to fit the average perimeter of a person, $s_x = s_y = 30$ cm. In order to account for rapid motion, factor α enlarges the evaluation xy area thus enlarging the search region for the target. However, this election of $\mathbf{s}_t$ will be very susceptible to changes in the z-axis. Eventually, the filter can not cope with this effect leading to a loss of the tracked target and a decrease of the performance. In order to alleviate this effect, the region of influence is enlarged in the z-axis by a scaling factor β, that is $\mathbf{s}_t = (\alpha s_x, \alpha s_y, \beta \mathbf{x}_{t-1,z})$. Setting $\alpha = 1.5$ and $\beta = 1.2$ provided the best experimental results.

It must be noted that the maximum velocity that a target may attain is constrained by the size of the evaluation region and the camera frame rate f_{R} as:

$$s_x = s_y \geq \frac{v_{\max}}{2 f_{\mathrm{R}}}. \tag{4.6}$$

Given a fixed frame rate, the xy radii of the evaluation region will grow together with the expected maximum velocity. Hence, in the case when capturing fast motions with low frame rates, the size of the evaluation region might no longer be anthropomorphically meaningful. Moreover, big sizes in the xy radii may define evaluation regions enclosing more than a single target thus leading to a drop in the performance of the algorithm.

Interaction among trackers is modeled by removing the overlapping area among all the sets $\mathcal{Y}_t^l$, $\forall l \neq k$, intersecting with the tracked k-th set before computing the estimation $\mathbf{x}_t^k$ in Eq.4.5. That is to compute:

$$\widetilde{\mathcal{Y}}_t^k = \mathcal{Y}_t^k - \bigcup_{\substack{l=1 \\ l \neq k}}^{N_\mathrm{T}} \left\{ \mathcal{Y}_t^k \cap \mathcal{Y}_t^l \right\}, \tag{4.7}$$

where N_T is the number of tracked objects at that time.

4.3.2 Particle Filtering Tracking

Particle Filtering (PF) techniques introduced in Chapter 3 are a suitable technique for problems involving multi-modal distributions and problems where data association might be difficult. In the current scenario it is a sound idea to apply the framework of PF to track multiple targets from the obtained 3D observations $\mathbf{z}_t = \{\boldsymbol{\mathcal{V}}_t, \boldsymbol{\mathcal{V}}_t^\mathrm{C}\}$. Surface information, $\boldsymbol{\mathcal{V}}_t^\mathrm{S}$, will not be employed explicitly, although it is employed to compute color histograms of voxels. A particle in this scenario will be an instance of the human body shape, modelled as an ellipsoid $\mathcal{E}_t^j$, centered at $\mathbf{x}_t^j \in \mathbb{R}^3$, with axis size $\mathbf{s}_t^j = (s_x, s_y, [\mathbf{x}_t^j]_z)$. Size of axes, s_x and s_y, are set to $s_x = s_y = 30$ cm as done with the Naïve tracker. The defining parameter of this ellipsoid is its centroid position $\mathbf{x}_t^j$ hence being the state space to be explored by this PF algorithm. Two main factors are to be taken into account when designing a PF system: likelihood evaluation and particle propagation. Moreover, the target interaction model is also a design factor to consider. N_p particles will be employed.

Likelihood evaluation

Binary and color information contained in $\mathbf{z}_t$ will be employed to define the likelihood function $p(\mathbf{z}_t|\ \mathbf{x}_t^j)$ relating the observation $\mathbf{z}_t$ with the human body instance given by particle $\mathbf{x}_t^j$, $1 < j \leq N_\mathrm{p}$. Two partial likelihood functions, $p_\mathrm{Raw}(\boldsymbol{\mathcal{V}}_t|\ \mathbf{x}_t^j)$ and $p_\mathrm{Color}(\boldsymbol{\mathcal{V}}_t^\mathrm{C}|\ \mathbf{x}_t^j)$, will be combined linearly to produce $p(\mathbf{z}_t|\ \mathbf{x}_t^j)$ as:

$$p(\mathbf{z}_t|\ \mathbf{x}_t^j) = \lambda\, p_\mathrm{Raw}(\boldsymbol{\mathcal{V}}_t|\ \mathbf{x}_t^j) + (1-\lambda)\, p_\mathrm{Color}(\boldsymbol{\mathcal{V}}_t^\mathrm{C}|\ \mathbf{x}_t^j). \tag{4.8}$$

Factor λ controls the influence of each term (foreground and color information) in the overall likelihood function. Empirical tests shown that $\lambda = 0.8$ provided satisfactory results. A more detailed review of the impact of color information in the overall performance of the algorithm is addressed in §4.4.2.

Likelihood associated to raw data is defined as the ratio of overlap between the input data $\boldsymbol{\mathcal{V}}_t$ and the ellipsoid $\mathcal{E}_t^j$ defined by particle $\mathbf{x}_t^j$ as

$$p_{\text{Raw}}(\boldsymbol{\mathcal{V}}_t|\ \mathbf{x}_t^j) = \frac{|\boldsymbol{\mathcal{V}}_t \cap \mathcal{E}_t^j|}{|\mathcal{E}_t^j|}. \tag{4.9}$$

For a given target k, an adaptive reference histogram $\mathbf{H}_t^k$ of the colored surface voxels is available. This histogram is constructed using the YCbCr color space. This color space is chosen due to its robustness against light variations, and 21 bins for every channel are employed in the calculations. The color likelihood function is constructed as:

$$p_{\text{Color}}(\boldsymbol{\mathcal{V}}_t^{\text{C}}|\ \mathbf{x}_t^j) = B(\mathbf{H}_t^k, H(\boldsymbol{\mathcal{V}}_t^C \cap \mathcal{E}_t^j)), \tag{4.10}$$

where $B(\cdot)$ is the Bhattacharya distance and $H(\cdot)$ stands for the color histogram extraction operation of the surface voxels of the enclosed volume. Update of the reference histogram is performed in a linear manner following the rule:

$$\mathbf{H}_t^k = \alpha \mathbf{H}_{t-1}^k + (1-\alpha) H(\boldsymbol{\mathcal{V}}_t^C \cap \mathcal{E}_t^{\tilde{x}}), \tag{4.11}$$

where $\mathcal{E}_t^{\tilde{x}}$ stands for the ellipsoid placed in the centroid estimation $\tilde{x}_t$ and α is the adaptation coefficient. In our experiments, $\alpha = 0.9$ provided satisfactory results.

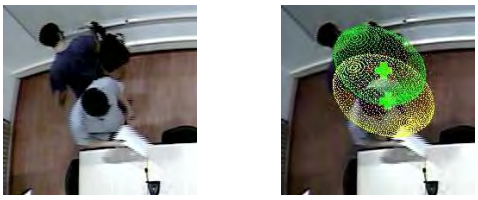
Particle propagation

Propagation model has been chosen to be a Gaussian noise added to the state of the particles after the re-sampling step: $\mathbf{x}_{t+1}^j = \mathbf{x}_t^j + \mathbf{N}$. The covariance matrix $\mathbf{P}$ corresponding to $\mathbf{N}$ is proportional to the maximum variation of the centroid of the target and this information is obtained from the development part of the testing dataset (see §4.4). More sophisticated schemes employ previously learned motion priors to drive the particles more efficiently [KBD03]. However, this would penalize the efficiency of the system when tracking unmodelled motions patterns and, since our algorithm is intended for any motion tracking, no dynamical model is adopted.

Interaction model

The proposed solution for multi-person tracking is to use a split tracker per person together with an interaction model. Let us assume that there are N_{T} independent trackers. Nevertheless, they are not fully independent since each tracker can consider voxels from other targets in both the likelihood evaluation or the 3D re-sampling step resulting in target merging or identity mismatches. In order to achieve the most independent set of trackers, we consider a blocking method to model interactions. Many blocking proposals can be found in 2D tracking related works [KBD03] and we extend it to our 3D case. Blocking methods penalize particles whose associated ellipsoid overlaps with other targets' ellipsoid. Hence, blocking information can be also considered when computing the particle weights for the k-th target as:

$$w_t^{k,j} = p(\mathbf{z}_t|\mathbf{x}_t^{k,j}) \prod_{\substack{l=1 \\ l\neq k}}^{N_{\text{T}}} \phi\left(\tilde{\mathbf{x}}_{t-1}^k, \tilde{\mathbf{x}}_{t-1}^l\right), \tag{4.12}$$



(a) (b)

Figure 4.6: *Particles from the tracker A (yellow ellipsoid) falling into the exclusion zone of tracker B (green ellipsoid) will be penalized by a multiplicative factor $\alpha \in [0, 1]$.*

where $\tilde{\mathbf{x}}_{t-1}^{k}$ stands for the estimation of the PF at time $t-1$ for target k and $\phi(\cdot)$ is the blocking function defining exclusion zones that penalize particles that fall into them. For our particular case, considering that people in the room are always sitting or standing up (this is a meeting room so we assume that they never lay down), this zone can be constrained to the xy plane. The proposed function was:

$$\phi\left(\tilde{\mathbf{x}}_{t-1}^{k}, \tilde{\mathbf{x}}_{t-1}^{l}\right) = 1 - \exp\left(-k \left\| \left[\tilde{\mathbf{x}}_{t}^{k}\right]_{x,y} - \left[\tilde{\mathbf{x}}_{t}^{l}\right]_{x,y} \right\|^{2}\right), \tag{4.13}$$

where $k \propto s_x^{-2}$ is the parameter that drives the sensibility of the exclusion zone.

4.3.3 Sparse Sampling Tracking

The presented PF approach to tracking defines a set of instances of the position of the tracked person, the particles, and a formulation to measure the fitness of these hypothesis with relation to the observable data. However, the evaluation of this likelihood function may be computationally expensive, as will be further proved in §4.4.3. Moreover, data is usually noisy and may contain merged blobs corresponding to different targets. Sparse sampling (SS) technique is presented as an efficient and flexible alternative to PF.

Assuming an homogeneous 3D object, it can be seen intuitively that its centroid can be exactly computed based only on the surface voxels since the interior voxels do not provide any relevant information. This computation is done through a discrete version of Green's theorem on the surface voxels [Leu91, CM98] while other approaches obtain an accurate approximation of the centroid using corner points (see [YA96] for a review). A common assumption of these techniques is the availability of surface data extracted beforehand hence a labelling of the voxels in the scene is available too. If we further assume that our object presents a radial symmetry in the xy plane, the computation of the centroid can be done as an average of the positions of the surface voxels:

$$\tilde{\mathbf{x}}_t = \frac{\sum\limits_{\mathcal{V} \in \boldsymbol{\mathcal{V}}_t} \mathcal{V}_{\mathbf{x}}}{|\boldsymbol{\mathcal{V}}_t|} = \frac{\sum\limits_{\mathcal{V} \in \boldsymbol{\mathcal{V}}_t^{\mathrm{S}}} \mathcal{V}_{\mathbf{x}}}{|\boldsymbol{\mathcal{V}}_t^{\mathrm{S}}|}. \tag{4.14}$$

Let us model the human body as an ellipsoid, as previously done in the Naïve and PF approaches. In order to test the robustness of the centroid computation in Eq.4.14 against

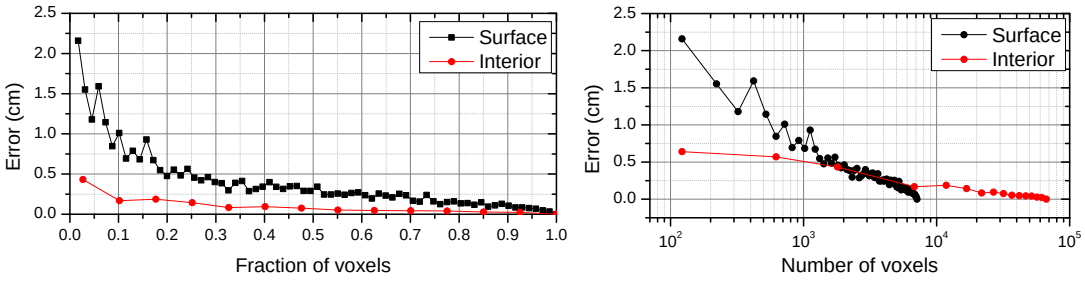


Figure 4.7: *Centroid's estimation error when computed with a fraction of surface or interior voxels. The employed ellipsoid had a radii* $\mathbf{s} = (30, 30, 100)$ *and voxels with* $s_{\mathcal{V}} = 2$ *cm where used.*

missing data, we studied the committed error when only a fraction of these input data is employed. A number of voxels (surface or interior voxels in each case) is randomly selected and employed to compute the centroid. Then, the error is computed showing that the surface based estimation is more sensitive than the estimation using interior voxels (see Figure 4.7). However, it proves that the centroid can be computed from a number of randomly selected surface voxels still achieving a satisfactory performance as it will be shown at the end of this section. This idea is the underlying principium of the SS algorithm.

Imagine that we would like to estimate the centroid of an object by analyzing a randomly selected number of voxels from the whole scene, denoted as $\boldsymbol{\mathcal{W}}$. An approach to the computation of the centroid would be:

$$\tilde{\mathbf{x}}_t \approx \frac{\sum\limits_{\mathcal{W}\in\boldsymbol{\mathcal{W}}_t} \rho\left(\mathcal{W}\right)\mathcal{W}_{\mathbf{x}}}{\sum\limits_{\mathcal{W}\in\boldsymbol{\mathcal{W}}_t} \rho\left(\mathcal{W}\right)}, \qquad \rho\left(\mathcal{W}\right) = \begin{cases} 1 & \text{if } \mathcal{W} \in \boldsymbol{\mathcal{V}}_t \\ 0 & \text{if } \mathcal{W} \notin \boldsymbol{\mathcal{V}}_t \end{cases}, \tag{4.15}$$

where $\rho(\mathcal{W})$ gives the mass density at voxel $\mathcal{W}$. Since it is assumed that all voxels have the same mass, this is a binary function that checks the occupancy of a given voxel. Hence, only the fraction of (randomly selected) voxels inside the object will contribute to the computation of the centroid. Eq.4.15 can be rewritten as:

$$\tilde{\mathbf{x}}_t \approx \sum_{\mathcal{W}\in\boldsymbol{\mathcal{W}}_t} \frac{\rho\left(\mathcal{W}\right)}{\sum\limits_{\mathcal{W}\in\boldsymbol{\mathcal{W}}_t} \rho\left(\mathcal{W}\right)}\mathcal{W}_{\mathbf{x}} = \sum_{\mathcal{W}\in\boldsymbol{\mathcal{W}}_t} \tilde{\rho}\left(\mathcal{W}\right)\mathcal{W}_{\mathbf{x}}, \tag{4.16}$$

where $\tilde{\rho}\left(\mathcal{W}\right)$ can be considered as the normalized mass contribution of voxel $\mathcal{W}$ to the computation of the centroid. If function $\rho\left(\mathcal{W}\right)$ takes values in the range $[0, 1]$ we may consider it as the *"degree of mass"* of $\mathcal{W}$ or the importance of voxel $\mathcal{W}$ into the calculation of $\tilde{\mathbf{x}}_t$. Then, $\rho\left(\mathcal{W}\right)$ might be considered as a normalized weight assigned to $\mathcal{W}$. Since we stated that the centroid can be computed using surface voxels, Eq.4.14 can be also posed as:

$$\tilde{\mathbf{x}}_t \approx \frac{\sum\limits_{\mathcal{W}\in\boldsymbol{\mathcal{W}}_t} \rho_{\mathrm{S}}\left(\mathcal{W}\right)\mathcal{W}_{\mathbf{x}}}{\sum\limits_{\mathcal{W}\in\boldsymbol{\mathcal{W}}_t} \rho_{\mathrm{S}}\left(\mathcal{W}\right)} = \sum_{\mathcal{W}\in\boldsymbol{\mathcal{W}}_t} \frac{\rho_{\mathrm{S}}\left(\mathcal{W}\right)}{\sum\limits_{\mathcal{W}\in\boldsymbol{\mathcal{W}}_t} \rho_{\mathrm{S}}\left(\mathcal{W}\right)}\mathcal{W}_{\mathbf{x}} = \sum_{\mathcal{W}\in\boldsymbol{\mathcal{W}}_t} \tilde{\rho}_{\mathrm{S}}\left(\mathcal{W}\right)\mathcal{W}_{\mathbf{x}}, \tag{4.17}$$

where $\rho_{\mathrm{S}}(\mathcal{W}) \in [0,1]$ measures the *"degree of being surface"* of voxel $\mathcal{W}$. Within this context, functions $\rho(\cdot)$ and $\rho_{\mathrm{S}}(\cdot)$ might be understood as pseudo-likelihood functions and Eq.4.17 and 4.16 as a sample based representation of an estimation problem. There is an obvious similarity between this representation and the formulation of particle filters but there is a significant difference. While particles in PF represent an instance of the whole state space, our samples ($\mathcal{W} \in \boldsymbol{\mathcal{W}}_t$) are points in the 3D space. Moreover, particle likelihoods are computed over all data while sample pseudo-likelihoods will be computed in a local domain.

The presented concepts are applied to define the Sparse Sampling (SS) algorithm. Let $\mathbf{y}_t^i \in \mathbb{R}^3$, a point in the 3D space and $\omega_t^i \in \mathbb{R}$ its associated weight measuring the likelihood of this position being part of the object or part of its surface. Under certain assumptions, it is achieved that the centroid can be computed as:

$$\tilde{\mathbf{x}}_t \approx \sum_{i=1}^{N_{\mathrm{s}}} \omega_t^i \mathbf{y}_t^i, \tag{4.18}$$

where N_{s} is the number of sampling points. When using SS we are no longer sampling the state space since $\mathbf{y}_t^i$ can not be considered an instance of the centroid of the target as happened with particles, $\mathbf{x}_t^j$, in PF. Hence, we will talk about *samples* instead of *particles* and we will refer to $\{(\mathbf{y}_t^i, \omega_t^i)\}_{i=1}^{N_{\mathrm{s}}}$ as the sampling set. This set will approximate the surface of the k-th target, $\boldsymbol{\mathcal{V}}^{\mathrm{S},k}$, and will fulfill the sparsity condition $N_{\mathrm{s}} \ll |\boldsymbol{\mathcal{V}}^{\mathrm{S},k}|$.

In order to define a method to recursively estimate $\tilde{\mathbf{x}}_t$ from the sampling set $\{(\mathbf{y}_t^i, \omega_t^i)\}_{i=1}^{N_{\mathrm{s}}}$, a filtering strategy has to be set. Essentially, the proposal is to follow the PF analysis loop (re-sampling, propagation, evaluation and estimation) with some opportune modifications to ensure the convergence of the algorithm.

One of the advantages of the SS algorithm is its computational efficiency. The complexity to compute $p(\mathbf{z}_t|\ \mathbf{y}_t^i)$ is quite reduced since it only evaluates a local neighborhood around the sample in comparison with the computational load required to evaluate the likelihood, $p(\mathbf{z}_t|\ \mathbf{y}_t^i)$, of a particle in the PF algorithm. This point will be quantitatively addressed in §4.4.3.

Pseudo-Likelihood evaluation

Associated weight ω_t^i to a sample $\mathbf{y}_t^i$ will measure the likelihood of that 3D position to be part of the surface of the tracked target. When computing the pseudo-likelihood, surface has been chosen instead of interior voxels, based on the efficiency of surface samples to propagate rapidly as it will be explained in the next subsection. As in the defined PF likelihood function, two partial likelihood functions, $p_{\mathrm{Raw}}(\boldsymbol{\mathcal{V}}_t|\ \mathbf{y}_t^i)$ and $p_{\mathrm{Color}}(\boldsymbol{\mathcal{V}}_t^{\mathrm{C}}|\ \mathbf{y}_t^i)$, are linearly combined to form $p(\mathbf{z}_t|\ \mathbf{y}_t^i)$ as:

$$p(\mathbf{z}_t|\ \mathbf{y}_t^i) = \lambda\ p_{\mathrm{Raw}}(\boldsymbol{\mathcal{V}}_t|\ \mathbf{y}_t^i) + (1-\lambda)\ p_{\mathrm{Color}}(\boldsymbol{\mathcal{V}}_t^{\mathrm{C}}|\ \mathbf{y}_t^i). \tag{4.19}$$

Partial likelihoods will be computed on a local domain centered in the position $\mathbf{y}_t^i$. Let $\mathcal{C}(\mathbf{y}_t^i, q, r)$ be a neighborhood of radius r over a connectivity q domain on the 3D orthogonal grid around a sample place in a voxel position $\mathbf{y}_t^i$. Then, we define the occupancy

and color neighborhoods around $\mathbf{y}_t^i$ as $\mathbf{O}_t^i = \boldsymbol{\mathcal{V}}_t \cap \mathcal{C}(\mathbf{y}_t^i, q, r)$ and $\mathbf{C}_t^i = \boldsymbol{\mathcal{V}}_t^{\mathsf{C}} \cap \mathcal{C}(\mathbf{y}_t^i, q, r)$, respectively.

For a given sample i occupying a voxel, its weight associated to the raw data will measure its likelihood to belong to the surface of an object. It can be modeled as:

$$p_{\text{Raw}}(\boldsymbol{\mathcal{V}}_t |\, \mathbf{y}_t^i) = 1 - \left| \frac{2|\mathbf{O}_t^i|}{|\mathcal{C}(\mathbf{y}_t^i, q, r)|} - 1 \right|. \tag{4.20}$$

Ideally, when the sample $\mathbf{y}_t^i$ is placed in a surface, half of its associated occupancy neighborhood will be occupied and the other half empty. The proposed expression attains its maximum when this condition is fulfilled. Although this likelihood can be computed using the surface voxel data, $\boldsymbol{\mathcal{V}}_t^{\mathsf{S}}$, this set tends to be noisy hence not suitable for this computation.

Function $p_{\text{Color}}(\boldsymbol{\mathcal{V}}_t^{\mathsf{C}} |\, \mathbf{y}_t^i)$ can be defined as the likelihood of a sample belonging to the surface corresponding to the k-th target characterized by an adaptive reference color histogram $\mathbf{H}_t^k$:

$$p_{\text{Color}}(\boldsymbol{\mathcal{V}}_t^{\mathsf{C}} |\, \mathbf{y}_t^i) = D(\mathbf{H}_t^k, \mathbf{C}_t^j). \tag{4.21}$$

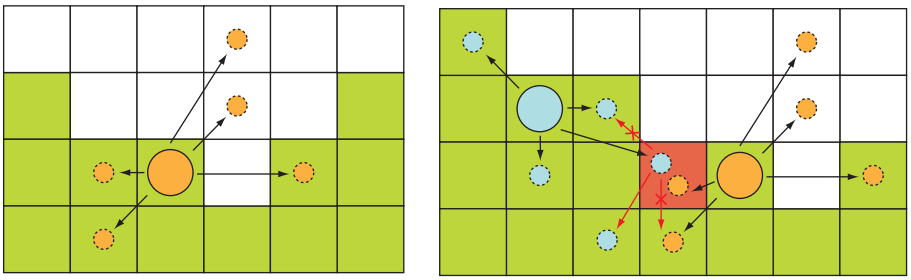
Since $\mathbf{C}_t^j$ contains only local color information with reference of the global histogram $\mathbf{H}_t^k$, the distance $D(\cdot)$ is constructed towards giving a measure of the likelihood between this local colored region and $\mathbf{H}_t^k$. For every voxel in $\mathbf{C}_t^j$, it is decided whether it is similar to $\mathbf{H}_t^m$ by selecting the histogram value for the tested color and checking whether it is above a threshold γ or not. Finally, the ratio between the number of similar color and total voxels in the neighborhood gives the color similarity score. Since reference histogram is updated and changes over time, a variable threshold γ is computed so that the 80% of the values of $\mathbf{H}_t^m$ are taken into account.

The parameters defining the neighborhood were set to $q = 26$ and $r = 2$ yielding to satisfactory results. Larger values of the radius r did not significantly improve the overall algorithm performance but increased its computational complexity.

Sample propagation and 3D discrete resampling

A sample $\mathbf{y}_t^i$ placed near a surface will have an associated weight ω_t^j with a high value. It is a valid assumption to consider that some surrounding positions might also be part of this surface. Hence, placing a number of new samples in the vicinity of $\mathbf{x}_t^j$ would contribute to progressively explore the surface of a voxel set. This idea leads to the spatial re-sampling and propagation scheme that will drive samples along time to place samples in the surface of the tracked target.

Given the discrete nature of the 3D voxel space, it will be assumed that every sample is constrained to occupy a single voxel or discrete 3D coordinate and there can not be two samples placed in the same location. Re-sampling method is mimicked from particle filtering so a number of replicas proportional to the normalized weight of the sample are generated. Then, these new samples are propagated and some *discrete* noise is added to their position meaning that their new positions are also constrained to occupy a discrete 3D coordinate (see an example in Figure 4.8(a)). However, two resampled and propagated particles may fall in the same 3D voxel location as shown in Figure



(a) (b)

Figure 4.8: *Example of discrete re-sampling and propagation (in 2D). In (a), a sample is re-sampled and its replicas are randomly placed occupying a single voxel. In (b), two re-sampled samples falls in the same position (red cell) and one of them (blue) performs a random search through the adjacent voxels to find an empty location.*

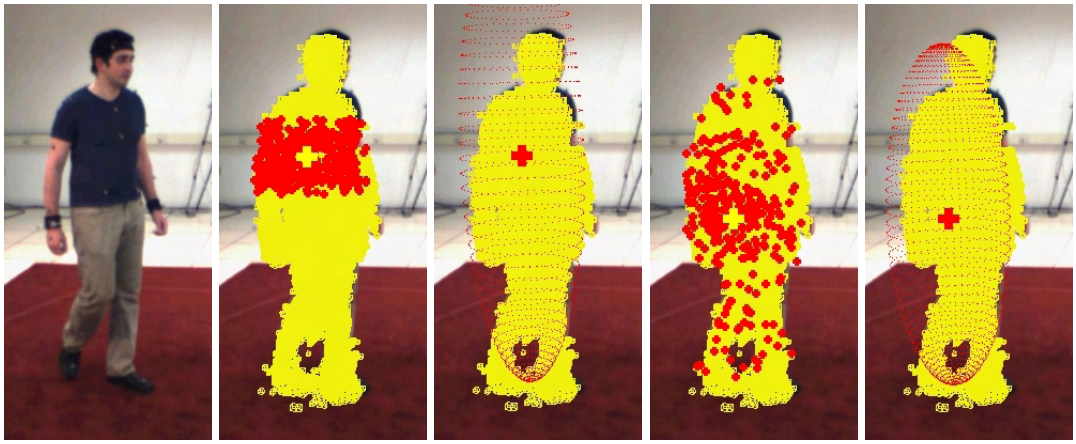
4.8(b). In such case, one of these particles will randomly explore the adjacent voxels until reaching an empty location; if there is not any suitable location for this particle, it will be dismissed.

The choice of sampling the surface voxels of the object instead of its interior voxels to finally obtain its centroid is motivated by the fact that propagating samples along the surface rapidly spread them all around the object as depicted in Figure 4.9. Propagating samples on the surface is equivalent to propagate them on a 2D domain, hence the condition of not placing two samples in the same voxel will make them to explore the surface faster (see Figure 4.9(c)). On the other hand, interior voxels propagate on a 3D domain thus having more space to explore and therefore becoming slower to spread all around the volume (see Figure 4.9(b)). Although both (pseudo-)likelihoods should produce a fair estimation of the object's centroid as explained in §4.3.3, both sampling sets must fulfill the condition to be randomly spread around the object volume, otherwise the centroid estimation will be biased.

Interaction model

The flexibility of a sample based analysis may, sometimes, lead to situations where particles spread out too much from the computed centroid. In order to cope with this problem, a intra-target samples interaction model is devised. If a sample is placed in a position such that $\|[\mathbf{y}_t^i]_{x,y} - [\tilde{\mathbf{x}}_{t-1}]_{x,y}\| > \delta$ it will be removed (that is to assign $\omega_t^i = 0$) and we set the threshold as $\delta = \alpha s_x$, with $s_x = 30$ cm. Towards a fair comparison with the Naïve algorithm, we set $\alpha = 1.5$.

The interaction among targets is modeled in similar way as in the PF approach. Formulas in Eq.4.12 and 4.13 are applied to samples with the appropriate scaling parameter k.



(a) Reference image (b) Interior based likelihood (c) Surface based likelihood

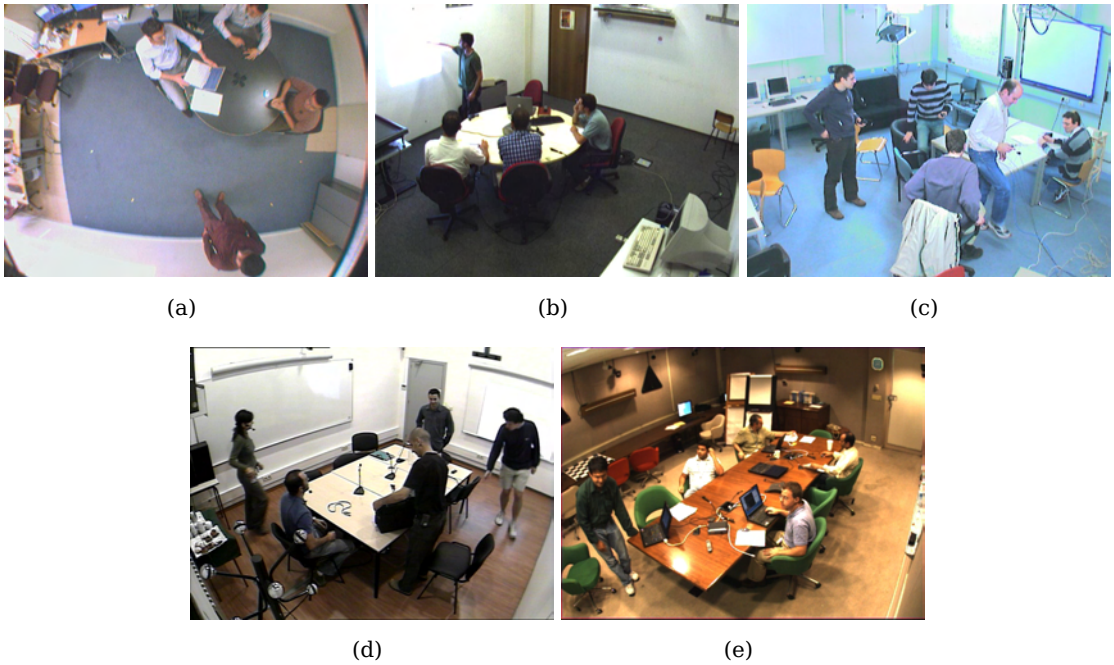
Figure 4.9: *Sample positions evolution when using a likelihood based on the interior (a) and surface (b) voxels.*

4.4 Results and Evaluation

In order to assess the performance of the proposed tracking systems, they have been tested on the set of benchmarking image sequences provided by the CLEAR Evaluation Campaigns 2006 and 2007 [CLE07] and a range of our own captured image sequences under several scenarios. Typically, these evaluation sequences involved up to 5 people moving around in a meeting room. This benchmarking set was formed by two separate data sets, development and evaluation, containing sequences recorded by 5 of the participating partners[1]. A sample of these data can be seen in Figure 4.10. The development set consisted in 5 sequences of an approximate duration of 20 minutes each, while the evaluation set was formed by 40 sequences of 5 minutes each, thus adding up to 5 hours of data. Each sequence was recorded with 4 cameras placed in the corners of the SmartRoom and a zenital camera placed in the ceiling. All cameras were calibrated and had resolutions ranging from 640x480 to 756x576 pixels at an average frame rate of $f_{\mathrm{R}} = 25$ fps. The test environments was a 5x4 m room with occluding elements such as tables and chairs. Images of the empty room were also provided to train the background/foreground segmentation algorithms.

In order to obtain the most statistically meaningful evaluation results, it is important for the dataset to provide enough instances and a rich variation of each event to be detected. With this aim, each sequence was recorded following a precise script including all possible situation an algorithm may encounter including a number of crossings among the moving people in the room, complex motion patterns, etc.

[1]These partners were: Athens Information Technology (AIT), Instituto Trentino di Cultura (ITC), University of Karlsruhe (UKA), Technical University of Catalonia (UPC) and IBM.



(a) (b) (c)

(d) (e)

Figure 4.10: *CLEAR [CLE07] evaluation dataset sample. Images from several partners showing a common indoor conference room configuration involving several participants.*

4.4.1 Evaluation metrics

Metrics proposed in [BES06] for multi-person tracking evaluation have been adopted. These metrics, being used in international evaluation contests [CLE07] and adopted by several research projects such as the European CHIL [CHI07] or the U.S. Vace [VAC] allow objective and fair comparisons with other methods.

This evaluation process assumes that for every time frame t, a multi-person tracker will output a set of hypotheses for the set of targets (persons). Ground truth information containing the position of the center of the head of every person in the room is available. It is assumed that relevant tracking information to be used by further analysis modules is usually found in the xy plane. Hence, during the evaluation process, z coordinate of both the hypotheses and the ground truth positions is disregarded and the errors metrics are computed only on the xy plane. The evaluation procedure comprises the following steps:

1. Establish the best possible correspondence between hypotheses produced by the tracker and ground truth positions.

2. For each found correspondence, compute the error in the object's position estimation.

3. Accumulate all correspondence errors. This includes the following:

   (a) count all objects for which no hypothesis was output as misses;

(b) count all tracker hypotheses for which no real object exists as false positives;

(c) count all identity exchanges among correctly tracked objects as mismatches.

According to these criteria, two metrics are defined. First, the **M**ultiple **O**bject **T**racking **P**recision (*MOTP*), which shows tracker's ability to estimate precise object positions. *MOTP* is defined as the total position error for matched object-hypothesis pairs over all frames, averaged by the total number of matches:

$$MOTP = \frac{\sum_{i,t} d_{i,t}}{\sum_t c_t}, \tag{4.22}$$

where c_t is the number of matches found for time frame t, and $d_{i,t}$ is the distance between the ground truth and its corresponding hypothesis. Evidently, *MOTP* is a distance metric, i.e., we express it in millimeters and the smaller the value, the better the performance. It must be noted that *MOTP* shows the ability of the tracker to estimate precise object positions, independent of its skill at recognizing object configurations, keeping consistent trajectories, and so forth.

There are three more kinds of errors to be accounted when evaluating tracking performance: misses, false positives and mismatches. These are jointly reported in the second accuracy metric, namely the **M**ultiple **O**bject **T**racking **A**ccuracy (*MOTA*), which expresses the tracker's performance at estimating the number of objects, and at keeping consistent trajectories. The *MOTA* is defined as the residual of the sum of these three error rates from unity:

$$MOTA = 1 - \frac{\sum_t m_t + \sum_t fp_t + \sum_t mme_t}{\sum_t g_t}, \tag{4.23}$$

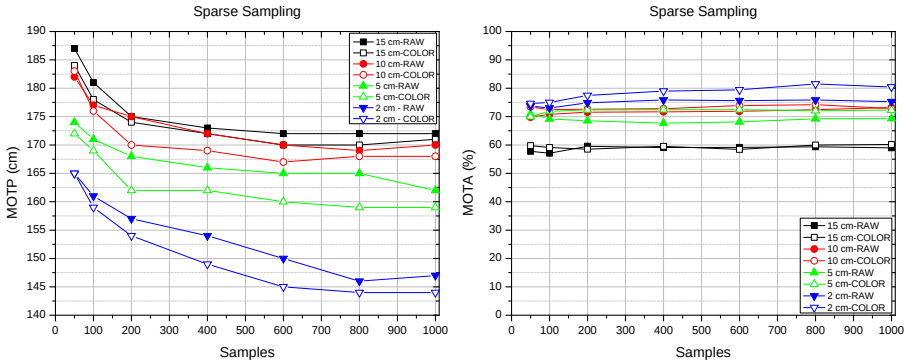
where m_t, fp_t and mme_t are the number of misses, false positives and mismatches, respectively, and g_t denotes the number of objects, at time t. The *MOTA* score is composed of 3 error ratios in the sequence: (1) the ratio of misses, $\overline{m} = \sum_t m_t / \sum_t g_t$, (2) the ratio of false positives, $\overline{fp} = \sum_t fp_t / \sum_t g_t$, and (3) the ratio of mismatches, $\overline{mme} = \sum_t mme_t / \sum_t g_t$, computed over the total number of objects $\sum_t g_t$ presented in all frames. The *MOTA* accounts for all object configuration errors made by the tracker over all frames.

The aim of a tracking system would be to produce high values of *MOTA* and low values of *MOTP* thus indicating its ability to correctly track all targets and estimate their positions accurately. When comparing two algorithms, there will be a preference to choose the one outputting the highest *MOTA* score.

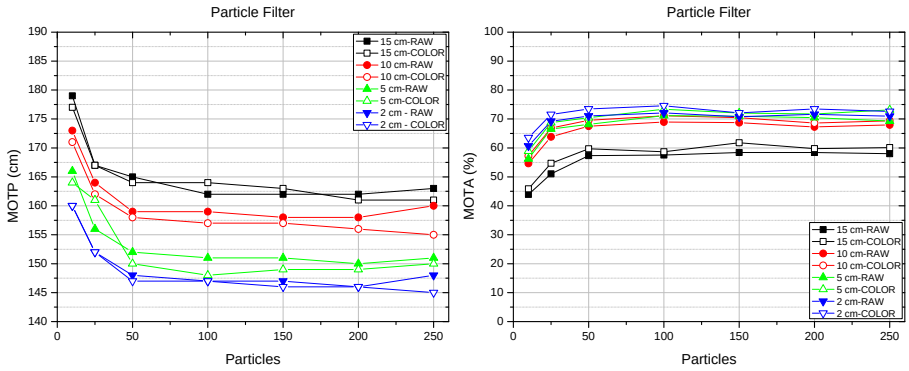
4.4.2 Results

To demonstrate the effectiveness of the proposed multi-person tracking approaches, a set of experiments were conducted over the CLEAR 2007 database. The development part of the dataset was used to train the creation/deletion of tracks modules as described in §4.2.3 and the remaining test part was used for our experiments.

First, the multi-camera data is pre-processed performing the foreground/background segmentation and 3D voxel reconstruction algorithms in §2.2.3 and §2.2.4. In order to



(a) Sparse Sampling



(b) Particle Filtering

Figure 4.11: MOTP *and* MOTA *scores for the Sparse Sampling (SS) and the Particle Filtering (PF) techniques using raw and colored voxels. Several voxel sizes* $s_{\mathcal{V}} = \{2, 5, 10, 15\}$ *cm have been used.*

analyze the dependency of the tracker's performance with the resolution of the 3D reconstruction, several voxel sizes were employed $s_{\mathcal{V}} = \{2, 5, 10, 15\}$ cm. A colored version of these voxel reconstructions was also generated, according to the technique introduced in §2.2.4. Then, these data was the input fed to the Naïve, Sparse Sampling (SS) and Particle Filtering (PF) proposed approaches.

Naïve tracking achieved the lowest performance due to its limited ability to explore the 3D space. However, this technique contains the seminal ideas of the region of influence (ellipsoid) and the exclusion zones based target interaction that are further extended in the PF and SS approaches.

In both types of filters, SS or PF, three parameters drive the performance of the algorithm: the voxel size $s_{\mathcal{V}}$, the number of samples N_{s} or particles N_{p}, and the usage of color information. Experiments carried out explore the influence of these two parameters in the *MOTP* and *MOTA* shown in Figure 4.11 and Table 4.3. Some remarks can be drawn:

- **Number of samples/particles:** There is a dependency between the *MOTP* score and the number of particles/samples, specially for the SS algorithm. The contribu-

		$s_{\mathcal{V}} = 15$ cm		$s_{\mathcal{V}} = 10$ cm		$s_{\mathcal{V}} = 5$ cm		$s_{\mathcal{V}} = 2$ cm	
	N	*MOTP*	*MOTA*	*MOTP*	*MOTA*	*MOTP*	*MOTA*	*MOTP*	*MOTA*
Naïve	-	190	40.15	189	47.87	185	52.18	179	53.51
SS-Raw	50	187	57.78	182	69.81	174	70.91	165	73.75
	100	181	57.09	177	70.82	171	69.18	161	73.01
	200	175	59.55	175	71.54	168	68.54	157	74.86
	400	173	59.08	172	71.71	166	67.69	154	75.87
	600	172	59.16	170	71.87	165	68.14	150	75.65
	800	172	59.35	169	72.44	165	69.24	**146**	**75.87**
	1000	**172**	**59.99**	**170**	**73.19**	**162**	**69.31**	147	75.28
SS-Color	50	184	59.78	183	73.44	172	70.01	165	74.62
	100	178	59.09	176	72.47	169	71.88	159	74.99
	200	174	58.53	170	72.62	162	72.38	154	77.47
	400	172	59.52	169	72.77	162	72.44	149	78.98
	600	170	58.43	167	73.86	**160**	**72.47**	145	79.43
	800	170	59.98	**168**	**74.23**	159	72.07	**144**	**81.50**
	1000	**171**	**60.17**	168	72.84	159	72.3	144	80.45
PF-Raw	10	179	43.62	173	54.64	166	56.21	160	60.66
	25	167	51.05	164	63.82	156	66.56	152	69.20
	50	165	57.33	159	67.52	152	68.13	148	71.02
	100	162	57.54	**159**	**68.96**	**151**	**71.18**	**147**	**72.10**
	150	**162**	**58.40**	158	68.71	151	70.95	147	70.76
	200	162	58.40	158	67.24	150	70.48	146	71.62
	250	163	57.99	160	67.94	151	69.28	148	70.97
PF-Color	10	177	45.86	171	57.25	164	58.91	160	63.54
	25	167	54.70	162	66.86	161	68.75	152	71.56
	50	164	59.71	158	69.52	150	70.42	147	73.46
	100	164	58.68	**157**	**71.11**	**148**	**73.35**	**147**	**74.56**
	150	**163**	**61.78**	157	70.44	149	72.04	146	72.09
	200	161	59.75	156	68.61	149	71.71	146	73.44
	250	161	60.10	155	69.38	150	73.13	145	72.61

Table 4.3: *Quantitative evaluation:* MOTP *and* MOTA *results for the Sparse Sampling and Particle Filtering algorithms presented using raw and color input data. Best case of each experiment is written in bold cases.*

tion of a new sample to the estimation of the centroid in the SS has less impact than the addition of a new particle in the PF, hence the slightly slower decay of the *MOTP* curves for the SS than for the PF. Regarding the *MOTA* score, there is not a significant dependency with N_{s} or N_{p}. Two factors drive the *MOTA* of an algorithm: the track creation/deletion modules, that mainly contributes to the ratio of misses and false positives in Eq.4.23, and the filtering step itself that has an impact to the mismatches ratio in the same equation. The low dependency of *MOTA* with N_{s} or N_{p} shows that the most of the influence in this score does not depend on the filtering technique employed but in the track creation/deletion modules. This assumption was validated by testing several classification methods in the creation/deletion modules yielding to a drop in the *MOTA* score proportional to their ability to correctly classify a blob as person/no-person.

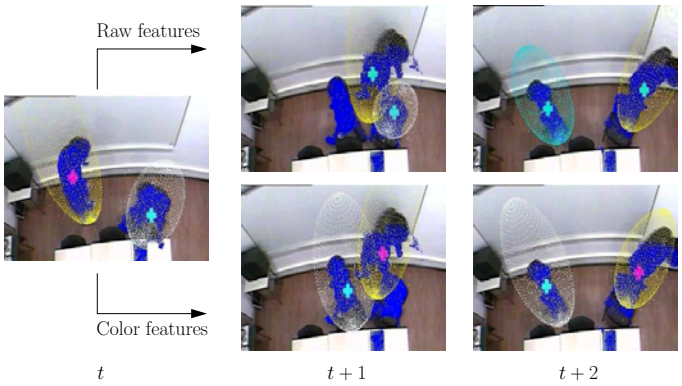


Figure 4.12: *Zenital view of two comparative experiments showing the influence of color in the SS algorithm. The cross-over between two targets is correctly tackled when using color information whereas using only raw features leads to a mismatch and, afterwards, a track loss (white ellipsoid) and the creation of a new one (cyan ellipsoid).*

- **Voxel size:** Scenes reconstructed with a large voxel size do not capture well all spatial details and may miss some objects thus decreasing the performance of the system (both in SS and PF). It can be observed that *MOTP* and *MOTA* scores improve as the voxel size decrease.

- **Color features:** Color information improves the performance of SS and PF in both *MOTP* and *MOTA* scores. First, there is an improvement when using color information for a given voxel size, specially for the SS algorithm. Moreover, the smaller the voxel size the most noticeable difference between the experiments using raw and color features. This effect is supported by the fact that color characteristics are better captured when using small voxel sizes. The performance improvement when using color in the SS algorithm is more noticeable since samples are placed in the regions with a high likelihood to be part of the target. For instance, this effect is more evident in cases where the subject is sitting and the particles concentrate in the upper body part, disregarding the part of the chair. In the SS algorithm, *MOTP* score benefits from this efficient sample placement. PF algorithm is constrained to evaluate the color likelihood in the ellipsoid defined in Eq.4.10 thus not being able to differentiate between parts of the blob that do not belong to the tracked target. Color information used within the filtering loop leads to a better distinguishability among blobs thus reducing the mismatch ratio and slightly improving the *MOTA* score. Merging of adjacent blobs or complex crossing among targets are also correctly resolved. An example of the impact of color information is shown in Figure 4.12 where the usage of color avoids the mismatch between two targets. This effect is more noticeable when targets in the scene are dressed in different colors.

We can compare the results obtained by SS and PF with other algorithms evaluated using the same CLEAR 2007 database whose scores are reported in Table 4.4. Most of these methods exploited multi-view information with the exception of [BGS07] that only

Method	*MOTP* (mm)	*MOTA* (%)	*FP* (%)	*Miss* (%)	*MM* (cases)
Face detection+Kalman filtering [KTPP07]	91	59.66	06.99	30.89	2.46
Appearance models+Particle filtering [LCB07]	141	59.62	18.58	20.66	1.14
Upper body detection+Particle filtering [BGS07]	155	69.58	14.50	15.09	0.83
Zenital camera analysis+Particle filtering [BGS07]	222	54.94	20.24	23.74	1.08
Voxel analysis+Heuristic tracker [CFSC07a]	168	30.49	40.19	27.74	1.58
Voxel analysis+Naïve filtering (best case)	179	53.51	18.63	22.05	18.63
Voxel analysis+Particle filtering (best case)	147	74.56	14.03	10.48	0.91
Voxel analysis+Sparse sampling (best case)	144	81.50	09.34	08.70	0.46

Table 4.4: *Results presented at the CLEAR 2007 [CLE07] by several partners. Multi-camera information is used to track multiple people using several methods*

used the zenital camera facing the associated distortion and perspective problems. Particle filtering is the most employed technique due to its suitability to the characteristics of this problem although Kalman filtering used by [KTPP07] provided fair results when fed by higher semantical features extracted from the input data (in this case, faces). Note the low *FP* score for this system as a consequence of the unlikely event of detecting a face in a spurious object. A 3D voxel reconstruction was used as the input data in [CFSC07a] together with a simple track management system. The rest of the methods [LCB07, BGS07] relied on a fixed human body appearance model similar to the ellipsoidal region of interest used in our PF proposal. However, the novelty of these methods are the strategies to combine the information coming from the analysis of different views without performing any 3D reconstruction. Comparing the best proposed tracking system[2] [BGS07] with our two approaches, we obtain a relative improvement of $\Delta(MOTP, MOTA)_{\mathrm{SS}} = (7.63, 17.13)\%$ and $\Delta(MOTP, MOTA)_{\mathrm{PF}} = (5.16, 7.15)\%$. The improvement in the *MOTA* score is mainly motivated by the statistical analysis of the scene when creating and deleting tracks.

4.4.3 Computational performance

Comparing obtained metrics among different algorithms can give an idea about their performance in an scenario where computational complexity is not taken into account. For example, if an algorithm requires an enormous computational load to attain a good performance, this might render it unsuitable for some applications. An analysis of the operation time of several algorithms under the same conditions and the produced *MOTP*/*MOTA* metrics might give a more informative and fairer comparison tool. Although there is not a standard procedure to measure the computational performance of a tracking process, we devised a method to assess the computational efficiency of our algorithms to present a comparative study.

[2] When selecting the best system, the *MOTA* score is regarded as the most significant value.

Measurement procedure

The overall time required by a tracking algorithm based on the scheme depicted in Figure 4.1 to analyze and process a new input data set $\mathbf{z}_t$ is:

$$\Delta T_{\mathbf{z}_t} = T_{\text{Input}} + T_{\text{New Track}} + T_{\text{Filter}} + T_{\text{Delete Track}}, \tag{4.24}$$

where T_{Input} is the time required to generate the input data, $T_{\text{New Track}}$ and $T_{\text{Delete Track}}$ are the time required by the creation/deletion classifiers and T_{Filter} is the time required by the filter to update the state variables. For a given sequence S, the frames-per-second (*FPS*) measure of a given algorithm can be computed as:

$$FPS_S \approx \frac{N_{\text{F}}^S}{\sum_t \Delta T_{\mathbf{z}_t}}, \tag{4.25}$$

where N_{F}^S is the overall number of frames in S and $\Delta T_S = \sum_t \Delta T_{\mathbf{z}_t}$ is the elapsed execution time required by the tracking algorithm to process the sequence S. A more robust (and realistic) measure would be the *FPS* score computed over all sequences as

$$FPS \approx \frac{\sum_i N_{\text{F}}^{S_i}}{\sum_i \Delta T_{S_i}}. \tag{4.26}$$

This measure, being an average over all sequences, reduces the errors introduced in the measure process and allows comparing the computational load required by the proposed algorithms. A more intuitive score is the Real-Time Factor (*RTF*), defined as

$$RTF = \frac{FPS}{f_{\text{R}}}, \tag{4.27}$$

where $f_{\text{R}} = 25$ is the average sequence frame-rate.

When measuring the lapse ΔT_{S_i} some issues must be taken into account to give an accurate measure towards a fair comparison:

- Computer load during the execution of a tracking experiment might affect the lapse measure. This load depends on the number of internal processes executed by the computer. The fraction of computer's CPU resources dedicated to the tracking experiment, denoted as α_{CPU}, is used to normalize the time measure:

$$\Delta T'_{S_i} = \alpha_{\text{CPU}} \Delta T_{S_i}. \tag{4.28}$$

- Although all the computers in the processing cluster have similar hardware configurations, the model of the processor might slightly differ from one to another. Taking the slowest computer as the patron reference, a normalization process is applied to measured times according to the *MIPS*[3] score of each computer. Let β_{S_i} be the *MIPS* score for the machine that processed the sequence S_i. Hence, the normalized time for this sequence would be

$$\Delta T''_{S_i} = \frac{\beta_{S_i}}{\min_i \beta_{S_i}} \Delta T'_{S_i}. \tag{4.29}$$

[3] *MIPS* (Millions of Instructions Per Second) is a measure of a computer's processor speed. This measure is usually provided by the computer system after conducting a benchmarking test.

The processing time per frame stated in Eq.4.24 shows that all algorithms share a common processing lapse formed by the creation/deletion of a track time, $T_{\text{New Track}}$ and $T_{\text{Delete Track}}$, and the time to generate the input data T_{Input}. In order to present a fair analysis among all algorithms, only the filtering time T_{Filter} should be compared. That implies assuming that creation/deletion of tracks and input data generation to be computed in a separate machine(s) and the filtering step to be carried out on another machine. This assumption is founded on the experience with real-time implementations of tracking algorithms in the framework of the CHIL Project [CHI07] where a distributed computing software was employed leading to satisfactory results.

Some words might be drawn on the common processing lapse involved in Eq.4.24 and the devised process to obtain a fair measure of the filtering lapse:

- **Input data generation**. This time lapse can be expressed as:

$$T_{\text{Input}} = \left(T_{\text{Read Image}} + T_{\text{Segmentation}}\right) N_{\text{C}} + T_{\text{SfS}} + T_{\text{Voxel coloring}}. \tag{4.30}$$

  All the involved time measures have a dependency on the number of cameras N_{C} used by the system. Typically, the result of this data generation process was pre-computed for each sequence and stored in a hard disk. In this way, the T_{Input} lapse was reduced to a mere disk access with a negligible value: $T_{\text{Input}} \approx 0$. In a continuous operation mode where images are grabbed directly from live cameras, i.e. in a SmartRoom scenario, the employed distributed system was able to perform satisfactorily. In the most demanding scenario where $s_{\mathcal{V}} = 4$ cm was used, a $RTF = 0.4$ was observed[4].

- **Track creation**. This time lapse can be written as:

$$T_{\text{New Track}} = T_{\text{CCA}} + T_{\text{Feature Extraction}} + T_{\text{Decision}}. \tag{4.31}$$

  Despite using an efficient implementation based on hierarchical queues [Vin93, SP94], it was observed that the most demanding operation was the computation of the connected component analysis (*CCA*), consuming most of the CPU of the system. Being this operation the bottleneck in all the processing chain, it was considered opportune to pre-process all input data and store the result of this *CCA* thus $T_{\text{CCA}} \approx 0$. Moreover, since it is intended to compare the execution times of the several presented algorithms, we opted for this pre-computation of the *CCA* operation. However, in a real case scenario, the *CCA* step may be addressed by a dedicated computer in a modular architecture, as carried out in [CHI07]. Feature extraction and the rule-based person/no-person binary decision modules can not be pre-computed since their operation depends on the state of the filter at the time of execution, as explained in §4.2.3.

- **Track deletion**. This step is composed of the following time factors:

$$T_{\text{Delete Track}} = T_{\text{Feature Extraction}} + T_{\text{Decision}}. \tag{4.32}$$

[4] For the reader in the future, machines used in this investigation were off-the-shelf computers with a 3.0 GHz processor and no dedicated or specific hardware.

Again, these two involved operations depend on the state of the filter hence can not be pre-computed.

- **Filtering**. Filtering operation is the desired time to analyze and compare. The execution time of each algorithm receives two contributions:

$$T_{\text{Filtering}} = T_{\text{Data Access}} + T_{\text{Process}}. \tag{4.33}$$

The first factor accounts for the lapse to access the memory and retrieve the stored input data while the second factor is the data processing lapse. In any case, these two times can not be measured separately.

Finally, the pre-computation of some of the bottlenecks involved in the processing chain allows the measured time per frame $\Delta T_{\mathbf{z}_t}$ to be a better approximation of the filtering time T_{Filter}. Indeed, unless these consideration are taken into account, this measure can not be achieved. *FPS* can be referred to $\Delta T_{\mathbf{z}_t}$ from Eq.4.25 as

$$FPS_{\mathbf{z}_t} \approx \frac{1}{\Delta T_{\mathbf{z}_t}} = \frac{1}{T_{\text{Input}} + T_{\text{New Track}} + T_{\text{Filter}} + T_{\text{Delete Track}}}. \tag{4.34}$$

However, if we enforce the relation

$$T_{\text{CCA}} \gg \{T_{\text{Input}}, T_{\text{Feature Extraction}}, T_{\text{Decision}}, T_{\text{Filter}}, T_{\text{Delete Track}}\}, \tag{4.35}$$

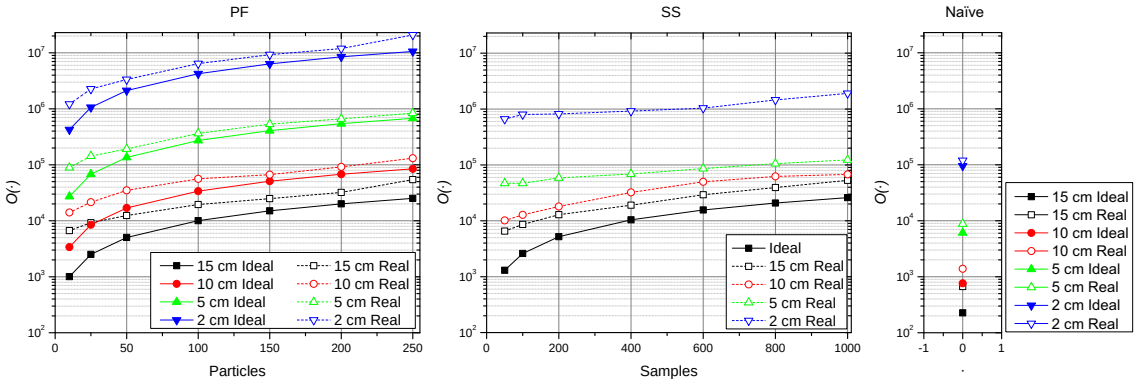
it becomes that

$$FPS_{\mathbf{z}_t} \approx T_{\text{CCA}}^{-1}. \tag{4.36}$$

Hence, this measure only depends on the voxel size $s_{\mathcal{V}}$ masking the time lapse that we want to compare, T_{Filter}.

Theoretical and measured performance

First, we will compare the measured execution time of an algorithm with their associated theoretical complexity reported in Table 4.5. The measured execution time can be transformed into a complexity measure from the expression obtained in Eq.4.26 as $O(\cdot) \propto FPS^{-1}$. A comparative plot between the theoretical and measured complexities of the algorithms is shown in Figure 4.13. In the theoretical expression, it can be seen that there is a dependency with the size of the analysis region, ellipsoid, in the Naïve and PF approaches ($s_x^2 \bar{h}$ term) while the SS algorithm only depends on the number of samples employed and the size of the sample's evaluation neighborhood, depending on constants r (radius) and q (connectivity). However, these expressions only account for the filtering time. Dealing with sparse data, such as the noisy 3D voxel reconstructions contained in $\mathbf{z}_t$, may be computationally inefficient thus leading to a difference between the observed and predicted computational cost. If the summands in Eq.4.33 are in the same order of magnitude or less, $T_{\text{Data Access}} \lesssim T_{\text{Process}}$, measured times will still follow the predicted theoretical curve, as fulfilled by the Naïve and PF approaches. However, this condition is not fulfilled by the SS filter, with a theoretical complexity lineal with the number of samples N_{s} and invariant to the voxel size $s_{\mathcal{V}}$. Contrarily, the measured complexity has



(a) PF (b) SS (c) Naïve

Figure 4.13: *Theoretical (ideal) and measured (real) algorithm complexities comparison.*

a dependency with $s_{\mathcal{V}}$ due to the increasing complexity to access data. For this algorithm, it might be assumed that $T_{\text{Data Access}} \gg T_{\text{Process}}$, hence only a biased measure of its computational performance can be obtained. This explains the dissimilarity between the theoretical and measured complexity curves.

Method	**Complexity**
Naïve filter	$O\left(\alpha^2 s_x^2 \bar{h} s_{\mathcal{V}}^{-3}\right)$
Particle filter	$O\left(N_{\text{p}} s_x^2 \bar{h} s_{\mathcal{V}}^{-3}\right)$
Sparse sampling filter	$O\left(N_{\text{s}} q r^3\right)$

Table 4.5: *Algorithm theoretical complexity expressions per track referred to the average person height, $\bar{h}$, and xy diameter, s_x.*

MOTP/MOTA vs *RTF*

The *RTF* factor associated with a performance measure *MOTP*/*MOTA* (in both vertical axes) of the SS and PF algorithms when dealing with raw and colored input voxels is presented in Figure 4.14. Each point of every curve is the result of an experiment conducted over all the CLEAR dataset associated to a number of samples/particles of each algorithm.

The first noticeable characteristic of these charts is that, due to the computational complexity of each algorithm, when comparing SS and PF algorithms under the same operation conditions, the *RTF* associated with SS is always higher than the associated with PF. Similarly, the computational load is higher when analyzing colored than raw inputs. All the plotted curves attain lower *RTF* performance values as the size of the voxel $s_{\mathcal{V}}$ decreases since the amount of data to process increases (note the different *RTF* scale ranges for each voxel size in Figure 4.14). Regarding the *MOTP*/*MOTA* metrics,

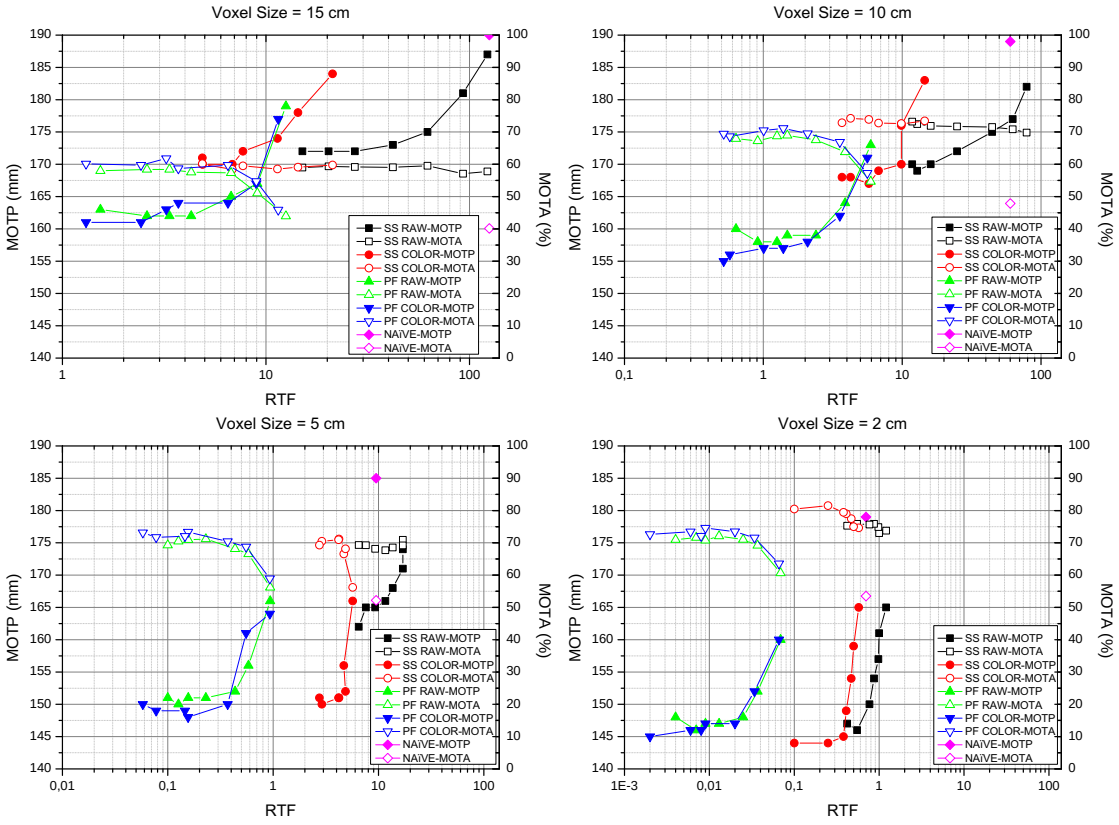


Figure 4.14: *Computational performance comparison among Naïve, Sparse Sampling (SS) and Particle Filtering (PF) using several voxel sizes* $s_{\mathcal{V}} = \{2, 5, 10, 15\}$ *cm and features (raw or colored voxels).* MOTP *and* MOTA *scores are related to the real-time factor (*RTF*) showing the computational load required by each algorithm to attain a given tracking performance.*

there is a common tendency to a decrease in the *MOTP* and an increase in the *MOTA* as the *RTF* decreases. The separation between the SS and PF curves is bigger as the voxel size decreases since the PF algorithm has to evaluate a larger amount of data.

The observation of these results yield to the conclusion that the SS algorithm is able to produce a similar and, in some cases, better results than the PF algorithm with a lower computational cost.

4.5 Conclusions

In this chapter, we have presented a number of contributions to the multi-person tracking task in a multi-camera environment. A block representation of the whole tracking process allowed to identify the performance bottlenecks of the system and address efficient solutions to each of them. Real-time performance of the system was a major goal hence efficient tracking algorithms have been produced as well as an analysis of their

performance.

The performance of these systems have been thoroughly tested over the CLEAR database and quantitatively compared through two scores: *MOTP* and *MOTA*. A number of experiments have been conducted towards exploring the influence of the resolution of the 3D reconstruction and the color information. These results have been compared with other state-of-the-art algorithms evaluated with the same metrics using the same testing data.

The relevance of the creation and deletion of filters has been proved since these modules have a major impact on the *MOTA* score. However, most of papers in the literature do not specifically address the operation of these modules. We proposed a statistical classifier based on classification trees as a way to discriminate blobs between the person/no-person classes. Training of this classifier was done using data available in the development part of the employed database and a number of features (namely weight, height, top in z axis, bounding box size) were extracted and provided as the input to the classifier. Another criterium such as a proximity to other already existing tracks was employed to create or destroy a track. Performance scores in Table 4.4 for the PF and SS systems present the lowest values for the false positives (*FP*) and missed targets (*Miss*) ratios hence supporting the relevance of the creation and deletion of tracks modules.

Three proposals for the filtering step of the tracking system have been presented: Naïve, Particle Filtering (PF) and Sparse Sampling (SS). An independent tracker was assigned to every target and an interaction model was defined. PF technique proved to be robust and leaded to state-of-the-art results but its computational load was unaffordable for small voxel sizes. As an alternative, SS algorithm has been presented achieving a similar and, in some occasions, better performance than PF at a smaller computational cost. Its sample based estimation of the centroid allowed a better adaptation to noisy data and distinguishability among merged blobs. In both PF and SS, color information provided a useful cue to increase the robustness of the system against track mismatches thus increasing the *MOTA* score. In the SS, color information also allowed a better placement of the samples allowing to distinguish among parts belonging to the tracked object and parts of a merging with an spurious object, leading to a better *MOTP* score.

As a final remark, the presented systems have been employed in more sophisticated schemes combining audio and video information towards providing a multi-modal tracking solution. The reader is referred to our contributions in this topic [BTNCF08a, BTNCF08b].

5 Human Motion Capture Evaluation

SYSTEMATIC evaluation of computer vision algorithms has raised a growing interest in recent times. Periodic evaluation campaigns allow fair comparison of different techniques, avoiding subjectivity through an agreed set of well-defined metrics for assessment and a reference corpus of pertinent data for testing. Along this line, noteworthy examples can be found in the field of face recognition [PHRR00, CLE07], person tracking [CLE07, PET07], articulated body motion [SB06] or gait recognition [SPL+05].

In the field of human motion capture (HMC) there is an incipient interest to compare existing algorithms towards assessing their performance. Although there is a number of existing datasets containing human motion for action or gait recognition, very few include ground truth information about the precise location of human landmarks. This ground truth, so crucial to evaluate HMC algorithms, is usually recorded using professional hardware, hence it is an expensive procedure. Up to now, the only widely accepted dataset intended for HMC evaluation is the HumanEva-I dataset [SB06].

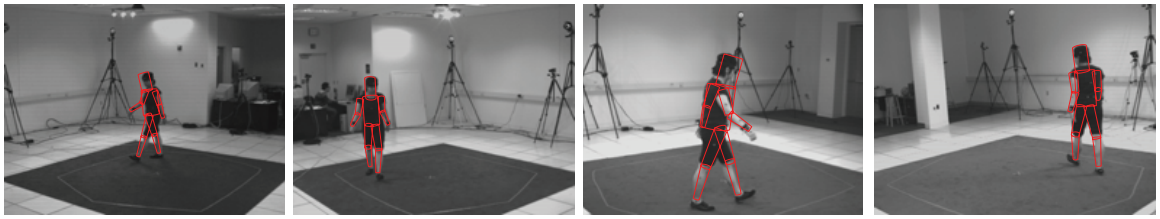
In this chapter we will analyze the existing HMC evaluation metrics based on the first and second statistical moments of the Euclidean error between the estimated pose and the ground truth data. However, we will discuss how these metrics exhibit some inconveniences and may lead to biased scores under certain conditions. Two alternative metrics are presented towards avoiding such effect and they will be the reference scores for further comparison among HMC methods presented along Chapter 6 and 7.

Within this field, the following articles have been published: [CFCPM09b].

5.1 Methodology

Two main efforts drive evaluation processes: the definition of metrics and data collection and labelling. Metrics are usually agreed beforehand by evaluation organizers, with input from the participants and the community at large. A set of scoring tools is usually provided to speed up the scoring process once the ground truth is released after the delivery of results. Data collection and labelling is, by far, the most demanding task for evaluation organizers, because they have to record and label a common dataset (corpus) over which every algorithm to be test will be run. The dataset contains a significant amount of data, including all possible situations an algorithm may encounter. It is important to provide enough instances and a rich variation of each event to be detected for the evaluation result to be statistically meaningful.

(a) C1 (b) C2 (c) C3



(d) BW1 (e) BW2 (f) BW3 (g) BW4

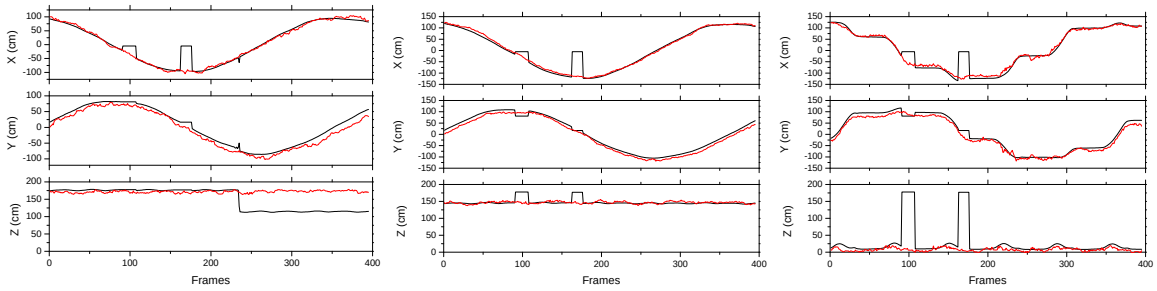
Figure 5.1: *HumanEva-I data sample. The synchronized ground truth data is overlaid on the multi-view image data for walking.*

5.1.1 HumanEva dataset

HumanEva-I dataset introduced by Sigal *et al.* [SB06] has become a standard for HMC performance evaluation. It consists of 7 cameras with a resolution of 640×480 pixels, non-interlaced, 3 of them in color and the rest in grayscale. Calibration information is available and obtained through Bouget's Calibration Toolbox for Matlab [Bou04]. In order to obtain ground truth information about the body pose, a Vicon professional motion capture system is employed [Vic] to retrieve the position of a set of landmarks placed on the performer. A sample of this dataset if shown in Figure 5.1. Synchronization between the Vicon system and the cameras is achieved by off line post-processing. Together with this dataset, a statistic model of the background for every camera is provided to facilitate foreground segmentation.

The dataset consists of 6 actions (walking, jogging, throw/catch, gesturing, boxing and combo) performed by 4 different performers. A portion of the ground truth information has been withheld by the organizers for the sake of fairness in future evaluations. Hence, for our experiments we will employ only the sequences provided with ground truth information thus reducing the dataset to 5 actions (walking, jogging, throw/catch, gesturing and boxing) for 3 subjects adding up to 10 minutes of data. Despite this figure might seem short in comparison with the size of CLEAR dataset [CLE07] employed in Chapter 4, the amount of data to be estimated per frame (body pose) is high enough to render this HumanEva-I dataset usable to obtain statistically meaningful results.

It must be noted that ground truth data present some glitches as shown in Figure 5.2. Perhaps the most noticeable artifacts are the step changes, well localized in time



(a) Head Top (b) Right Shoulder (c) Right Foot

Figure 5.2: *HumanEva-I ground truth data example (x, y and z coordinates). In black, the ground truth data and, in red, an example of estimation.*

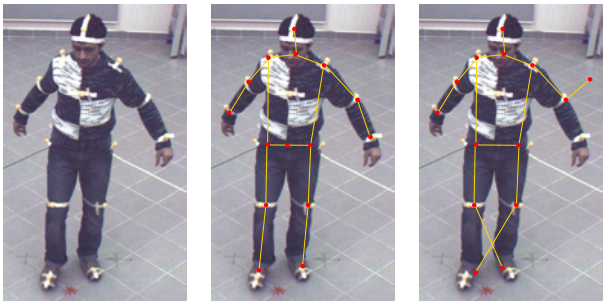
and appearing simultaneously in all ground truth measurements. Another distortion in the ground truth can be seen in Figure 5.2(a) where z coordinate drops to a fixed value. Although these artifacts are to appear seldom, they might introduce an offset into the performance measures, but comparisons among methods will still be valid.

Recently, the same authors released a second version of this set, HumanEva-II. However, only the images have been distributed while the ground truth data has been withheld. Researchers willing to use this dataset have to submit their HBM pose hypotheses and they receive the performance scores computed using the ground truth data.

5.2 Performance Evaluation

In the field of articulated body motion, there is still no general agreement on a principled evaluation procedure using a common set of objective and intuitive metrics for measuring the performance of different articulated motion tracking algorithms. Due to this lack of metrics, some researchers present their tracking systems without quantitative evaluation of their performance [DR05, RBM05]. On the other hand, a multitude of isolated measures were defined in individual contributions to validate their trackers using various features and algorithms. Recently, a significant contribution [SB06] released two metrics that have been adopted in several evaluation campaigns. Nevertheless, these metrics present some inconveniences and may produce biased scores under certain conditions.

In order to define objective and informative performance evaluation metrics, two design criteria should be followed. First, they should allow to judge the tracker's precision in determining the exact location of the articulated structure landmarks. Secondly, they should reflect its ability to consistently track the landmark locations through time, i.e., to correctly trace their trajectories. Finally, useful metrics should have as few adjustable thresholds as possible to help make evaluations straightforward and keep results comparable. Two sets of metrics are proposed in this thesis to measure the performance of the tracking of an arbitrary articulated structure.



(a) (b) (c)

Figure 5.3: *Point based metrics comparison example. In (a), the reference image with the visual yellow markers. In (b), a good body pose estimation produces* $\mu = 48.91$ *and* $\sigma = 16.21$*, and* MMTP $= 48.91$ *and* MMTA $= 1.0$*. In (c), a poor body pose estimation produces* $\mu = 51.22$ *and* $\sigma = 24.37$*, and* MMTP $= 46.35$ *and* MMTA $= 0.77$ *with* $\epsilon = 50$ *(all distance units in mm).*

5.2.1 Problem Formulation

Given a HBM whose pose is represented by a state vector $\mathbf{y} \in \mathcal{X}$, we may represent any adopted pose by a set of M virtual markers encoded as a vector $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$, where $\mathbf{x}_m \in \mathbb{R}^3$. A mapping from $\mathbf{y} \in \mathcal{X}$ to X can be always derived, either if the state vector encodes landmark positions (using a linear mapping) or joint angles (applying forward kinematics). We will denote as *point based metrics* those measures computed directly from this vector X.

Basically, two point based metrics have been widely adopted as stated by Sigal and Black [SB06]. Let us define the error between an estimated pose $\hat{X}$ with reference to the ground truth pose X as:

$$D(X, \hat{X}) = \frac{1}{M} \sum_{m=1}^{M} \|\mathbf{x}_m - \hat{\mathbf{x}}_m\| . \tag{5.1}$$

This error figure can be assumed to have a Gaussian distribution and the first and second statistical moments can be derived. Hence, when a sequence of poses of length T is analyzed, the performance of the tracking algorithm may be assessed by averaging error along time and computing the standard deviation. This produces the metrics:

$$\mu = \frac{1}{T} \sum_{t=1}^{T} D(X_t, \hat{X}_t), \tag{5.2}$$

$$\sigma = \sqrt{\frac{1}{T} \sum_{t=1}^{T} \Big(D(X_t, \hat{X}_t) - \mu \Big)^2}. \tag{5.3}$$

Good performance of a HBM tracking algorithm will yield low values of both μ and σ, whereas high values will denote a poor efficiency. This metric produces meaningful

results at a given time instant when the sets X and $\hat{X}$ fulfill the condition:

$$\|\mathbf{x}_m - \hat{\mathbf{x}}_m\| \leq \epsilon, \qquad \forall m, \tag{5.4}$$

being ϵ a fixed threshold. This parameter ϵ discriminates whether the position $\mathbf{x}_m$ and the estimation $\hat{\mathbf{x}}_m$ can be considered as matched. This is the case of a pose configuration $\hat{X}$ similar to the one depicted in Figure 5.3(b) where the estimation of the landmark positions are close to the ground truth positions. When this condition is not fulfilled for some values of m, then the algorithm outputs estimate poses $\hat{X}$ as depicted in Figure 5.3(c). A limitation of these metrics is that the non matched case is not distinguished from a matched case. In the former case, the error produced when the estimation of a certain landmark is clearly far away from the ground truth (that is when $\|\mathbf{x}_m - \hat{\mathbf{x}}_m\| > \epsilon$) is still accounted as a gross estimation inaccuracy thus severely penalizing both μ and σ scores. Therefore, when a landmark subset is not tracked properly (typically, the end of the limbs), the figures produced by these metrics are not informative enough to describe the tracker's performance.

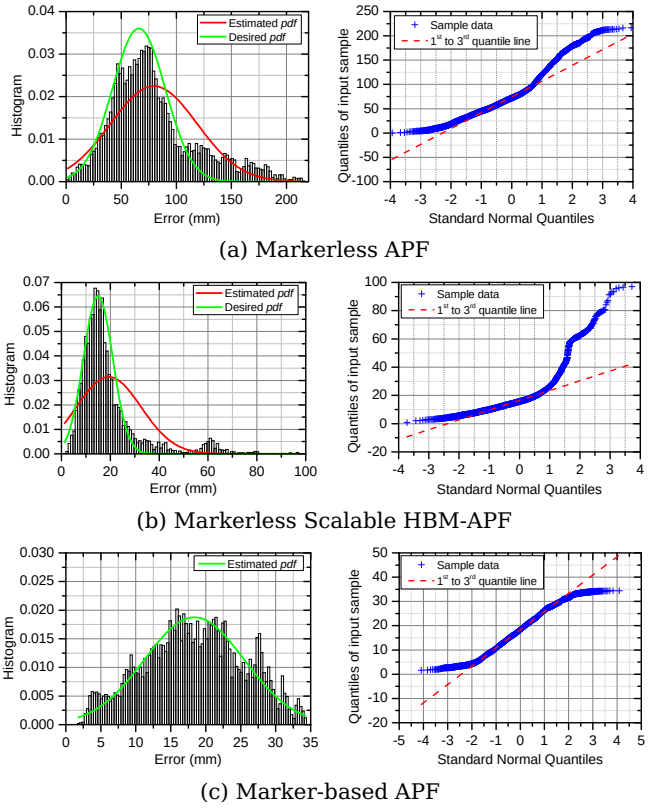
5.2.2 Statistics

In order to test the Gaussianity distribution assumption stated in Eqs.5.2 and 5.3, the error vector E is generated, $E = \{\mathbf{e}_k\} = \{\|\mathbf{x}_{m,t} - \hat{\mathbf{x}}_{m,t}\|\}$, $\forall m, t$, including the estimation error associated to every body marker along the whole analysis sequence. When analyzing the histogram of E for several human motion capture algorithms shown in Fig.5.4, it can be seen that there is a dominant peak with a Gaussian shape associated to error values fulfilling $\mathbf{e}_k \leq \epsilon$, while the long tail spreading to large error values is derived from those satisfying $\mathbf{e}_k > \epsilon$. Indeed, when analyzing the estimated *pdf* associated to E, it can be seen how it does not properly fit to a Gaussian distribution (red line). In an ideal case, as in Fig.5.4c, the estimated *pdf* matches a Gaussian function while, in the other cases, the desired (green line) and computed *pdf* differ substantially.

The quantile-quantile plot is an efficient way to assess the Gaussianity of a distribution [JW07] in the sense that the empirical quantiles of the data are plotted versus the quantiles of a Gaussian. If the data belongs to a Gaussian distribution, the points are spread roughly following a line. If the data is skewed or has longer/shorter tails than a Gaussian, instead of having a line, the scatter plot shows either flat or vertical parts. As it can be seen in Fig.5.4, both markerless algorithms error data do not properly align with the regression line while the marker-based one does, as expected from the associated histograms.

5.3 Metrics

A new point based metric is proposed in this thesis in order to better express the performance of a HBM tracking algorithm. It takes into account that there might be situations were a subset of the landmarks in $\hat{X}$ is not estimated properly while the rest is done accurately. A similar problem is found in the field of multiple object tracking by Bernardin *et al.* [BES06] and previously employed in Chapter 4. These authors proposed a set of



(a) Markerless APF

(b) Markerless Scalable HBM-APF

(c) Marker-based APF

Figure 5.4: *Histograms associated to the estimation error and the quantile-quantile plot between the error vector E and a reference normal distribution.*

metrics that were validated and largely accepted as performance and comparison scores in international evaluation campaigns [CLE07]. The underlying concept of these performance metrics may be extended to the field of pose estimation evaluation to produce two intuitive and more informative metrics.

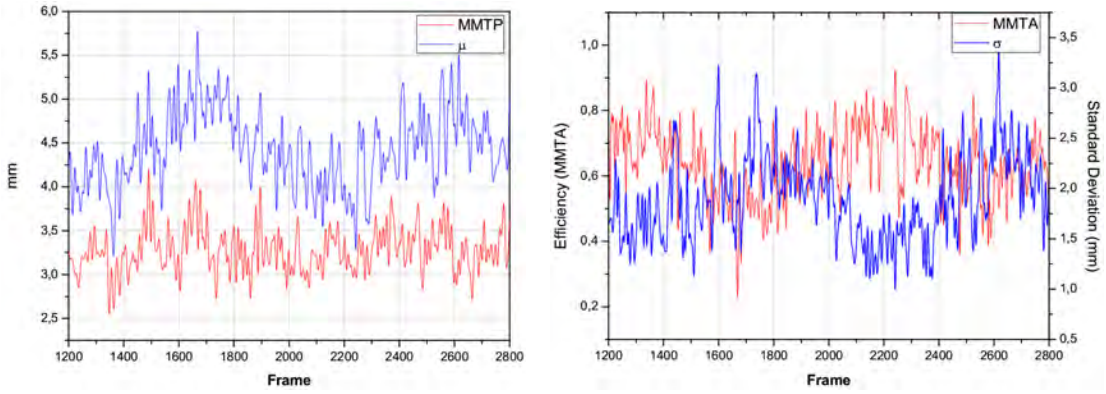
5.3.1 Point Based Metrics

Let us define the set Ω as the set of pairs estimation-ground truth locations whose distance is below the threshold ϵ, that is $\Omega = \{(\mathbf{x}_m \in X, \hat{\mathbf{x}}_m \in \hat{X}) / \|\mathbf{x}_m - \hat{\mathbf{x}}_m\| \leq \epsilon\}$. The two metrics can be defined:

1. The *M*ultiple *M*arker *T*racking *P*recision (*MMTP*),

$$MMTP = \frac{\sum_{t=1}^{T} \sum_{m \in \Omega_t} \|\mathbf{x}_{t,m} - \hat{\mathbf{x}}_{t,m}\|}{\sum_{t=1}^{T} \#\Omega_t}, \tag{5.5}$$

where $\#\Omega$ denoted the cardinality of the set Ω. This metric shows the total position error for the matched ground truth-estimation pairs, averaged by the total number



(a) (b)

Figure 5.5: *Quantitative performance of point based metrics. In (a), landmark estimation precision scores μ and* MMTP *along time and, in (b), the plot of the evolution of scores σ and* MMTA*.*

of matches made along time. It reflects the ability of the tracker to estimate precise landmark positions, independent of the performance of the algorithm to correctly match all the landmarks in the HBM.

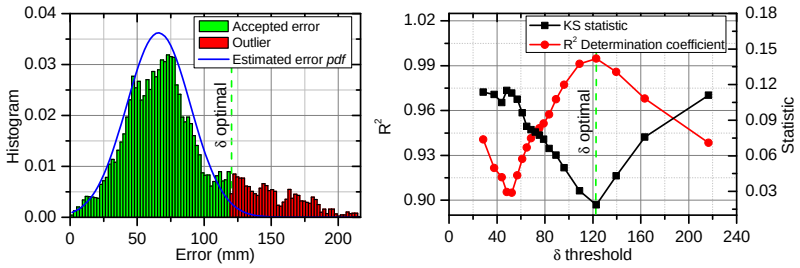
2. The *M*ultiple *M*arker *T*racking *A*ccuracy (*MMTA*),

$$MMTA = 1 - \frac{\sum_{t=1}^{T} \#\Omega_t}{M \cdot T}, \tag{5.6}$$

where M is the total number of landmarks in the HBM. This score accounts for the ability of the tracker at producing matched ground truth-estimation pairs.

Finally, a supplementary metric might be defined: the standard deviation of the *MMTP* score as a measure of the quality of the estimation of the correctly matched estimation-ground truth pairs. Although this figure is upper bounded as $\sigma_{MMTP} < \epsilon$, it provides information about the quality of the estimation of the matched pairs.

In the example depicted in Figure 5.3, these two sets of metrics are compared. When the condition expressed in Eq.5.4 is fulfilled as in Figure 5.3(b), metrics $\mu = 48.91$ and $\sigma = 16.21$ properly evaluate the estimated pose. In Figure 5.3(c), a typical situation of landmark estimation swapping is found in the ankles, while the left hand track is lost. In this case, the implicit assumption that these landmark estimation inaccuracies follow a Gaussian distribution clearly biases the scores, $\mu = 51.22$ and $\sigma = 24.37$. *MMTP* and *MMTA* can nicely handle both situations: in the first case $MMTA = 1.0$ indicates that the tracker has correctly produced a valid estimation for all markers and the average precision is $MMTP = 48.91$. In the second case, $MMTA = 0.77$ indicates that the tracker could only track the 77% of the landmarks during the analysis period of time and the average precision of the correctly tracked landmarks was $MMTP = 46.35$ which is not biased by the non matched pairs.



(a) (b)

Figure 5.6: *ϵ selection. In (a), parameter ϵ_{opt} partitions the error histogram between the Gaussian and outlier parts.*

A quantitative comparison of the temporal evolution of the presented point-based metrics is depicted in Figure 5.5. It is shown that the score μ is more sensitive than *MMTP* since it agglutinates both information from precision and lost tracks. An instantaneous version of σ computed every frame and *MMTA* is depicted to show the noticeable correlation between both scores: when there are less matched pairs estimation-ground truth, *MMTA* figure decreases while the deviation of the error increases and viceversa. However, the value of σ has little physical interpretation when some landmarks are not tracked properly while *MMTA* presents the amount of correctly tracked landmarks.

It must be noted that these results have been presented for $\mathbf{x}_m \in \mathbb{R}^3$, but these metrics can be adapted to the case where the landmark locations are measured directly on images, that is $\mathbf{x}_m \in \mathbb{R}^2$.

Parameter-free evaluation

Selecting an adequate value of ϵ is crucial to obtain meaningful *MMTP* and *MMTA* scores. When selecting small values of ϵ, the proposed metrics will be very restrictive thus yielding to a low *MMTA* and high *MMTP* values. On the other hand, large values of ϵ will report a tendency to $MMTA \rightarrow 1$ and $MMTP \rightarrow \mu$. Although ϵ may be set up manually allowing a maximum allowed error, a parameter-free evaluation procedure would be desirable.

The optimal value of ϵ, ϵ_{opt}, should be one that partitions the histogram of E in such a way that values fulfilling $\mathbf{e}_k \leq \epsilon_{\text{opt}}$ tend to have a Gaussian distribution, as shown in Figure 5.6a. Therefore, *MMTP* and σ_{MMTP} will stand for the mean and variance of the green bins approximated by the Gaussian function plotted in blue. In this way, *MMTA* will account for the fraction of the error that can not be considered as belonging to this Gaussian distribution.

In order to select the adequate value of ϵ_{opt}, we formulated the following optimization problem:

$$W(\epsilon) = \{\mathbf{e}_k \in E \mid \mathbf{e}_k \leq \epsilon\}, \tag{5.7}$$

$$\epsilon_{\text{opt}} = \min_{\epsilon} f\left(W\left(\epsilon\right)\right), \tag{5.8}$$

where $f(\cdot)$ stands for a normality test function over the values of $W(\epsilon)$. Two options have been considered for $f(\cdot)$. First, we employed the Lilliefors/Kolmogorov-Smirnov statistic [Con80] that measures the maximum difference between the empirical cumulative distribution function (CDF) of the input data $W(\epsilon)$ and the theoretical CDF of a Gaussian. This statistic measures a local feature of both CDF's, which is the worst discrepancy. In our optimization problem, we search the value of ϵ that minimizes the maximum absolute difference between the empirical CDF up to ϵ and the theoretical CDF of a Gaussian. Second, a linear regression [JW07] is applied over the quantiles of the input data with reference to the quantiles of a Gaussian. Then, we compute the coefficient of determination R^2 which is related to the explained variance and measures a global feature, i.e. the dispersion around the regression line. Values of R^2 near 1 mean that the data is aligned with the regression line, while low values hint at a lack of linear dependence. When employing the R^2 figure, Eq.5.8 minimum is replaced by a maximum. Despite these two scores measure different aspects of the problem, we have found that they usually agree on the value of ϵ_{opt} as depicted in Figure 5.6b.

Although a beforehand agreed ϵ parameter should be employed in an evaluation campaign to fix the maximum allowed error in the estimation of a pose, its value should be carefully selected not to produce biased values of *MMTP* and *MMTA* due to a wrong Gaussian distribution assumption of set $\mathbf{e}_k \leq \epsilon$. In a more thorough evaluation process, the value of ϵ_{opt} may give a useful clue to determine the range of correct operation of an algorithm, understood as the error range where limbs can be considered correctly tracked.

5.3.2 Angle based metrics

A natural choice in evaluating the performance of an articulated motion capture system would be to produce a score directly related to the defining parameters of the HBM, that is angles. Moreover, angle-based metrics have the advantage of exactly measuring the error at each joint of the articulated structure. The advantage over point based metrics lies in the fact that measured angles are relative to the two vertexes of the articulation ending at the joint and, therefore, tracking errors do not accumulate towards the end of the limbs, as happens with the spatial position measured in point based metrics. Encoding the pose of a given HBM $\mathcal{H}$ by a set of N angles[1], $\Theta_{\mathcal{H}} = \{\theta_1, \theta_2, ..., \theta_N\}$, $\theta_n \in \mathbb{R}$, is a common approach in articulated motion tracking, since these magnitudes are directly related with the kinematic structure of the human body [AT04, SKLM05]. In addition, this information has a straightforward application for gait and gesture analysis purposes.

Defining metrics based on an angle representation of the HBM presents some issues to be taken into account. For example, every pose encoding based on angles assumes a parameterization of the human body that can not be the same among algorithms enforcing different degrees of freedom in every joint. Furthermore, joint angle representations are not unique (quaternions, Euler angles, exponential maps, etc.) thus making comparisons among algorithms difficult. Some researchers have already proposed metrics measuring the error in terms of degrees at every joint [AT04] but due to the aforemen-

[1] The body root position and orientation are omitted for the sake of notation simplicity.

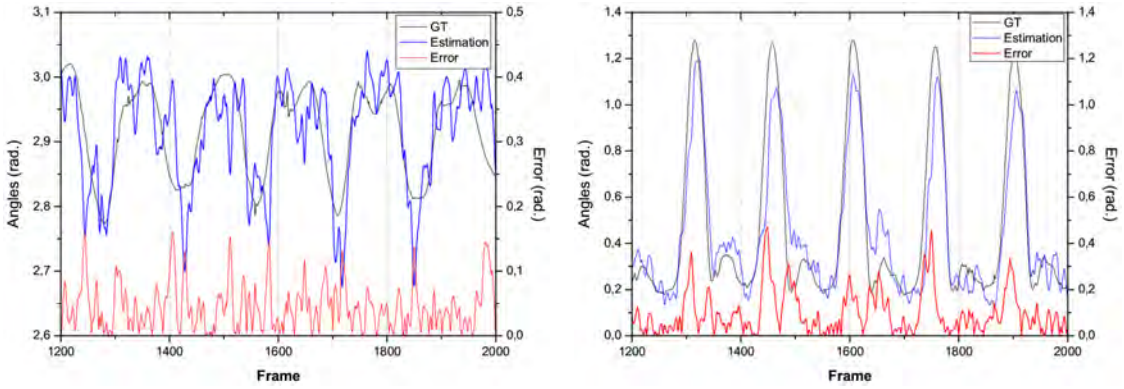


Figure 5.7: *Angular re-parameterization example for the elbow, (a), and the knee (b) articulations executing the action walking. Once the angles of the ground truth (black) and the estimation (blue) are expressed with the same HBM parameters, the error (red) between them can be computed and analyzed by the angular metrics.*

tioned issues, no angular metric has yet been widely adopted by the community. In this thesis, we propose a general method for evaluating the performance of an articulated motion capture system in terms of angles, regardless of the parameterization employed during the analysis.

In order to define an angle based metric, a reference HBM $\widetilde{\mathcal{H}}$ representation should be adopted. An obvious choice would be to define a transformation between the HBM $\mathcal{H}$ used by the tracking algorithm and $\widetilde{\mathcal{H}}$ but this mapping cannot be always computed due to the differences among HBM parameterizations. We propose the following re-parameterization technique. First, a given pose $\Theta_{\mathcal{H}}$ is transformed into a set of 3D coordinates, X, by applying forward kinematics since, as noted in §5.3.1, it is always possible to perform a mapping from a pose $\mathbf{y} \in \mathcal{X}$ to X regardless of the parameterization of the HBM. This set X of 3D coordinates is implicitly labelled because it is known which body landmark is described by every 3D location. Finally, the inverse kinematic problem has to be solved by extracting the angles $\Theta_{\widetilde{\mathcal{H}}}$ of $\widetilde{\mathcal{H}}$ from the set X. We propose using the model defined by Mikič [Mik03] as $\widetilde{\mathcal{H}}$, since it presents an accurate parameterization of all joints in the body. Moreover, this particular choice of $\widetilde{\mathcal{H}}$ allows an algebraic expression of all of its angles if the 3D coordinates are labelled [Mik03]. An example of this process is depicted in Figure 5.7 where $\mathcal{H} \neq \widetilde{\mathcal{H}}$ to prove the described re-parametrization scheme.

This process is applied to both the ground truth 3D positions X to obtain the ground truth angles $\Theta_{\widetilde{\mathcal{H}}}$ and to the estimated pose $\hat{\Theta}_{\mathcal{H}}$ to derive $\hat{X}$ and, then, to obtain the estimated angles $\hat{\Theta}_{\widetilde{\mathcal{H}}}$. The error between an estimated pose $\hat{\Theta}_{\widetilde{\mathcal{H}}}$ to the ground truth pose $\Theta_{\widetilde{\mathcal{H}}}$ is defined as:

$$D(\Theta_{\widetilde{\mathcal{H}}}, \hat{\Theta}_{\widetilde{\mathcal{H}}}) = \frac{1}{M} \sum_{n=1}^{N} |(\theta_n - \hat{\theta}_n) \text{mod} \pm \pi|. \tag{5.9}$$

Two angular metrics are proposed: the angular mean estimation error μ_θ and its associated standard deviation σ_θ. However, computing these scores directly over all angles over a period of length T would generate biased results due to the reasons already discussed in §5.3.1. Hence, it is proposed to compute these metrics over the angles associated to two vertices fulfilling the matching criterium described in Eq.5.4.

The proposed angular metrics complement the information provided by the point based metrics and can not be presented alone. While the efficiency of the tracking system is assessed by the score *MMTA*, both *MMTP* and the pair μ_θ and σ_θ provide information about the precision of the system in the spatial and angular domains respectively.

5.4 Conclusion

In this chapter, we have discussed the efficiency of standard metrics for HMC performance evaluation based on the mean and deviation of the Euclidean error between the estimated pose and the ground truth data, measured at several landmarks places on the body of the performer. Widely adopted metrics introduced in the context of the HumanEva-I dataset have been found to produce biased results when the estimated pose is far from the ground truth for some landmark positions. We have presented *MMTP* and *MMTA* as an efficient alternative rid of such pitfalls. Two complementary metrics based on angles have been also introduced. Point and angular metrics together are proposed as informative figures to fairly asses HMC evaluation.

6

Multi-camera Human Motion Capture

OVER THE YEARS, there has been a growing interest in the topic of human motion capture (HMC), basically fostered by the number of applications that benefit from the retrieved information. For instance, automatic action recognition has been found useful for human computer interfaces and detection of unusual behaviors in security applications. Gait analysis derived from HMC data is used in the medical field to assess bio-mechanical pathologies and, in the biometrics domain, it provides informative cues for person recognition.

HMC has been usually addressed as an estimation problem entailing a number of challenges derived from the high dimensionality of the state space to be estimated (that is the human body pose) and the multimodal shape of the likelihood function relating this state space with the observable data. Monte Carlo based algorithms fed with information provided by multiple camera views are the mainstream research direction due to its ability to efficiently tackle these two aforementioned problems. In particular, particle filtering with annealing has been found to be the state-of-the-art in HMC and it will be the theoretical basis to build up the systems proposed in this chapter.

We address the HMC problem by processing a multi-camera video stream from two different perspectives: marker and markerless. Marker-based approaches rely on detecting a number of distinguishable markers attached to some body landmarks to infer the pose of the body. They are typically used in the cinema industry to create detailed avatars mimicking human motion, and usually require expensive and dedicated hardware. A system for real-time automatic marker based HMC using multiple standard cameras is presented as an economic alternative to this ad hoc equipment.

Markerless HMC using information gathered by multiple cameras is commonly achieved by analyzing all images separately and then combining the obtained data through a likelihood function in what can be understood as an information fusion at feature level. Instead, we propose a system for HMC relying on a voxelized reconstruction of the scene at each time instant. This choice is motivated by the technical requirements of the of the projects that UPC is involved with and the increasing availability of computer capacity to deal with multiple camera streams. By generating this 3D reconstruction, we perform an information fusion at data level thus getting rid of perspective and occlusion considerations. In all systems, automatic initialization of the body pose and the size of limbs is derived from information provided by the tracking module presented in Chapter 4.

The following publications by the author are related to this work: [CFCP05b, CFCTP05, CFCP^{+}06c, CFCP06b, OCFT^{+}08b, ODCF^{+}08, CFCP09a, CFCP09b, CFCP09c].

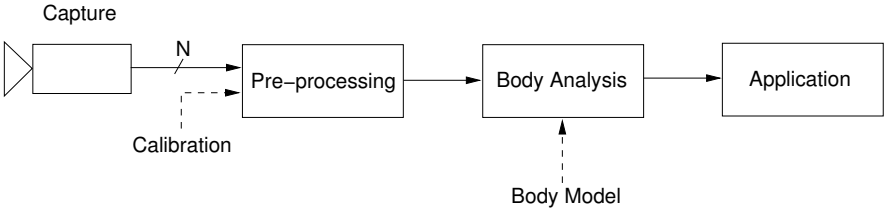


Figure 6.1: *Classical data flowchart in a human centered application: from input data (images) to a higher semantical analysis (application).*

6.1 State of the art

In general, the literature on video-based motion capture is vast and several approaches using very heterogeneous techniques have been proposed. In the last decade, some comprehensive surveys have covered the latest developments on vision-based human motion capture organizing the existing bibliography, each with a different focus and taxonomy [MHK06, Pop07b, Wer07]. A review of the literature reveals a common underlying structure when tackling the problem of human motion capture and analysis. Figure 6.1 depicts the standard pipeline of data flow approach when dealing with the process that goes from images to a human-centered application (understood as a higher semantic analysis extracting information related with humans: motion, gesture, gait, behavior, etc.). The first step of this processing chain consists on the data capture by a set of N_{C} cameras. These data is fed to a pre-processing step that may need some calibration information. This module extracts the information relevant for the next body analysis module, taking into account that not all types of information might be useful for all HMC algorithms. An analysis on these data is performed and some output is produced encoding information related to the humans in the scene. Finally, the application employs this output to perform a given action: detect a gesture, analyze gait, etc. In the following subsections, we briefly review the state of the art in human motion capture and analysis according to the blocks presented in Figure 6.1.

6.1.1 Data Capture

Input data features provided to the tracking system are a decisive factor when determining the employed methodologies and algorithms. Several capture devices are found in the literature: IR cameras, thermal cameras, CCD cameras, etc. The capture device providing more cues for processing in the environment of gesture analysis is the standard CCD camera since it outputs color information. Hence, other capture devices are not further reviewed in this thesis. A particular case is the input data employed by marker based systems where the scene is usually illuminated by IR light and the obtained images are processed in this domain.

The number of cameras, N_{C}, employed in the capture process is fundamental design parameter. Many analysis methods are devoted to the monocular case, $N_{\mathrm{C}} = 1$, [JBY96, BK01, GG04, AT04, SKLM05] and most of these methods encounter the inherent

problems of a single camera analysis: perspective and disambiguation of occlusions (with other objects in the scene or auto-occlusions). These limitations may be addressed by using some information from the human body structure towards retrieving the perspective information and overcoming occlusions. However, under certain conditions, these methods tend to fail at estimating an accurate human body pose.

Multiview capture, $N_{\mathrm{C}} > 1$, allows exploiting redundant information from several views in order to face the problems encountered in the monocular case [Mik03, CGH05, KG06]. The position of cameras is indeed crucial to obtain informative views (see [OM02] for more information on optimal camera position planning). The number of employed cameras vary among authors: $N_{\mathrm{C}} = 12$ [VU05], $N_{\mathrm{C}} = 8$ [Mik03], $N_{\mathrm{C}} = 3$ [CFCP08], etc. The more cameras available, the more information a HMC algorithm may use at the cost of augmenting the overall complexity of the system, which is usually proportional to N_{C}. Hence, there is a general trade-off between computational load and robustness/accuracy depending on N_{C}. Typically, systems fed by multiple video inputs make use of distributed capture and processing systems to avoid dropping frames, and parallelized algorithms (when possible).

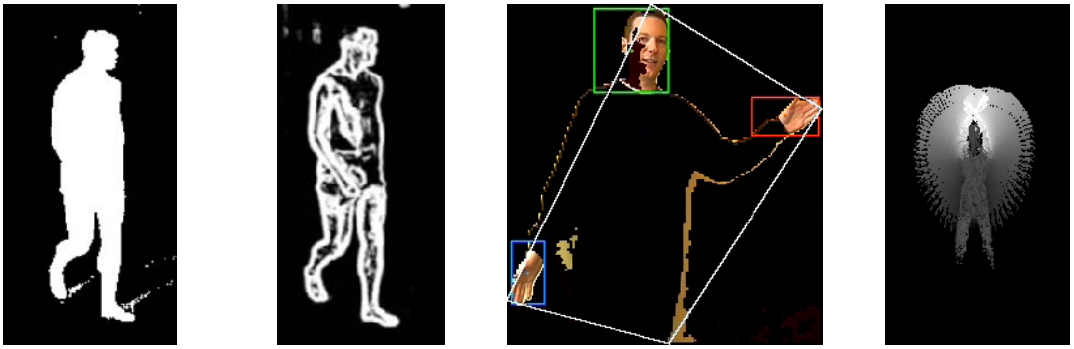
A special multiview capture configuration is the stereo vision case, $N_{\mathrm{C}} = 2$, given when two cameras are placed coplanar and very close to each other, thus resembling the field of view of human eyes. 3D depth may be retrieved from these images by means of a disparity map and can be useful for HMC when the subject is close to the cameras [GGTS01, ZNS06].

Synchronization among cameras is highly desirable since it allows analyzing data coming from the N_{C} video streams without adding complexity to the system. When video flows are not synchronized, off-line processing or on-line alignment is required at the cost of a delay in the processing chain.

6.1.2 Data Pre-Processing

Raw images acquired by the cameras are fed to the pre-processing module together with calibration information. Calibration information allows relating data in images with the real world through projective geometry [HZ04]. This information may be used to process the captured images and to embed 3D information in the data that will be processed by the analysis module [CFCP09a]. Some systems do not take into account calibration information, relying on cross-correlation analysis among images to state spatial correspondences [MSKS03]. Uncalibrated cameras are usually employed in environments where calibration can not be ensured for a long time (outdoor) or when the sequences have been recorded without calibration information available (i.e. sports events [YS05]). Calibration of multiple cameras has been addressed thoroughly by [Zha02] and [SMP05] (see previous §2.2.2 for more information on the employed camera model).

Once a set of N_{C} images are available at the input of the pre-processing module, several procedures may be applied in order to retrieve information useful for the body analysis task. In the following, we will enumerate the different features that are usually extracted from the images taking into account no calibration information (see Figure 6.2):



(a) Silhouette [DR05] (b) Edges [DR05] (c) Color [GBC02] (d) Motion [BD01]

Figure 6.2: *2D features useful for motion capture that do not employ calibration information.*

- **Silhouettes:** Input images can be segmented into background and foreground regions denoting those areas where motion has occurred [SG00]. Since we assume the entities that produce motion are humans, silhouettes contain information relevant to motion analysis. This feature has been widely employed in the literature [DF99, SC07, RRR08].

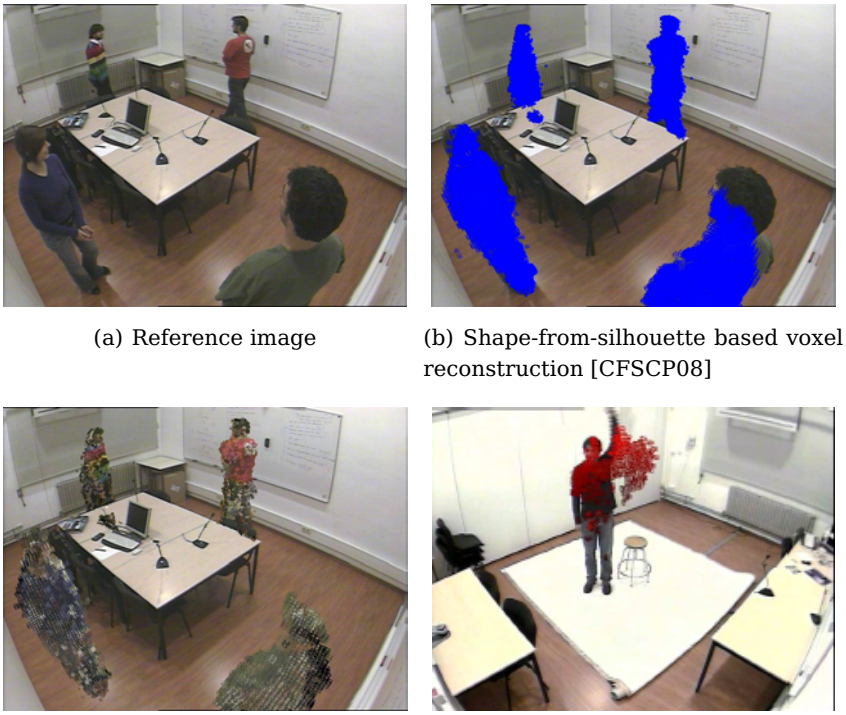
- **Edges and contours:** Contrast and color discontinuities generated by the person under study provide an informative cue for HMC since they present a recognizable pattern associated with the body structure (i.e. legs and arms). It has been used in [JBY96, KM00, DR05]. However, this method is prone to fail when applied to cluttered background scenarios.

- **Color blobs:** Grouping areas with similar color allows analyzing human motion as done in [Lan06]. In the particular case of well characterized colors like the skin, it has been used to detect and track the head and hands [GBC02].

- **Motion:** Motion may be extracted from consecutive images thus detecting those areas where a movement has occurred. Assuming this movement is produced by a person, these data can be analyzed to recognize certain types of motion as done by [BD01] with the motion-history-image descriptor or to recognize people using gait [MB06].

- **Markers:** Marker-based systems [SB06] are usually fed by the 2D detections onto images of a number of projected markers placed on the human body. In order to simplify the task, reflective markers are often used such, that when illuminated by IR light, they are clearly distinguishable by IR cameras.

When taking into account calibration, one may think about combining information from the previous pre-processing steps (or even the raw images) to generate a set of features embedding 3D information. Two main options are available in the literature:



(a) Reference image

(b) Shape-from-silhouette based voxel reconstruction [CFSCP08]

(c) Colored voxels [CFSCP08]

(d) 3D motion voxels [CFCP06b]

Figure 6.3: *3D features obtained after combining multiple images and the calibration information associated to each camera.*

- **Point-wise or 2D feature correspondence:** By establishing spatial correspondences among features detected in 2D, the 3D information may be retrieved. For instance, establishing correspondences of the top of the extracted silhouettes in multiple views allows performing 3D tracking of multiple people in a room [CFCP05b]. 3D correspondences among skin colored blobs allow detecting hands and head of a person to perform gesture recognition [FLD08]. Other research [LC03] computes correspondences among silhouette features (curvature, main axis, etc.) to derive 3D anthropometric measures for further analysis. Finally, intrusive methods based on placing distinguishable markers on the joints of the person allow retrieving accurate 3D positions of these joints through triangularization [KOF05]. Marker-based approaches use the detected 2D coordinates of distinguishable markers placed on the body using standard [CFCP09a] or adapted cameras [CPF03, Vic].

- **3D representation of the space:** Voxelization[1] is a method to combine information provided by N_C cameras based on generating a synthetic 3D discrete representation of the analyzed space. It has been found useful for body analysis purposes. Voxels, unities defining a regular grid partition of the 3D space, capture information about the presence and characteristics of an object in the scene as well as informa-

[1] For a more detailed information on the voxelization methods employed within this thesis, check §2.2.4.

tion about its real shape (in contraposition with their projections onto the images). There is a number of cues derived from a 3D reconstruction that have been used in HMC: raw voxels [Mik03, Che03], color [KG06] and motion [CFCP$^+$06c]. Some authors distilled this 3D information to produce higher semantic unities relying on grouping regions that fulfill a uniformity criteria and afterward applying the analysis algorithm on these entities, as done in [CGH05]. A depiction of these features is shown in Figure 6.3.

6.1.3 Body Analysis

Information about the structure and dynamics of the human body can be of great benefit to analyze the input data. We can categorize the existing HMC and analysis algorithms into two groups depending on the usage of an implicit human body model: model-free and model-based analysis.

- **Model Free Analysis**: One approach to the analysis of human movements has been to omit the underlying human body model and to describe motion in terms of simple low-level features. Motion features, in both 2D [BD01, MB06] and 3D [CFCP06b, CFCP$^+$06c], have been employed to recognize a small set of actions using simple classifiers. Learnt statistics over detected edges describing a person have been employed to analyze human gestures [SB01]. Although these techniques are suitable for real-time systems since they have a low computational demand [CCU$^+$05], the range of applications they may address is limited.
- **Model Based Analysis**: Taking into account the underlying structure of human body, an analysis system can exploit it and be able to resolve occlusions and output more accurate results. A myriad of models have been presented in the literature [GD96, DBT03, MHK06]. Those models may be classified in two categories at a very general stage:
  - **Non Articulated Model:** Information about the degrees of freedom of the joints of a human body are not taken into account thus resulting in a model with fixed limb positions. Even though this approach may not be able to capture all types of human motion, it has been useful for tracking and other simple body analysis tasks as shown in [Lan06] applied for person tracking.
  - **Articulated Model:** Taking into account the dynamic constraints and range of movements a human body may achieve allow HMC algorithms to better analyze the input data towards estimating a correct pose. Nowadays, many researchers opt for this direction and many articulated models are employed in the literature, ranging from very simple to highly detailed ones. The research presented within this thesis is placed within this category.

Once a human body model is chosen for a given analysis problem, an algorithm able to accommodate this model to the input data cues and generate a meaningful spatial information is required. We can group the analysis algorithms in two main classes depending on the origin of the employed features marker and markerless. The first rely

on detecting some markers placed on the joints of the person and process these data to estimate a valid pose, while the markerless approach is based on fitting the given human body model to a set of features that have been extracted directly from the images. Methods for human body model fitting, both using the marker-based or markerless approach, are numerous and heterogeneous. State-of-the-art on these technologies will be further covered in the following specific sections: problem formulation in §6.2, an overview of the human body models which are common to all techniques in §6.3, marker based approaches in §6.4, and markerless approaches in §6.5.

6.1.4 Application-High semantic level analysis

Human body tracking has a wide spectrum of applications. Film industry has been pioneering the need for human motion capture since the emergence of realistic computer graphics. The capture and re-targeting of the movements of actors to animated characters or puppets is a commonly used technique in films, but also in video games and entertainment applications.

Recognition of gestures has been used to re-shape human-computer interfaces towards a communication paradigm where the user interacts with machines in a more natural manner. Examples of this technology are the smart environments [CHI07] and tele-assistance systems for disabled people [CGPV05].

The arising interest in security issues has been one of the influential application areas that have boosted developing very diverse techniques aiming at the analysis of human body motion in video sequences. Surveillance applications [HHD00] demand very robust real-time tracking techniques to enable fast distinction between simple authorized and non-authorized activities or recognition of suspicious behaviors in indoor and outdoor environments. Gait is an informative cue for person recognition and can be used in vision surveillance systems. However, extracting precise and robust human body poses from video streams using markerless algorithms is still an open problem [BHP05].

In the medical domain, full human body tracking has provided useful cues to assess bio-mechanical pathologies and gait disorders in a non-intrusive way. Dockstader *et al.* [DBT03] addressed this problem in a multi-camera environment analyzing the subject's gait and searching for anomalies during the walking cycle. Very detailed extraction of motion parameters with the aid of markers [CPF03] has been found useful for sportsmen to locate their weaknesses, and improve their performances.

6.2 Monte Carlo Based Human Motion Capture

Once a set of features related to the human body to be analyzed are available, we should analyze the most suitable method to extract the pose of the body from them. In this section, we formulate the problem and discuss the available alternatives.

6.2.1 Problem formulation

The temporal evolution of a physical articulated structure can be better captured with model-based tracking techniques. In this process, the defining parameters of a model

are sequentially estimated over time using video data from a number of cameras. The articulated structure can be fully described by a state vector $\mathcal{X} \in \mathbb{R}^D$ that we wish to estimate, where D is the dimension of the vector. Two approaches are found in the literature to define the state vector encoding the pose of an articulated structure: 3D locations and joint angles. The latter have been extensively used [AT04, BMP04, DR05] in comparison with 3D locations [ODE+07] since angles are a more natural way to encode an articulated structure. The state vector can be enlarged by adding the derivatives, velocity ($\dot{\mathcal{X}}$) and acceleration ($\ddot{\mathcal{X}}$), of the defining parameters. Typically, some variables within this vector exhibit cross dependencies and can not be considered independent one to another.

From a Bayesian perspective, the articulated motion capture and tracking problem is to recursively estimate a certain degree of belief in the state vector $\mathbf{y}_t \in \mathcal{X}$ at time t, given the history of observations $\mathbf{z}_{1:t}$ described in §6.1.2. Hence, it is required to estimate the posterior *pdf* function $p(\mathbf{y}_t|\mathbf{z}_{1:t})$. For the problem of HMC, this function is usually peaky and may present several local maxima and minima thus rendering linear and gradient based methods unsuitable for this task [MH03].

6.2.2 Particle filtering

Particle filtering (PF) introduced in §3 is an appropriate technique to deal with problems where the posterior distribution is multimodal. This usually happens when state space dimensionality is high, like in body motion tracking. However, to maintain a fair representation of $p(\mathbf{y}_t|\mathbf{z}_{1:t})$, a certain number of particles is required in order to find its global maxima instead of a local one. It has been proved in [MI00] that the amount of particles required by a standard PF algorithm [IB98] to achieve a successful tracking follows an exponential law with the number of dimensions. Articulated motion tracking typically employs state spaces with dimension $D \sim 25$ thus normal PF turns out to be computationally unfeasible.

There exist several possible strategies to reduce the complexity of the problem based on refinements and variations of the seminal PF idea. MacCormick *et al.* [MI00] presented partitioned sampling as a highly efficient solution to this problem. However, this technique imposes a linear hierarchy of sampling which may not be related to the true body structure assuming certain statistical independence among state variables. Hierarchical sampling presented by Mitchelson *et al.* [MH03] tackles the dimension problem by exploiting the human body structure and hierarchically explore the state space. In the instance when there exists a tractable sub-structure between some variables of the state model, specific states can be marginalized out of the posterior, leading to the family of Rao-Blackwellized PF algorithms [DFG01]. This technique has been applied to articulated motion tracking by Madapura *et al.* [MB07] even though limited experimental evidence with motion parallel to the camera has been presented. In [SC07], a clipping of the likelihood is presented as a technique to rapidly concentrate particles on the main modes of the weighting function but limiting its recovery from loss of track. Finally, annealed PF presented by Deutscher *et al.* [DR05] is one of the most general solutions to the problem of dimensionality. This technique employs a simulated annealing strategy to concentrate the particles around the peaks of the likelihood function by propagating

particles over a set of progressively smoothed versions of the likelihood functions thus avoiding getting trapped in local maxima (check §3.2.1 for a detailed review on this technique).

Other techniques rely on applying previously learnt information about the dynamics of the executed motion in order to place the particles more efficiently in the state space and, therefore, improve the efficiency of the system. Moreover, assuming that the subject is performing a specific action, it is possible to reduce the range of movements of the articulated structure. Low dimensional latent models in the framework of PF presented by Urtasun *et al.* [UFF06], employing a learnt reduced state space in order to manage the complexity of the problem. A similar approach is followed by Raskin *et al.* [RRR08] using latent spaces together with an annealed PF leading to promising results. Markov models have been proposed by Caillette *et al.* [CGH05] as a propagation model when tracking motion that has been previously analyzed. However, due to the restricting assumptions imposed in the type of motion executed by the subject, the resulting systems are not capable of tracking general unconstrained human motion.

As it has been mentioned in Chapter 3, the main issue in the implementation of a PF will be the definition of a meaningful likelihood function relating the measurements with the state space variables and the propagation of the state variables along time iterations. In the presented systems, these two factors are analyzed taking into account the particularities of the employed input data.

6.3 Modelling a Human Body

Modelling a human body implies first defining an articulated 3D structure able to represent the human body bio-mechanical features and, secondly, adopting a mathematical model to govern the movements of such articulated structure. This model should support the tracking of a broad variety of motion/poses while being adaptable to different human shapes. At the same time, the parameter set for describing the pose should be kept small as each additional parameter increases the dimensionality of the problem.

The level of detail of a human body model (HBM) is a design parameter and will be conditioned by the degree of accuracy we would like to achieve. While the major limbs such as arms, legs and head are necessary, other articulated parts like fingers might not be compatible with the achievable level of detail. In our case, the maximum data resolution of the 3D reconstruction is $s_{\mathcal{V}} = 2$ cm thus posing a limitation and making the tracking of small body parts impossible. This choice of the maximum data resolution is motivated by the exponential complexity increase of the 3D reconstruction algorithm with reference to $s_{\mathcal{V}}$ (see §2.2.4).

6.3.1 Human Body Model in the Literature

Several articulated representations and mathematical formalisms have been proposed in the literature to model both the structure and movements of a human body. The most common type of HBM is a hierarchy of bones (analogy with the term “skeleton”) and joints. The kinematic model is then a tree with a root usually placed at the pelvis, as

illustrated in Figure 6.5. The number of degrees of freedom (DoF) assigned to every joint will define the complexity of the HBM.

HBMs found in the literature range from very simple ones involving few DoF [CFCP06b] to very detailed [UFF06]. Typically, HBM complexity varies between 20 and 32 DoF and includes only the main limbs (torso, legs, arms and head), already amounting to between 16 and 20 dimensions. When adding the 6 dimensions of the root of the tree describing the global position and rotation, it is easy to realize that even these basic models represent a challenge for tracking. HBMs including the main limbs are widely employed in the literature [Mik03, CGH05]. It must be noted that, in analysis applications, the selected HBM is bound to the resolution of the input data: fingers, ankles or wrists are indeed impossible to discern in many sequences.

6.3.2 Parameterization of the joints

Defining a HBM involves the encoding of a number of successive rotations and translations that will represent the flexions of the articulations and the length of bones, respectively. The way to encode such information is not unique. The available techniques attempt to encode the one, two or three degree-of-freedom (DoF) existing in the human joints and their inter-dependencies to more faithfully model the kinematics of the HBM.

Parameterizing rotations is problematic mainly because rotations are non-Euclidean in nature. Perhaps, the more widely used are Euler angles that attempt to parameterize the non-Euclidean space by an Euclidean one by means of successive rotations about one particular axis. The main pitfall of Euler angles is that they may incur in singularities for specific rotation values: when two of the three rotation axes align, one axis could override the rotation in the other, effectively loosing a DoF. This effect is known as “gimbal lock” and can be mitigated in some cases by enforcing angular limits on the legal range of motion for Euler angles [Gra98].

Some authors [CGH05] define HBMs that are not valid in the real world, that is impossible DoF are assigned to some joints in order to decrease the complexity of the problem. Typically, the 3 DoF assigned to the shoulder may eventually lead to “gimbal lock” situations thus impairing the performance of the algorithm. Instead, re-distributing 1 DoF from the shoulder to the elbow helps avoiding such situations at the cost of assigning a flexion not achievable in such joint.

Unit quaternions [Hor87] are an elegant way to tackle the gimbal lock problem by encoding any arbitrary 3D rotation as a hyper-sphere in a 4D space. The 3D rotation encoded by a quaternion is equivalent to a single rotation around an axis which changes with the quaternion. The absence of fixed rotation axes poses the problem of data interpretation and constraints enforcement. In this field, Villa-Uriol [VU05] uses unit quaternions to drive the animation of an avatar in 3D with applications to motion capture and Herda *et al.* [HUF05] use quaternions to represent 3D DoF, and learn an implicit valid subspace from motion capture data.

Joint	Description	ω	θ^-	θ^+	l/height†
$J^x_{0,0}$	Neck left/right	X	$-\pi/2$	$\pi/2$	0.16
$J^y_{0,0}$	Neck front/back	Y	$-\pi/2$	$\pi/2$	–
$J^x_{1,0}$	Shoulder R. up/down	X	$-\pi$	π	0.20
$J^y_{1,0}$	Shoulder R. twist	Y	$-3\pi/4$	$\pi/2$	–
$J^z_{1,0}$	Shoulder R. front/back	Z	$-\pi/2$	π	–
$J^z_{1,1}$	Elbow R.	Z	0	π	0.20
$J^x_{2,0}$	Shoulder L. up/down	X	$-\pi$	π	0.20
$J^y_{2,0}$	Shoulder L. twist	Y	$-\pi/2$	$3\pi/4$	–
$J^z_{2,0}$	Shoulder L. front/back	Z	$-\pi$	$\pi/2$	–
$J^z_{2,1}$	Elbow L.	Z	$-\pi$	0	0.20
$J^x_{3,0}$	Hip R. left/right	X	$-\pi/2$	$\pi/3$	0.26
$J^y_{3,0}$	Hip R. front/back	Y	$-\pi$	$\pi/2$	–
$J^z_{3,0}$	Hip R. twist	Z	$-\pi/4$	$\pi/4$	–
$J^y_{3,1}$	Knee R. front/back	Y	0	π	0.24
$J^x_{4,0}$	Hip L. left/right	X	$-\pi/3$	$\pi/2$	0.26
$J^y_{4,0}$	Hip L. front/back	Y	$-\pi$	$\pi/2$	–
$J^z_{4,0}$	Hip L. twist	Z	$-\pi/4$	$\pi/4$	–
$J^y_{4,1}$	Knee L. front/back	Y	0	π	0.26
J^x_5	Torso left/right	X	$-\pi/4$	$\pi/4$	0.33
J^y_5	Torso front/back	Y	$-\pi/4$	$\pi/2$	–
J^z_5	Torso twist	Z	$-\pi/8$	$\pi/8$	–

† Results obtained after measuring 25 individuals.

Table 6.1: *Description of all joints employed to parameterize the selected human body model, including the rotation axis ω and the angular range $[\theta^-, \theta^+]$. The relative length with respect to the height of the person of every limb whose origin is associated to a given joint is also presented.*

Exponential maps

Exponential maps (also called twists) is a well known technique employed in robotics [MSZ94] that allows encoding rotations and translations in a compact framework. They have been extensively used in motion tracking since do not suffer from singularities, like quaternions. Bregler *et al.* [BM98, BMP04] pioneered the usage of exponential maps in HBM tracking using an optimization process to estimate the HBM pose. Mikič [MSJ00] performed inverse kinematics on a HBM parameterized with twists and exponential maps together with a Kalman filter to track a HBM in 3D. Exponential maps are chosen to be the formulation of rotations and translations in this work since they provide a singularity-free representation and are easier to interpret than quaternions.

This technique allows us to define an exponential map matrix [MSZ94], or also called roto-translation transformation [CPF03], as a 4×4 matrix Λ_k^{k+1} that transforms the coordinate system of a given joint k to the coordinate system of joint $k+1$. In this way,

assuming that all defining parameters of a HBM are known, we can express the position of a given point $\mathbf{p}_k$ by means of the so called forward kinematic equation:

$$\mathbf{p}_k = \Lambda_{k-1}^{k} \cdot \Lambda_{k-2}^{k-1} \cdots \Lambda_{0}^{1} \mathbf{p}_0 = \left(\prod_{d=1}^{k} \Lambda_{d-1}^{d} \right) \mathbf{p}_0, \tag{6.1}$$

where $\mathbf{p}_0$ is the origin of the real world. For more details check Appendix A.

6.3.3 Linking PF with a HBM

Once the HBM that will be employed is chosen, we can straightforwardly define how to relate it with the already presented PF theory. Basically, we construct our state space $\mathcal{X}$ as the concatenation of all defining parameters of the HBM, and every particle $\mathbf{y}_t^j$ will be an instance of $\mathcal{X}$. Hence, a particle encodes a possible pose of the HBM and the associated weight encodes its likelihood as depicted in Figure 6.4(a) and Figure 6.4(b). Applying the annealing PF process to a HBM will result on progressively concentrating the particles around the main mode of the likelihood function as shown in Figure 6.4(c).

Information about the centroid of the person in the xy plane obtained in Chapter 4 is employed to initialize the state space variables corresponding to the root position of the HBM in this location. Temporal evolution of the centroid for few frames is used to estimate the root orientation assuming that the subject does not move backwards. The remaining variables of the state space are set to a neutral position when initializing the HBM.

6.3.4 Our HBM choice

Let us define a HBM as the set $\mathcal{H}$ formed by a root part (torso) denoted as $\mathcal{T}$ and a set of $N_{\mathcal{L}}$ open kinematic chains that will define the body limbs (wide sense), that are head, arms and legs. Each limb will be formed by a variable number of parts (links in this kinematic chain) denoted as $\mathcal{P}$. Hence,

$$\mathcal{H} = \{\mathcal{T}, J_{\text{Torso}}, \mathcal{P}_{i,j}, J_{i,j}\}, \qquad 1 \leq i \leq N_{\mathcal{L}}, 1 \leq j \leq N_{\mathcal{P}(i)}, \tag{6.2}$$

where $N_{\mathcal{P}(i)}$ stands for the number of parts in the i-th limb. The torso, limbs and their sub-parts are connected to one another by means of joints, $J_{i,j}$. Joint J_{Torso} is a particular case standing for the torso rotation on the pelvis. It must be noted that this rotation and the rotation of the whole body with respect to the real world is applied at the same point. While the global rotation affects all the body, the pelvis rotation allows rotating the upper body without affecting the legs.

In order to constrain the possible poses that this HBM may adopt to be valid, we define a number of DoF and an angular range at each joint as shown in Table 6.1. Lengths of body parts, $\mathcal{P}_{i,j}$, $\forall i, j$, are set in a linear manner proportionally to the height of the subject as suggested in [DBT03]. Although this assumption may collapse for individuals out of a standard height (between 160 and 190 cm), it provided sufficient accuracy for our experiments. The selected body model $\mathcal{H}$ is illustrated in Figure 6.5.

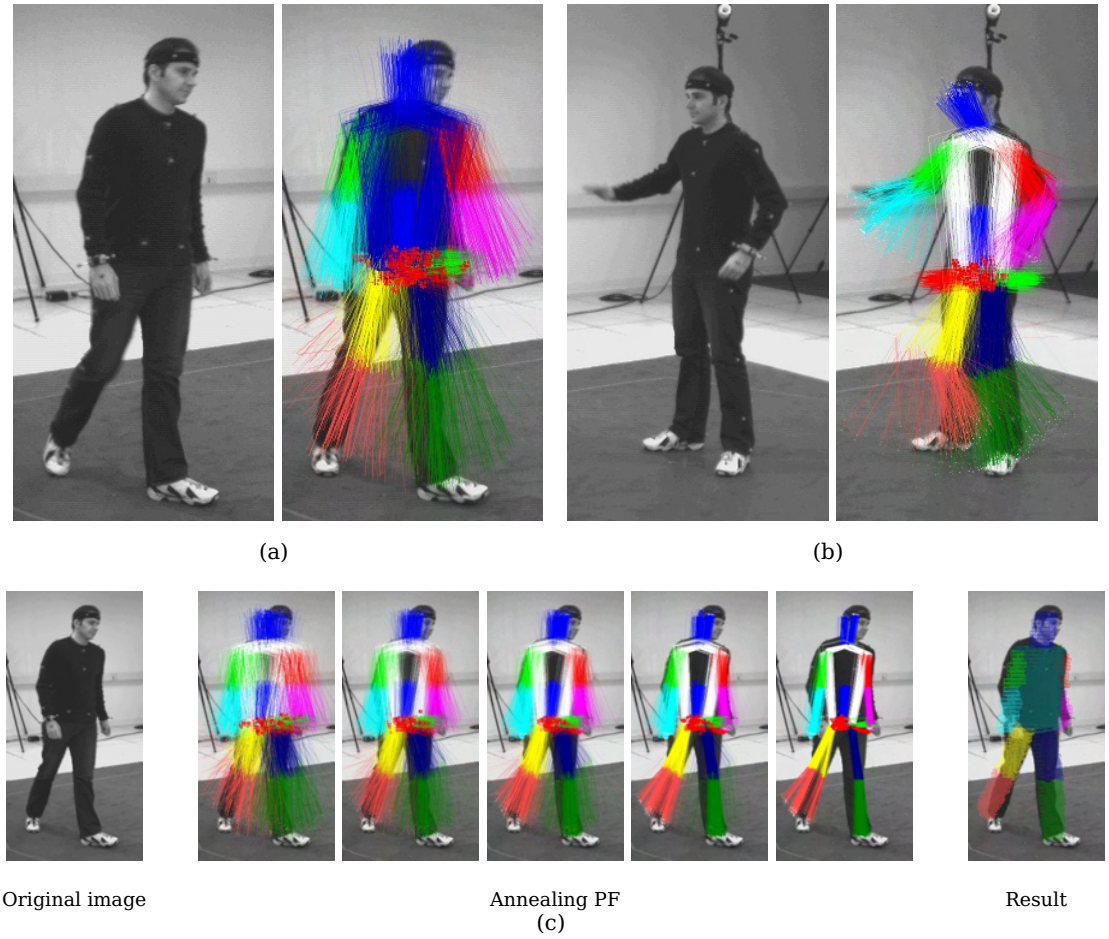


Figure 6.4: *In (a) and (b), two examples of particle sets ($N_p = 500$) populating the state space. Every limb is plot in different colors for visualization purposes and the intensity of every pose is proportional to the associated weight. In (c), an example of the annealing PF algorithm.*

The proposed model has its root position in the pelvis and adds up to 21 DoF distributed as follows: 3 DoF at each shoulder, 1 DoF at the elbows, 3 DoF at the hips, 1 DoF at the knees and 3 DoF at the waist. This last joint allows rotations of the body upper part without affecting the hips and legs positions. Apart of these DoF, we must consider the translation and rotation of the root with respect to the world, resulting in 27 DoF associated to $\mathcal{H}$.

Kinematic restrictions imposed by the angular limits at each joint may produce a more robust tracking output. In this field, some methods rely on modelling the angular cross-dependencies among joints [HUF05] or learning dynamic models associated to a given action [CGH05, RRR08]. In our case, these angular constraints will be enforced in the propagation step of the PF scheme. As it has already been presented in §3.1.2.2, the propagation step consists in adding a random component to the state vector of a particle

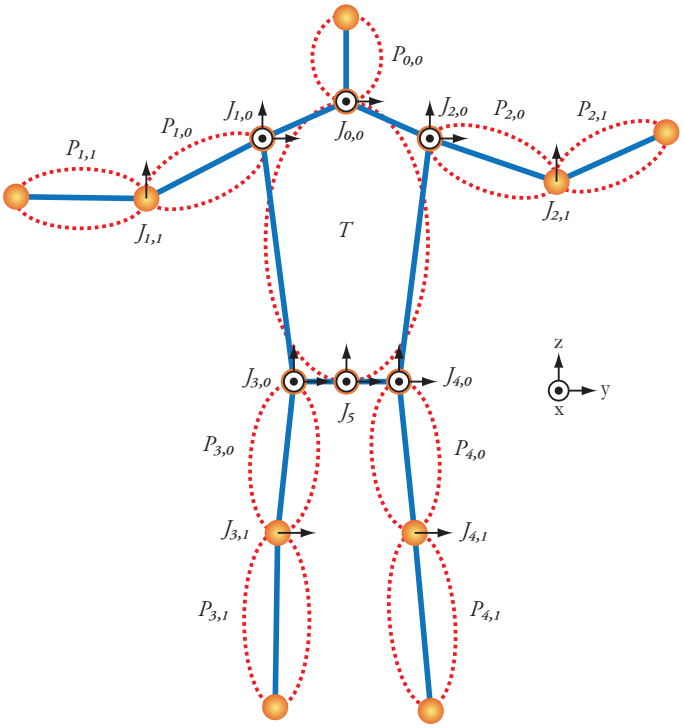
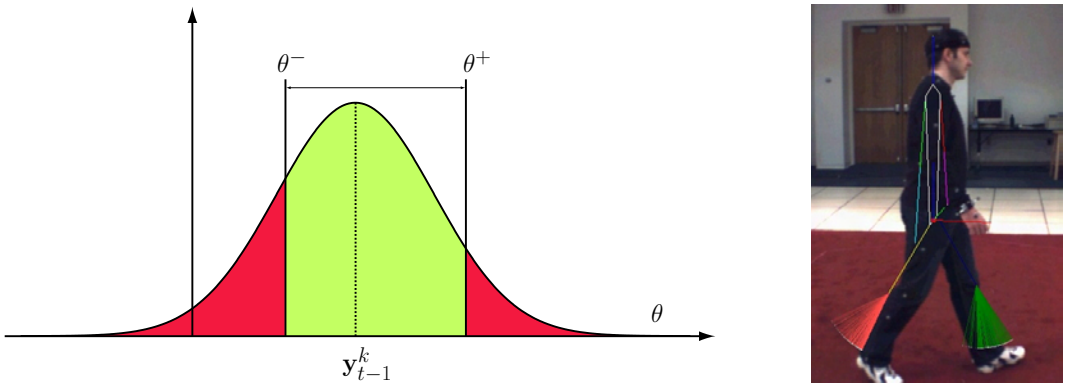


Figure 6.5: *Full articulated human body model employed in this chapter. Every joint has a number of DoF associated, adding up to 21 plus the 6 DoF that describe the position and rotation w.r.t. the real world.*

as:

$$\mathbf{y}_t^k = \mathbf{y}_{t-1}^k + \mathcal{N}(\mathbf{0}, \Sigma) = \mathcal{N}(\mathbf{y}_{t-1}^k, \Sigma). \tag{6.3}$$

This propagation may lead to poses out of the legal angular ranges displayed in Table 6.1. In order to avoid such effect, Husz *et al.* [HW07] add a term into the likelihood function that penalizes particles when not fulfilling the angular constraints. We present the following alternative: when propagating particles, we take into account the angular constraints and generate samples of a truncated Gaussian distribution, denoted as $\mathcal{N}^\star$, instead of a complete Gaussian distribution, as shown in Figure 6.6(a). In this way, particles are generated always within the allowed ranges. Generation of samples from such a truncated Gaussian distribution is indeed a non-trivial problem [PP02] and it may be even impossible to obtain a closed form. In order to generate a sample from such truncated Gaussian *pdf*, we first draw a sample from $\mathcal{N}(\mathbf{y}_{t-1}^k, \Sigma)$ and check whether it falls within the valid angular range. If this condition is not fulfilled, the sample is dismissed and redrawn. Despite the simplicity (and inefficiency) of the employed technique, the overall performance of the system was improved since it is computationally more expensive to compute the likelihood of an invalid pose (out of the angular limits) and then dismissing it (Husz *et al.* [HW07] approach) than generating valid poses in the state space (despite using an inefficient method). Within the same scope, a detailed analysis of the angular



(a) (b)

Figure 6.6: *Angular constraints enforcement by propagating particles within the allowed angular ranges $[\theta^-, \theta^+]$. In (a), samples are propagated following a truncated Gaussian distribution $\mathcal{N}^\star$ centered at $\mathbf{y}^k_{\mathbf{t}-1}$ with covariance matrix $\Sigma = \sigma$ bounded between θ^- and θ^+ (green zone). In (b), an example of particle propagation in the knee angle displaying how propagated particles never fall out the legal ranges ($\theta < 0$).*

relations among joints [HUF05] may allow dealing with the problem of inter-penetration among limbs at the cost of collecting a large set of annotated human motion data.

6.4 Marker Based Tracking

Accurate retrieval of an articulated structure from the information provided by multiple cameras is a field that found numerous applications in the recent years. The grown of computer graphics technology together with HMC[2] systems have been extensively used by the cinematographic and video games industry to generate virtual avatars, fantastic characters, etc. Medicine also benefited from these advances in the field of orthopedics, assess locomotive pathologies, sports performance improvement, etc. However, all these applications require accurate input data to work and, nowadays, only HMC systems aided by intrusive gadgets may produce the desired degree of accuracy.

Depending on the markers used, the motion capture systems are classified in non-optical (inertial, magnetic and mechanic) or optical systems (active and passive). On one hand, non-optical systems require expensive and dedicated hardware to produce highly accurate results. Inertial motion capture is based on miniature sensors in a special suit, bio-mechanical models and sensor fusion algorithms [Roe06, Mov]. Magnetic systems calculate position and orientation of the markers by the relative magnetic flux of three orthogonal coils, but movements are usually limited by the wiring. Finally, mechanical systems also use special suits with skeletal-like structures that move with the body movements [KOF05]. In the other hand, optical systems are the most widely used and

[2]Although we restrict our research to human motion capture (HMC), the more general term motion capture (MoCap) is widely applied to such systems.

they are based on photogrammetric methods and give complete freedom of movement. They require a high number of cameras (typically more than 7 cameras) and/or a high frame rate (typically 60-120 Hz) to produce an accurate output in form of a set of the 3D positions corresponding to the markers attached to the performer's body. The capture system infers the time-varying location in the 3D space of the markers by triangulation based on the projection of the markers onto each camera's image plane. We can distinguish between the systems using passive and active markers. The more usual are passive markers, that use a retroreflective material to reflect light back that is generated near the cameras lens [Vic]. The active systems triangulate positions by illuminating LED markers that can also be uniquely identified by pulse modulation.

This section focuses in the HMC systems with passive markers in a multi-camera scenario. These systems first require an accurate reconstruction of the markers' 3D position from its 2D projections which is not a trivial problem. Matches need to be established between the detected markers in the different views, defining the multiple view correspondences through homographies or algebraic methods. This process is prone to errors due to occlusions, detection noise and the proximity between markers. A temporal tracking of the markers also needs to be performed, to identify the markers in each sequence frame, thus yielding a 3D trajectory for each marker. Although professional systems exist for this purpose, errors occur when crucial markers become occluded or when markers' trajectories are confused. Most applications finally require the transformation of the markers localization and trajectories to the motion parameters of a kinematic skeleton model. Commercial tools that perform this transformation are generally semi-automatic, and thus this is also a labor-intensive task and prone to errors.

In most commercial systems, the estimation of the markers' 3D position and the fitting of the HBM are decoupled. One of the first attempts to use an anatomical human model for increasing the robustness of a MoCap system was presented by Herda *et al.* [HFP$^+$01]. Their system computes a skeleton-and-marker model using a standardized set of motions and uses it to resolve the ambiguities during the 3D reconstruction process. Another approach using a body model was presented by Kirk *et al.* [KOF05]. In this system, 3D markers are first clustered into segment groups, then the topological connectivity among these groups is determined, and finally the position of their connecting joints is computed. In [ATS06], the system also works in various steps: first, it identifies individual rigid bodies by means of a variant of spectral clustering. Thereafter, it determines joint positions at each time step of motion through numerical optimization, reconstructs the skeleton topology, and finally enforces fixed bone length constraints. Detection of 2D markers in separate images and its analysis using calibration information has been used in [GF05] enforcing a HBM afterwards. A similar technique using a Kalman filter involving the HBM in the data association step was presented by Cerveri *et al.* [CPF03].

In this section we propose a low cost marker based motion capture system. It tackles the correspondence problem together with the instantiation of the skeleton model and the tracking, increasing robustness. In our system, only the 2D marker detection is decoupled from the rest of the system, thus performing the spatial correspondence, tracking and model fitting in a single framework. The system can work with any mark-

ers detectable onto a set of 2D planes under perspective projection and, as it will be shown, it is robust to markers' occlusion and noisy detections. The presented algorithm is intended to work with any multi-camera setup and regardless of the complexity of the selected human body model. As commented in §6.2 and §6.3.3, an annealed PF will be employed, being the measurement and likelihood function the only defining factors of the algorithm. These factors are discussed in the following section.

6.4.1 Filter implementation

6.4.1.1 Measurement

For a given frame in the video sequence, a set of N_{C} images are obtained from the N_{C} cameras. Each camera is modeled using a pinhole camera model based on perspective projection. Accurate calibration information is available. The input data $\mathbf{z}_t$ to our tracking system will be the 2D projection of the set of distinguishable markers attached to the body of the performer onto these N_{C} images. Let $\mathcal{D}_n = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_{Q_n}\}$ be the set of Q_n locations detected in the image captured in the n-th view, $\mathbf{I}_n$, $1 < n \leq N_{\mathrm{C}}$. Ideally, this set would contain the 2D projections of the markers that are not affected by the occlusions produced by the body itself onto the n-th camera view. In order to generate $\mathcal{D}_n$, a marker detection algorithm $\Gamma : \mathbf{I}_n \rightarrow \mathcal{D}_n$ is employed whose performance is assessed by the detection rate ($\overline{DR}$), the false positive rate ($\overline{FP}$) and the variance estimation error (σ_Γ^2). This generic formulation of Γ will allow performance comparisons of the tracking algorithm when using different marker detection algorithms.

Markers are usually placed at the joints, the end of the limbs, the top of the head and the chest of the subject. In this work, some experiments were conducted using little yellow balls as body markers thus a color-based marker detection algorithm was employed to retrieve their 2D positions. Nevertheless, the proposed method is general enough to be applied to any type of markers detectable onto a set of 2D planes under perspective projection. An example of the detections obtained by the color-based marker detection is shown in Figure 6.7.

6.4.1.2 Likelihood Evaluation

In order to evaluate the likelihood between the body pose represented by a given particle state $\mathbf{y}_t^j \in \mathcal{X}$ with reference to the input data $\mathbf{z}_t = \{\mathcal{D}_n\}_{n=1}^{N_{\mathrm{C}}}$, a fitness function $w(\mathbf{z}_t, \mathbf{y})$ should be defined.

The M 3D positions of the HBM landmarks (the joints and the end of the limbs) corresponding to the pose described by the state variable $\mathbf{y}$ are computed. Let us denote these coordinates as the set $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$, $\mathbf{x}_m \in \mathbb{R}^3$. The fitness function relating the 3D locations set X with the 2D observations $\{\mathcal{D}_n\}_{n=1}^{N_{\mathrm{C}}}$ should measure how well these 2D points fit as projections of the set X. We have tackled a similar problem in [CFCP05b, CFCTP05] in a Bayesian framework and the underlying idea is applied in this context.

For every element $\mathbf{x}_m$ from the set X, we compute its projection onto every camera as

$$\mathbf{p}_{m,n} = P_n(\mathbf{x}_m), \qquad 1 \leq m \leq M, \quad 1 \leq n \leq N_{\mathrm{C}}, \tag{6.4}$$

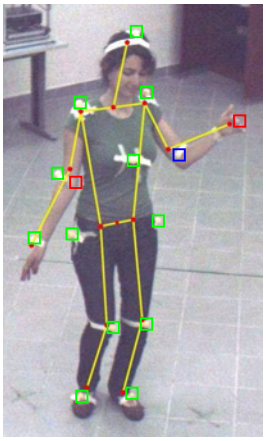


Figure 6.7: *Measurement example. The output of the employed color based marker location detection algorithm. Colors describe the correct detections (green), the miss detections (blue) and the false positive detections (red). All this detections will conform the measurement set* $\mathcal{D}_n$.

where $P_n(\cdot)$ is the perspective projection operator from 3D to 2D on the view n [HZ04] (see §2.2.2 for more details). Then, the set $T_m = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_{N_C}\}$ containing the closest measurement in every camera view associated to every HBM landmark $\mathbf{x}_m$ is constructed as follows:

$$\mathbf{t}_n = \min_{\mathbf{d}_q} \|\mathbf{p}_{m,n} - \mathbf{d}_q\|, \quad \mathbf{d}_q \in \mathcal{D}_n, \quad \forall n. \tag{6.5}$$

However, not all the 3D points $\mathbf{x}_m$ may have a projection onto every view due to occlusions or a miss detection of the marker detection algorithm. In order to detect such cases, a thresholding is applied to the elements $\mathbf{t}_n$ dismissing those measurements above a threshold ρ. In this case, $\mathbf{t}_n = \emptyset$. At this point, we need to measure how likely are the set of 2D measurements T_m to be projections of the 3D HBM landmark $\mathbf{x}_m$. This can be done by means of the generalized symmetric epipolar distance $d_{\mathcal{SE}}(\cdot)$ presented in [CFCP05b].

Let $L\left(\mathbf{p}^i, j\right)$ be the epipolar line generated by the point $\mathbf{p}$ in a given view i onto another view j. Symmetric epipolar distance between two points $d_{\mathcal{SE}}(\mathbf{p}^i, \mathbf{p}^j)$, in the two views i,j, is defined as:

$$d_{\mathcal{SE}}(\mathbf{p}^i, \mathbf{p}^j) \triangleq \sqrt{d^2(L(\mathbf{p}^i, j), \mathbf{p}^j) + d^2(L(\mathbf{p}^j, i), \mathbf{p}^i)}, \tag{6.6}$$

where $d(L(\mathbf{p}^i, j), \mathbf{p}^j)$ is defined as the Euclidean distance between the epipolar line $L\left(\mathbf{p}^i, j\right)$ and the point $\mathbf{p}^j$ as depicted in Figure 6.8. It has been shown in [CFCP05b] that the extension of the symmetric epipolar distance for $k \geq 2$ points (in k different views) $d_{\mathcal{SE}}(\mathbf{p}^1, ..., \mathbf{p}^k)$ can be written in terms of the distance defined in Eq.6.6 as:

$$d_{\mathcal{SE}}(\mathbf{p}^1, ..., \mathbf{p}^k) = \sqrt{\sum_{i=1}^{k-2} \sum_{j=i+1}^{k-1} d^2_{\mathcal{SE}}(\mathbf{p}^i, \mathbf{p}^j)}. \tag{6.7}$$

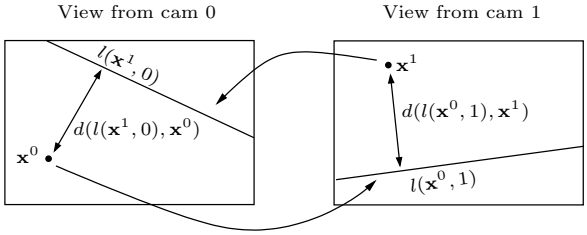


Figure 6.8: *Symmetric epipolar distance between two points* $d(\mathbf{x}^0, \mathbf{x}^1)$.

This distance produces low values when the 2D points are coherent, that is when they are projections from the same 3D location. The score s_m associated to T_m, and therefore to $\mathbf{x}_m$, is defined as:

$$s_m(\mathbf{z}_t, \mathbf{x}_m) \equiv s_m(\mathbf{z}_t, T_m) \propto d_{\mathcal{SE}}(T_m), \tag{6.8}$$

and normalized such that $s_m(\mathbf{z}_t, T_m) \leq 1$. In the case where the non-empty elements of T_m is below 2, the distance $d_{\mathcal{SE}}(T_m)$ can not be computed. Under these circumstances, we set $s_m(\mathbf{z}_t, T_m) = 1$.

Finally, the cost function $C(\mathbf{z}_t, \mathbf{y})$ is constructed as the average of the distances over the M HBM 3D landmark points:

$$C(\mathbf{z}_t, \mathbf{y}) = \frac{1}{M} \sum_{m=1}^{M} s_m(\mathbf{z}_t, \mathbf{x}_m). \tag{6.9}$$

The associated weighting function is defined accordingly (see §3.1.2.3):

$$w(\mathbf{z}_t, \mathbf{y}) = \exp\left(-\frac{C(\mathbf{z}_t, \mathbf{y})^2}{2\sigma^2}\right). \tag{6.10}$$

In our experiments, $\sigma = 1$ provided satisfactory results. Other values of this constant might only affect the speed of convergence of the APF algorithm, but not modifying the final estimation.

6.5 Markerless Based Tracking

Markerless HMC and tracking is a difficult problem which has been thoroughly researched by the computer vision community for the last two decades. The approaches to this problem are diverse but, despite this great deal of attention, the general problem remains unsolved. Nowadays, multi-camera approaches together with Monte Carlo based techniques are the mainstream direction when addressing this problem.

Fitting a HBM to a set of features directly extracted from images has been tackled by means of minimization methods. Gavrila *et al.* [GD96] are among the first to address the problem of tracking an articulated human body model using multiple views by searching for the best fit between detected edges and the projection of HBM on multiple views. Given the HBM and the multi-camera images or 3D reconstructions, the tracking process could also be formulated as a registration problem. A method based

on creating a set of physical forces to align the projection of the HBM and the contours of the input data is proposed by Delamarre and Faugeras by either using multi-camera images [DF99] or a set of reconstructed 3D points of the scene [DF01]. A similar force-based technique was applied to occluding contours in multiple images by Kakadiaris and Metaxas [KM00]. Stochastic meta-descent optimization was presented by Kehl and Gool [KG06] as an efficient technique to avoid local minima in a fitness function relating a 3D voxel reconstruction of the space with an articulated HBM. Another approach using a 3D representation of the space was presented by Borovikov *et al.* [BD02] where the optimal pose was retrieved by solving a convex optimization problem.

Monte Carlo based methods have been found particularly useful when tackling problems that involve dealing with multimodal functions and, therefore, markerless HMC has benefited from these techniques as mentioned in §6.2. As done in the marker based case, an annealed PF will be employed for markerless tracking where the measurement and likelihood function will be the defining factors of the algorithm.

In the recent years, there has been a new tendency in markerless HMC where this problem is no longer posed as an estimation but instead as a classification problem. Typically, a large annotated database with examples of all possible poses is collected and the HBM fit to the new data input is computed by searching over the analyzed database and selecting the pose that better resembles the input data. These techniques have been denoted as "example based pose estimation" [OS08] and usually rely on statistical mappings such as the sparse probabilistic regression presented by Urtasun *et al.* [UD08]. Nonetheless, the main drawback of these techniques is the preparation and availability of such large amount of annotated data and its inadequateness to track motion not present in the training dataset.

6.5.1 Filter implementation

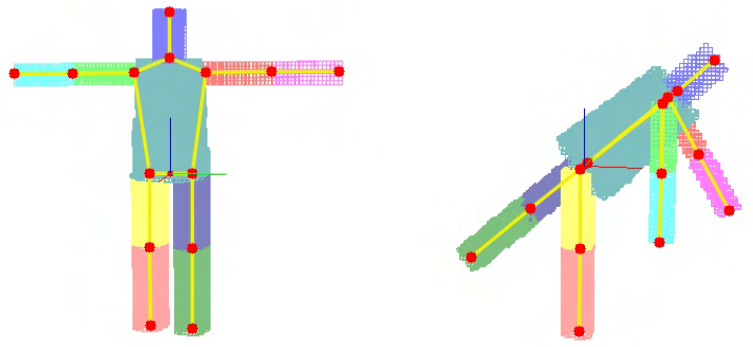
6.5.1.1 Measurement

In order to define a meaningful measure between the pose encoded by a given particle $\mathbf{y} \in \mathcal{X}$ and the available data $\mathbf{z}_t = \{\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{V}}^{\mathrm{C}}, \boldsymbol{\mathcal{V}}^{\mathrm{S}}\}$, we have to establish a relation between $\mathbf{y}$ and the 3D voxelized space. This can be achieved by defining an appearance model of the HBM, that is to "flesh out" the HBM skeleton with a volumetric model of the limbs, torso and head. Typically, this has been addressed by means of polyhedrons, ellipsoids, superquadrics, etc. In our particular case, we will use truncated cones in the 3D discretized space. Let us define the voxel representation of this fleshed HBM as the set $\boldsymbol{\mathcal{V}}_{\mathbf{y}}^{\mathrm{HBM}}$ related with the pose described by $\mathbf{y}$; Figure 6.9 depicts some examples.

In order to keep an affordable complexity when generating the $\boldsymbol{\mathcal{V}}_{\mathbf{y}}^{\mathrm{HBM}}$ set for every particle required by our algorithm, the following procedure has been devised. First, in the initialization phase of the algorithm, the body is set to the neutral position $\boldsymbol{\mathcal{V}}_{0}^{\mathrm{HBM}}$ shown in Figure 6.9(a). Then, the center of each voxel (in homogeneous coordinates) belonging to each body part is stored in a look-up table (LUT):

$$\mathrm{LUT}_{\boldsymbol{\mathcal{Y}}} = \{\mathcal{V}_{\mathbf{x}}, \mathcal{V} \in \boldsymbol{\mathcal{Y}}\}, \qquad \boldsymbol{\mathcal{Y}} \in \{\boldsymbol{\mathcal{V}}_{\mathcal{T}}, \boldsymbol{\mathcal{V}}_{\mathcal{P}_{i,j}}\}, \forall i, j, \tag{6.11}$$

where $\boldsymbol{\mathcal{V}}_{\mathcal{T}}$ and $\boldsymbol{\mathcal{V}}_{\mathcal{P}_{i,j}}$ are the voxels associated to the torso and body parts, respectively.



(a) (b)

Figure 6.9: *The relation between the set of kinematic chains that define a human body model and the real 3D voxelized space is achieved by adding a model of flesh to torso, head, arms and legs. In our case, 3D discretized truncated cones have been employed. In (a), the neutral pose $\mathcal{V}_0^{HBM}$ used in the initialization phase and, in (b), a pose obtained after applying a transformation to the voxel positions stored in the LUT. Note that used colors are only for display purposes.*

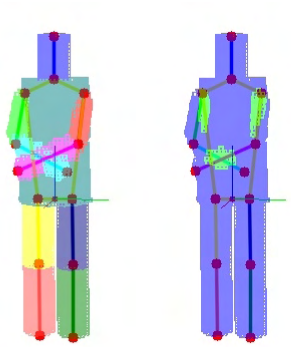
Essentially, for each body part, the LUT will be a matrix of size $4 \times |\mathcal{Y}|$, in homogeneous coordinates. Finally, when generating the set $\mathcal{V}_{\mathbf{y}}^{\text{HBM}}$ associated to a given particle $\mathbf{y}$, we only have to apply the forward kinematics equation Eq.6.1 to all elements in the LUT associated to every body part. This process involves multiplications and additions that can be efficiently implemented.

The set $\mathcal{V}_{\mathbf{y}}^{\text{HBM}}$ will allow us measuring the fitness of a given pose with respect to the input data $\mathbf{z}_t$. This set will be constructed by performing an union (with addition) among the individual volumes of the torso, $\mathcal{V}_{\mathcal{T}}$, and all limbs, $\mathcal{V}_{\mathcal{P}_{i,j}}$, $\forall i, j$, that is:

$$\mathcal{V}_{\mathbf{y}}^{\text{HBM}} = \biguplus_{\mathcal{Y} \in \{\mathcal{V}_{\mathcal{T}}, \mathcal{V}_{\mathcal{P}_{i,j}}\}} \mathcal{Y}, \qquad 1 \leq i \leq N_{\mathcal{L}}, 1 \leq j \leq N_{\mathcal{P}(i)}. \tag{6.12}$$

Operator $\uplus$ refers to the operation that assigns to each voxel of the 3D space the number of intersections among all body parts in that position, as shown in Figure 6.10.

Although color information ($\mathcal{V}^{\text{C}}$) was extensively used in Chapter 4 and allowed distinguishing among different targets, we noticed that there is not a high difference among colors of different body parts since the subjects under study often dressed in a single color. In some cases, if the subject wears different colors at the upper and lower body part, color might be helpful to track actions involving contact between legs and arms which is not found in our evaluation databases. In our case, the contribution of color information to the accuracy of the tracking algorithm was little in comparison with the computational cost required to generate the colored voxels (see §2.2.4). Therefore, only surface and occupancy information will be employed, disregarding color information.



(a) (b)

Figure 6.10: *HBM analysis based on the voxel set $\mathcal{V}_{\mathbf{y}}^{HBM}$. In (a), the pose under study depicted with false colors to differentiate body parts and, in (b), the real set $\mathcal{V}_{\mathbf{y}}^{HBM}$. Blue voxels stand for places with only one body limb occupying that space while green regions stand for places with two limbs occupying that space. Three limbs occupying the same region is an odd case and may be produced with very awkward poses.*

Raw voxel data measurement

According to the representation $\mathcal{V}_{\mathbf{y}}^{\text{HBM}}$ and the available raw voxel data $\mathcal{V}$, we may define the output, double occupancy and occupancy scores as:

$$\rho_{\mathcal{Y}}^{\text{Out}} = \frac{\#\left\{\mathcal{V} \in \mathcal{Y} | \mathcal{V}_{\mathbf{x}} \notin [\mathcal{V}_{\mathbf{y}}^{\text{HBM}}]\right\}}{|\mathcal{Y}|}, \tag{6.13}$$

$$\rho_{\mathcal{Y}}^{\text{DO}} = \frac{\#\left\{\mathcal{V} \in \mathcal{Y} | \mathcal{V}_{\mathbf{y}}^{\text{HBM}}(\mathcal{V}_{\mathbf{x}}) > 1\right\}}{|\mathcal{Y}|}, \tag{6.14}$$

$$\rho_{\mathcal{Y}}^{\text{Occ}} = \frac{\#\left\{\mathcal{V} \in \mathcal{Y} | \mathcal{V}_{\mathbf{y}}^{\text{HBM}}(\mathcal{V}_{\mathbf{x}}) \geq 1 \& \mathcal{V}(\mathcal{V}_{\mathbf{x}}) \neq 0\right\}}{|\mathcal{Y}|}, \tag{6.15}$$

$$\mathcal{Y} \in \left\{\mathcal{V}_{\mathcal{T}}, \mathcal{V}_{\mathcal{P}_{i,j}}\right\}, \forall i, j.$$

These set of measures will allow assessing the fitness between the pose $\mathbf{y}$ and the data $\mathcal{V}$. Outside score $\rho_{\mathcal{Y}}^{\text{Out}}$ will quantize the amount of voxels of a given body part that fall out of the analyzed scene. Sometimes, methods relying on a separate analysis of multiple images to estimate the HBM pose [DR05, RRR08] can not easily estimate whether the HBM body instance they are evaluating falls out of the scene. By using a 3D reconstruction, this problem can be straightforwardly addressed through the value of $\rho_{\mathcal{Y}}^{\text{Out}}$.

When tackling the HBM pose estimation through a PF approach, a number of random pose instances are generated, typically with defining parameters contained in $\mathbf{y}$ within the valid ranges. However, inter-penetration between limbs may occur and still be $\mathbf{y}$ within the legal values. Very few researchers explicitly mention this problem or rely on very detailed parameterizations among the angles of the joints to avoid it [HUF05]. Other approaches rely on combining information from image edges and occupancy to

avoid such cases [Che03, DR05]. By analyzing the figure given by $\rho_{\mathcal{Y}}^{\mathrm{DO}}$, inter-penetration among body parts can be efficiently measured. These two already presented figures will determine those regions of the state space $\mathcal{X}$ to be avoided since poses resulting in high values of $\rho_{\mathcal{Y}}^{\mathrm{Out}}$ and/or $\rho_{\mathcal{Y}}^{\mathrm{DO}}$ are likely to be invalid.

The occupancy score $\rho_{\mathcal{Y}}^{\mathrm{Occ}}$ measures the fraction of the body part that is occupied. Ideally, a good match will yield to low values of $\rho_{\mathcal{Y}}^{\mathrm{Out}}$ and $\rho_{\mathcal{Y}}^{\mathrm{DO}}$ and high values of $\rho_{\mathcal{Y}}^{\mathrm{Occ}}$, for every body part. Note that the measures $\rho_{\mathcal{Y}}^{\mathrm{Out}}$ and $\rho_{\mathcal{Y}}^{\mathrm{DO}}$ only depend on the body pose scoring its validity while $\rho_{\mathcal{Y}}^{\mathrm{Occ}}$ accounts for the likelihood between this pose and the observed data.

Surface data measurement

Edge information provides informative cues about the structure of an articulated object since they usually provide a good outline of visible arms and legs. This fact fueled its usage in most of the HBM tracking works: [GD96, MH03, DR05, RRR08]. In our case, we extended the usage of edges to 3D, thus employing surfaces to generate a score that measures the fitness of the HBM surface with the available surface data $\boldsymbol{\mathcal{V}}^{\mathrm{S}}$.

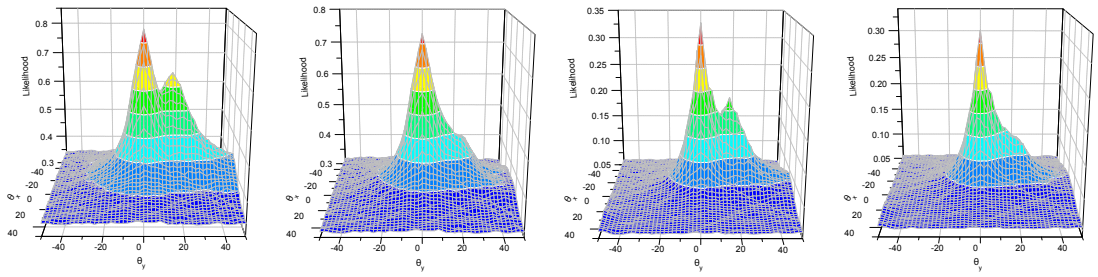
Surface data is first smoothed with a Gaussian mask and the obtained voxel values are re-mapped between 0 and 1. This produces a voxel map $\widetilde{\boldsymbol{\mathcal{V}}}^{\mathrm{S}}$ where, in each voxel, it is assigned a value related to its proximity to a surface. Finally, we can define the surface measurement as:

$$\rho_{\mathcal{Y}}^{\mathrm{Surf}} = \frac{1}{|\mathcal{Y}|} \sum_{\mathcal{V} \in \mathcal{Y}} \left(1 - \widetilde{\boldsymbol{\mathcal{V}}}^{\mathrm{S}}\left(\mathcal{V}_{\mathbf{x}}\right)\right), \qquad \mathcal{Y} \in \left\{\boldsymbol{\mathcal{V}}_{\mathcal{T}}^{\mathrm{S}}, \boldsymbol{\mathcal{V}}_{\mathcal{P}_{i,j}}^{\mathrm{S}}\right\}, \forall i, j. \tag{6.16}$$

6.5.1.2 Likelihood Evaluation

Given the obtained scores that quantize the fitness of a generic particle $\mathbf{y}$ with reference to the input data $\mathbf{z}_t$, we may construct the likelihood function that will assign a weight to that particle. Constructing a weighting function associated to a HBM is a problem that has been tackled in some ways in the literature, each of them making several assumptions. There are two main approaches: global and factorized likelihoods.

In both types of evaluation, information provided by the scores obtained from raw and surface voxels will be employed. Let us consider a simple case, Figure 6.11, to analyze the influence of the mentioned scores into a generic likelihood function. The multiple scores are combined using a multivariate normal function as defined in §3.1.2.3. In the first case, Figure 6.11(a), only occupancy and output scores (Eq.6.13 and 6.15) are employed leading to a clearly multimodal shape of the likelihood function. Double occupancy score (Eq.6.14) enforces regions of inter-penetrated limbs to be penalized hence some modes of the likelihood are lowered as seen in Figure 6.11(b). Information provided by the surface proximity score (Eq.6.16) allows sharpening the likelihood function since it is a very discriminative score as displayed in Figure 6.11(c) and 6.11(d). Combining the three scores into a common function provides a likelihood function with well localized modes suitable for HMC.



(a) Volume (b) Volume+Double Occupancy (c) Volume+Surface (d) Volume+Double Occupancy+Surface

Figure 6.11: *Shape of the likelihood function for the hip angles θ_x and θ_y when employing different scores obtained from raw and surface voxels.*

Global Likelihood

The most common technique in PF based HBM tracking is to construct a global likelihood function, following the approach introduced by Deutscher *et al.* [DR05] and adopted by many other authors [RRR08, FLD08][3]. Body parts are not differentiated among them and the previously derived scores are referred to the whole body. In our case, this fact might lead to combine all previous scores from all body parts into a unique figure:

$$\rho^{\text{Out}}_{\text{Global}} = \frac{\sum_{\boldsymbol{\mathcal{Y}}\in\{\boldsymbol{\mathcal{V}}_{\mathcal{T}},\boldsymbol{\mathcal{V}}_{\mathcal{P}_{i,j}}\}} \rho^{\text{Out}}_{\boldsymbol{\mathcal{Y}}}|\boldsymbol{\mathcal{Y}}|}{\sum_{\boldsymbol{\mathcal{Y}}\in\{\boldsymbol{\mathcal{V}}_{\mathcal{T}},\boldsymbol{\mathcal{V}}_{\mathcal{P}_{i,j}}\}} |\boldsymbol{\mathcal{Y}}|}, \tag{6.17}$$

$$\rho^{\text{DO}}_{\text{Global}} = \frac{\sum_{\boldsymbol{\mathcal{Y}}\in\{\boldsymbol{\mathcal{V}}_{\mathcal{T}},\boldsymbol{\mathcal{V}}_{\mathcal{P}_{i,j}}\}} \rho^{\text{DO}}_{\boldsymbol{\mathcal{Y}}}|\boldsymbol{\mathcal{Y}}|}{\sum_{\boldsymbol{\mathcal{Y}}\in\{\boldsymbol{\mathcal{V}}_{\mathcal{T}},\boldsymbol{\mathcal{V}}_{\mathcal{P}_{i,j}}\}} |\boldsymbol{\mathcal{Y}}|}, \tag{6.18}$$

$$\rho^{\text{Occ}}_{\text{Global}} = \frac{\sum_{\boldsymbol{\mathcal{Y}}\in\{\boldsymbol{\mathcal{V}}_{\mathcal{T}},\boldsymbol{\mathcal{V}}_{\mathcal{P}_{i,j}}\}} \rho^{\text{Occ}}_{\boldsymbol{\mathcal{Y}}}|\boldsymbol{\mathcal{Y}}|}{\sum_{\boldsymbol{\mathcal{Y}}\in\{\boldsymbol{\mathcal{V}}_{\mathcal{T}},\boldsymbol{\mathcal{V}}_{\mathcal{P}_{i,j}}\}} |\boldsymbol{\mathcal{Y}}|}, \tag{6.19}$$

$$\rho^{\text{Surf}}_{\text{Global}} = \frac{\sum_{\boldsymbol{\mathcal{Y}}\in\{\boldsymbol{\mathcal{V}}^{\text{S}}_{\mathcal{T}},\boldsymbol{\mathcal{V}}^{\text{S}}_{\mathcal{P}_{i,j}}\}} \rho^{\text{Surf}}_{\boldsymbol{\mathcal{Y}}}|\boldsymbol{\mathcal{Y}}|}{\sum_{\boldsymbol{\mathcal{Y}}\in\{\boldsymbol{\mathcal{V}}^{\text{S}}_{\mathcal{T}},\boldsymbol{\mathcal{V}}^{\text{S}}_{\mathcal{P}_{i,j}}\}} |\boldsymbol{\mathcal{Y}}|}. \tag{6.20}$$

The weighting function can be constructed by assuming that these scores present some independence among them and follow a multivariate normal distribution[4]:

$$w(\mathbf{z}_t, \mathbf{y}) \propto p(\mathbf{z}_t|\mathbf{y}) = p(\{\boldsymbol{\mathcal{V}}_t, \boldsymbol{\mathcal{V}}^{\text{S}}_t\}|\boldsymbol{\mathcal{V}}^{\text{HBM}}_{\mathbf{y}}) \propto \mathcal{N}(\mathbf{d}, \boldsymbol{\mu}, \Sigma) \tag{6.21}$$

where $\boldsymbol{\mu} = \mathbf{0}$ and

$$\mathbf{d} = \left[\rho^{\text{Out}}_{\text{Global}},\ \rho^{\text{DO}}_{\text{Global}},\ \rho^{\text{Empty}}_{\text{Global}},\ \rho^{\text{Surf}}_{\text{Global}}\right], \qquad \Sigma = \text{diag}\left(\sigma^2_{\text{Out}}, \sigma^2_{\text{DO}}, \sigma^2_{\text{Empty}}, \sigma^2_{\text{Surf}}\right), \tag{6.22}$$

[3] See §3.1.2.3 for more information on likelihood construction procedures.

[4] Note: $\mathcal{N}(\mathbf{d}, \boldsymbol{\mu}, \Sigma) \propto \exp\left(-\frac{1}{2}(\mathbf{d}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{d}-\boldsymbol{\mu})\right)$

with $\rho^{\text{Empty}}_{\text{Global}} = 1 - \rho^{\text{Occ}}_{\text{Global}}$. Independence among variables leads to the diagonal configuration of Σ. Coefficients of the covariance matrix Σ will drive the influence of each score to the computed weight. Indeed, values of σ^2_{Out} and σ^2_{DO} will be low to penalize those particles whose associated volume $\boldsymbol{\mathcal{V}}^{\text{HBM}}$ falls out of the scene and those leading to impossible poses. These low values will act as a sharp cut-off, thus defining regions in the state space $\mathcal{X}$ where particles will be dismissed. On the other hand, factors σ^2_{Empty} and σ^2_{Surf} will assess the influence of occupancy and surface into the weight. Too small values of these variances will yield to peaky likelihood functions. Therefore, the proposed annealed PF (in fact, any PF scheme) will require a large number of particles to properly sample this state space. Varying the ratio among variance values, we may modify the shape of the likelihood function and, in our experiments, we set $\Sigma = \text{diag}(0.01, 0.01, 0.1, 0.1)$ leading to satisfactory results.

Factorized Likelihood

Partitioned or factorized likelihood assumes that all scores associated to every body part are independent from each other. This approach has been employed in some HMC algorithms such as in [HW07, GMD08], to cite some. It can be stated that:

$$w(\mathbf{z}_t, \mathbf{y}) \propto p(\mathbf{z}_t|\mathbf{y}) = p\left(\{\boldsymbol{\mathcal{V}}_t, \boldsymbol{\mathcal{V}}^{\text{S}}_t\}|\boldsymbol{\mathcal{V}}^{\text{HBM}}_{\mathbf{y}}\right) = \prod_{\boldsymbol{\mathcal{Y}} \in \{\boldsymbol{\mathcal{V}}_T, \boldsymbol{\mathcal{V}}_{\mathcal{P}_{i,j}}\}} p\left(\{\boldsymbol{\mathcal{V}}_t, \boldsymbol{\mathcal{V}}^{\text{S}}_t\}|\boldsymbol{\mathcal{Y}}\right). \tag{6.23}$$

Individual body parts likelihood function is defined through a multivariate normal distribution as:

$$p\left(\{\boldsymbol{\mathcal{V}}_t, \boldsymbol{\mathcal{V}}^{\text{S}}_t\}|\boldsymbol{\mathcal{Y}}\right) \propto \mathcal{N}\left(\mathbf{d}, \boldsymbol{\mu}, \Sigma_{\boldsymbol{\mathcal{Y}}}\right), \tag{6.24}$$

where, analogously,

$$\mathbf{d} = \left[\rho^{\text{Out}}_{\boldsymbol{\mathcal{Y}}}, \rho^{\text{DO}}_{\boldsymbol{\mathcal{Y}}}, \rho^{\text{Empty}}_{\boldsymbol{\mathcal{Y}}}, \rho^{\text{Surf}}_{\boldsymbol{\mathcal{Y}}}\right], \quad \rho^{\text{Empty}}_{\boldsymbol{\mathcal{Y}}} = 1 - \rho^{\text{Occ}}_{\boldsymbol{\mathcal{Y}}}, \quad \boldsymbol{\mu} = \mathbf{0}. \tag{6.25}$$

In this case, we can define different covariance matrices for each body part, $\Sigma_{\boldsymbol{\mathcal{Y}}}$, in order to better design the resulting likelihood function. In our case, we selected the values shown in Table 6.2. Typically, end parts of a limb (arms and legs) move faster than their preceding part in the kinematic chain (forearms and forelegs) therefore a higher variance in the propagation of the angles of the associated joints (elbows and knees) is applied. In order to better concentrate particles in these body parts, the associated variances of the occupancy and surface scores are decreased. Torso, being the root of the HBM must be accurately fitted hence the restrictions imposed on the occupancy and surface variances. Moreover, torso has more chances than any other body part to intersect with other limbs, hence the increase of the double occupancy variance.

6.6 Marker Based APF HBM Tracking Results

In order to test the proposed algorithm two tracking scenarios have been chosen. The first scenario is the already mentioned HumanEva-I dataset. As a complementary test for the proposed tracking system, a real case scenario is analyzed where the subject under

Body part	Σ
Forearms ($\mathcal{P}_{1,0}$,$\mathcal{P}_{2,0}$)	$\Sigma = \text{diag}\left(\sigma^2_{\text{Out}}, \sigma^2_{\text{DO}}, \sigma^2_{\text{Empty}}, \sigma^2_{\text{Surf}}\right)$
Forelegs ($\mathcal{P}_{3,0}$,$\mathcal{P}_{4,0}$)	$\Sigma = \text{diag}\left(\sigma^2_{\text{Out}}, \sigma^2_{\text{DO}}, \sigma^2_{\text{Empty}}, \sigma^2_{\text{Surf}}\right)$
Arms ($\mathcal{P}_{1,1}$,$\mathcal{P}_{2,1}$)	$\Sigma = \text{diag}\left(\sigma^2_{\text{Out}}, \sigma^2_{\text{DO}}, \frac{1}{2}\sigma^2_{\text{Empty}}, \frac{1}{2}\sigma^2_{\text{Surf}}\right)$
Forelegs ($\mathcal{P}_{3,1}$,$\mathcal{P}_{4,1}$)	$\Sigma = \text{diag}\left(\sigma^2_{\text{Out}}, \sigma^2_{\text{DO}}, \frac{1}{2}\sigma^2_{\text{Empty}}, \frac{1}{2}\sigma^2_{\text{Surf}}\right)$
Head ($\mathcal{P}_{0,0}$)	$\Sigma = \text{diag}\left(\sigma^2_{\text{Out}}, \sigma^2_{\text{DO}}, \sigma^2_{\text{Empty}}, \sigma^2_{\text{Surf}}\right)$
Torso ($\mathcal{T}$)	$\Sigma = \text{diag}\left(\sigma^2_{\text{Out}}, 2\sigma^2_{\text{DO}}, \frac{1}{2}\sigma^2_{\text{Empty}}, \frac{1}{2}\sigma^2_{\text{Surf}}\right)$
$\sigma^2_{\text{Out}} = 0.01$, $\sigma^2_{\text{DO}} = 0.01$, $\sigma^2_{\text{Empty}} = 0.1$, $\sigma^2_{\text{Surf}} = 0.1$	

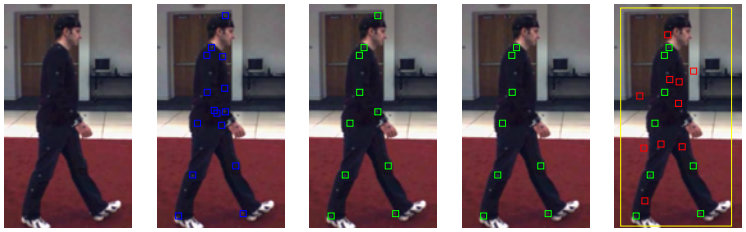
Table 6.2: *Partitioned likelihood covariance matrices setup for every body part.*

study is performing a series of dancing figures. This scenario was particularly challenging for any tracking system due to the rapid movements involved in the motion. No ground truth information was available hence no quantitative results are presented for this task. However, visual inspection corroborates the good performance of our system.

6.6.1 HumanEva-I Results

HumanEva database [SB06] has been chosen to test our algorithm since it provides synchronized and calibrated data from both several cameras and a professional motion capture (MoCap) system. Furthermore, this database has been chosen since it has been largely employed thus allowing comparison with other already presented results [BSB05]. The MoCap system used an array of 3 adapted cameras to capture reflective markers. An estimated triangularization of 3D landmark positions was produced at a frame rate of 60 Hz. This output is used as the ground truth data when evaluating marker-less body motion trackers due to the high precision of the 3D estimated positions. The presented system would replace the triangularization process, allowing the usage of standard off the shelf cameras instead of dedicated and expensive hardware. Exploiting the underlying information bounded to the structure of the HBM may help in producing more accurate results and a robust triangularization process.

As it has been presented in §6.4.1.1, the input measurements $\mathbf{z}_t$ of the proposed PF are a set of 2D detections, $\mathcal{D}_n$, measured over N_{C} cameras for every time t. Unfortunately, images used by the MoCap system were not distributed and the markers were not visible in the released RGB images. For the sake of comparison with other algorithms previously evaluated within the HumanEva framework, a synthetic data generation strategy has been devised where the 2D projection of the markers onto all camera views will be derived from the 3D ground truth data. The image formation process and the marker detection algorithm have been rigorously simulated to generate the observation set $\mathcal{D}_n$ out of the MoCap output 3D landmark positions noted as X. First, frame rate of X is adapted to the camera frame rate, that is 30 frames per second. Then, for every time



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

Figure 6.12: *Synthetic data generation process. Since the reflective markers are not distinguishable in the original RGB image (a), the sets* $\{\mathcal{D}_n\}_{n=1}^{N_C}$ *are generated from the 3D locations provided by the MoCap system. First, for a given view* n*, all 3D markers are projected onto the corresponding image, (b), and those affected by body auto-occlusions are removed, (c). Then, the marker detection algorithm* Γ *is applied: some markers are missed due to the detection ratio, (d), and a number of false measurements are generated, (e). Finally, an amount of Gaussian noise with variance* σ_Γ^2 *is added simulating the position estimation error.*

instant, the synthetic data generation was driven by the following steps:

Image formation

1. Inverse kinematics are applied to X_t to estimate the pose of a HBM $\mathcal{H}$ and body parts are fleshed out with super-ellipsoids.

2. Every 3D location in X_t is projected onto every camera in order to generate the sets $\mathcal{D}_n$, $0 \leq n < N_C$. The previously estimated HBM pose proves for the visibility of markers onto a specific camera view by modelling the possible auto-occlusions among body parts. At this point, the 2D locations contained in $\mathcal{D}_n$ would be the positions obtained by an ideal marker detection algorithm.

Marker detection algorithm

3. The detection rate ($\overline{DR}$) of the marker detection algorithm Γ will determine whether a marker has been detected by drawing a sample from a uniform random variable in the range [0,1] and comparing its value with $\overline{DR}$.

4. The number of wrong measurements will be determined by the false positive rate ($\overline{FP}$). Again, a sample drawn from a uniform random variable in the range [0,2$\overline{FP}$] will determine the number of false measurements found in every image plane. The positions of these false measurements are set to be samples from a bivariate uniform random variable in the rectangular area tightly surrounding the subject.

5. Finally, a Gaussian noise with variance $\sigma_\mathcal{D}^2$ and mean 0 is added to all the elements in $\mathcal{D}_n$ simulating the estimation error committed by the marker detection algorithm.

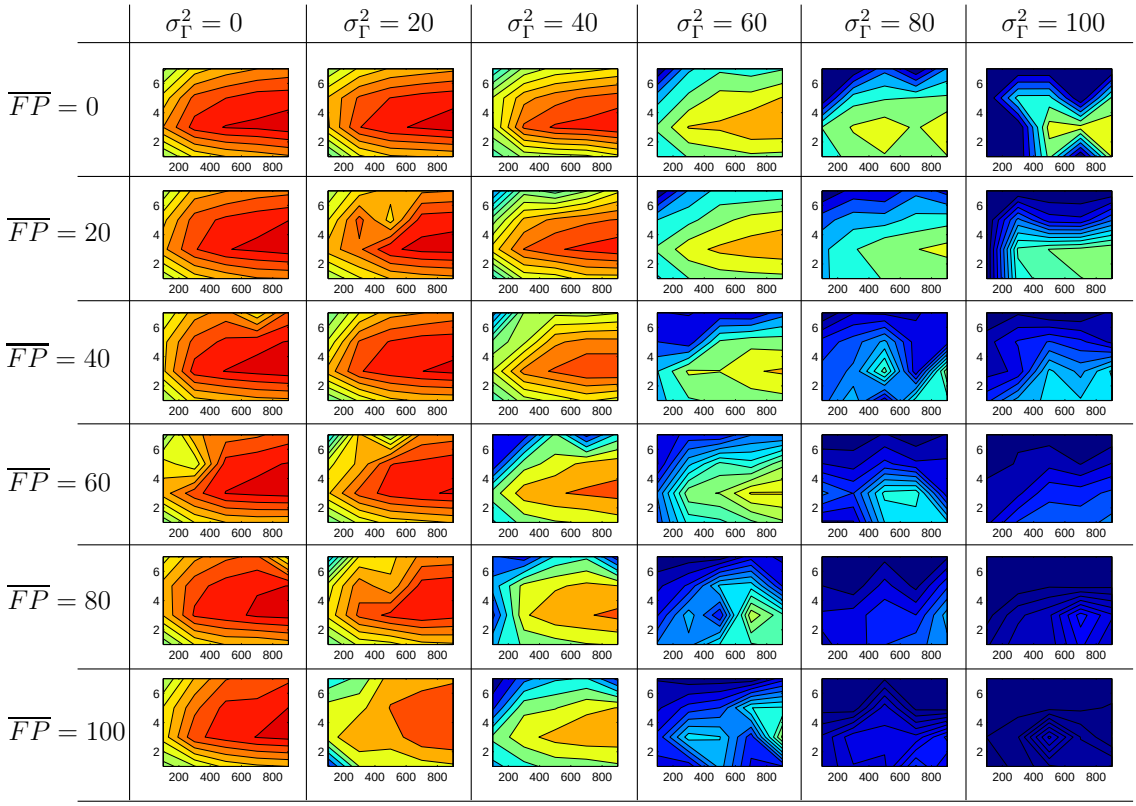


Figure 6.13: *Quantitative results over the HumanEva-I dataset. Score* MMTA *is displayed in pseudo-color for the marker detection algorithms characterized by* $\overline{DR} = 0.9$,$\overline{FP} = \{0, 20, 40, 60, 80, 100\}$ *and* $\sigma_\Gamma^2 = \{0, 20, 40, 60, 80, 100\}$ *mm. In the subplots, the* y *axis accounts for the number of layers* L *and the* x *axis for the number of particles per layer* $N_{p,L}$.

An example of this synthetic data generation process is shown in Figure 6.12.

In order to test the performance of the proposed tracking system, two factors must be taken into account: the performance of the marker detection algorithm Γ (determined by the triplet $\{\overline{DR}, \overline{FP}, \sigma_\Gamma^2\}$) and the design parameters of the PF, that is the number of layers L and the number of particles per layer $N_{\mathrm{p,L}}$. The following parameter values were used to simulate different performances of Γ: $\overline{DR} = \{1, 0.9, 0.8, 0.7, 0.6\}$, $\overline{FP} = \{0, 20, 40, 60, 80, 100\}$ and $\sigma_\Gamma^2 = \{0, 20, 40, 60, 80, 100\}$ mm. All the combinations among these parameters were tested producing 360 possible Γ marker detection algorithms. To address the performance of the tracking system, the following parameters where employed $L = \{1, 3, 5, 7\}$ and $N_{\mathrm{p,L}} = \{100, 300, 500, 700, 900\}$. A large simulation was then conducted over the HumanEva-I database testing all the resulting 7200 combinations between Γ and the proposed PF algorithm. With this test, we thoroughly explored the performance of the PF, even in very adverse conditions. An example of this simulation

	Marker based APF					
	μ	σ	*MMTP*	*MMTA*	μ_θ	σ_θ
Walking	56.01	14.46	45.81	96.15	6.02	2.55
Jog	62.51	18.71	47.77	90.12	7.85	2.75
Throw/Catch	58.31	18.64	47.13	91.72	9.22	6.17
Gesture	44.70	4.31	42.42	97.46	5.89	2.83
Box	77.89	30.64	46.12	87.03	10.55	7.04
Average	**59.88**	**17.35**	**45.85**	**95.32**	**7.09**	**4.21**

Table 6.3: *Quantitative results for the HumanEva-I dataset when using a marker detection algorithm with $\overline{DR} = 0.9$, $\overline{FP} = 20$ and $\sigma_\Gamma^2 = 4$. PF parameters were set to $L = 3$ and $N_p = 700$. Distances are measured in millimeters, angles in degrees and $\delta = 100$ mm.*

is depicted in Figure 6.13 where the *MMTA* score is displayed as the more informative metric to quantize the PF performance.

Analyzing the results shown in Figure 6.13, it may be seen that the algorithm is robust against the number of false detections $\overline{FP}$, since it is very unlikely that false 2D measurements in different views keep a 3D coherence. In this case, the spacial redundancy is efficiently exploited to discard these measurements. On the other hand, the performance of the algorithm decreases as the position estimation error increases. Another evident fact to be emphasized is the over-annealing effect introduced in §3.2.3. The performance of the algorithm as the number of annealing layers employed increases is not monotonically increasing. Indeed, for a certain number of layers, the performance starts decreasing. This happens when the particles concentrate too much around the peaks of the weighting function, hence impoverishing the overall representation of the likelihood distribution. For this motion tracking problem, we found that optimal PF configuration was $L = 3$ and $N_{\text{p,L}} = 700$.

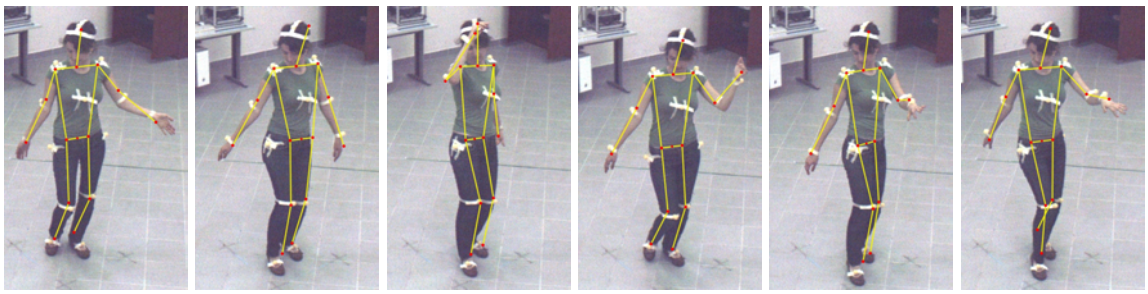
Finally, a reasonable configuration of the marker detection algorithm Γ was set to $\overline{DR} = 0.9$, $\overline{FP} = 20$ and $\sigma_\Gamma^2 = 4$. Detailed results obtained for this case with $L = 3$ and $N_{\text{p,L}} = 700$ are shown in Table 6.3.

6.6.2 Real case

The presented body tracking algorithm has been applied to capture motion figures from 4 different types of dances: *salsa*, *belly dancing* and two Turkish folk dances. The output of this algorithm was a building block of a more general system aiming at an audio-visual analysis of dances [OCFT$^+$08b, ODCF$^+$08]. The analysis sequences were recorded with 6 fully calibrated cameras in the SmartRoom at Koc University with a resolution of 1132x980 pixels at 30 fps.

Markers attached to the body of the dance performer were little yellow balls and a color-based detection algorithm Γ has been used to generate the sets $\mathcal{D}_n$ for every incoming multi-view frame. The original images are processed in the YCrCb color space which gives flexibility over intensity variations in the frames of a video as well as among

(a) Salsa figures



(b) Belly dancing figures

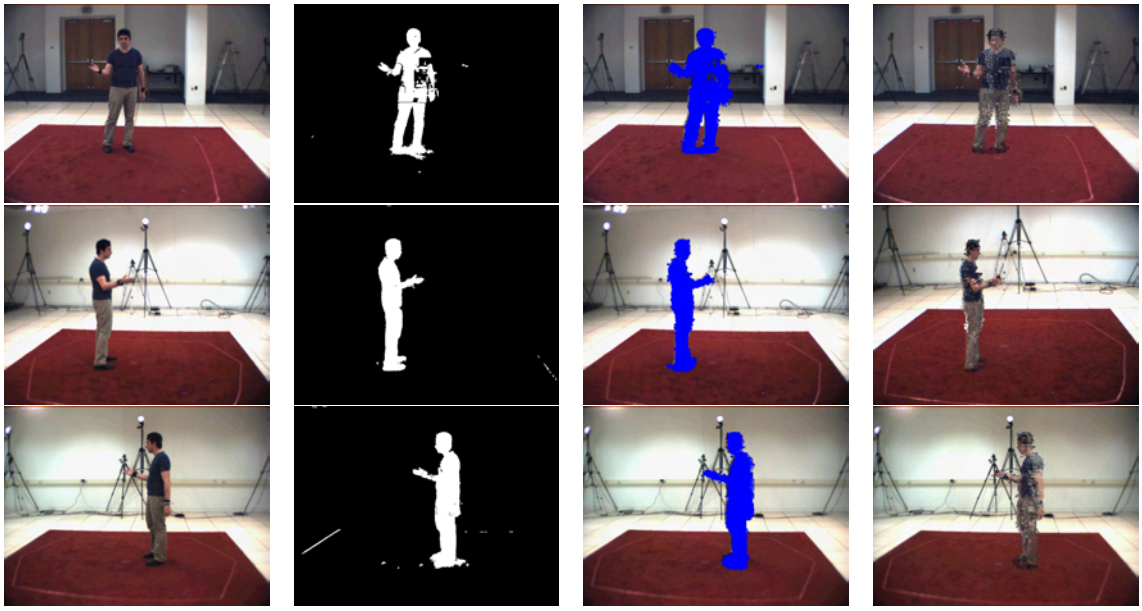
Figure 6.14: *Dance motion tracking results. Two examples of dance tracking: salsa and belly dancing.*

the videos captured by the cameras from different views. In order to learn the chrominance information of the marker color, markers on the dancer are manually labeled in one frame for all camera views. It was assumed that the distributions of Cr and Cb channel intensity values belonging to marker regions are Gaussian. Thus, the mean can be computed over each marker region (a pixel neighborhood around the labeled point). Then, a threshold in the Mahalanobis sense is applied to all images in order to detect marker locations. An empirical analysis showed that the detector Γ had the following performance triplet: $\overline{DR} = 0.98$, $\overline{FP} = 4$ and $\sigma^2_{\mathcal{D}} = 2$.

In this particular scenario, the algorithm had to cope with very fast motion associated to some figures. Even though these harsh conditions, the results were satisfactory and visually accurate. Check `http://www.cristiancanton.org/projects` for some example videos.

6.6.3 Computational cost

Marker based HMC is a low demand system in the sense that the number of involved operations is relatively small in comparison with the markerless approach. In the presented system, the main bottleneck is the pixel based marker detection algorithm Γ that has to be applied to all input images from multiple streams. Apart from this, the core APF filter was able to perform at twice the real-time speed when processing off-line extracted markers. It must be said that we aimed at studying the computational cost of the algorithm itself since markers extraction is not our topic of research. Moreover, this



(a) Raw images (b) Segmented images (c) 3D reconstruction (d) 3D colored reconstruction

Figure 6.15: *Sample images from the HumanEva-I dataset.*

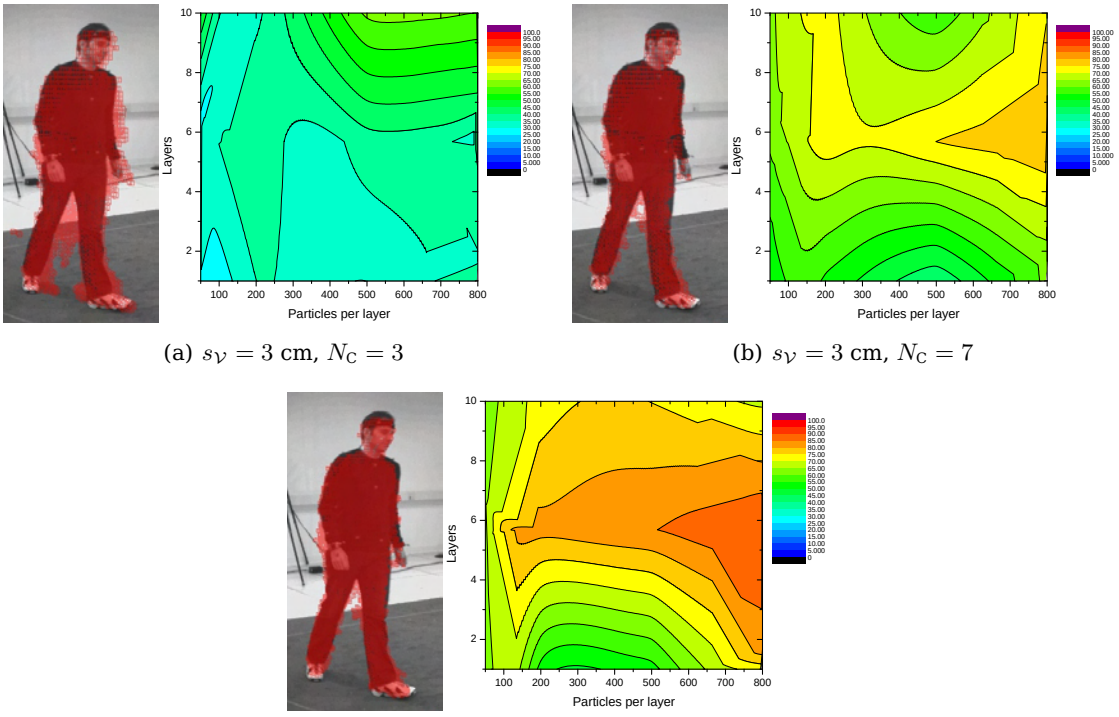
step can be addressed employing hardware implementations as done by [Vic] yielding to real-time performance of the marker extraction operation.

6.7 Markerless Based APF HBM Tracking Results

As previously done with the marker based proposal, we use HumanEva-I dataset to assess the performance of the proposed markerless HMC algorithm. In this way, we will be able to compare our results with other markerless HMC methods that also employed such dataset. Reconstruction of the 3D space represented by means of voxels will be the input data to all presented markerless algorithms (see an example in Figure 6.15).

6.7.1 Parameter setting

The number of involved parameters and factors into the markerless APF HBM tracking algorithms is high. Hence, presenting results over a large database exploring all the dependencies is an unachievable goal. Instead, a portion of the HumanEva-I database has been employed to study such dependencies towards determining the optimal parameter set. This portion includes a sample of the four executed actions, each of them by a different performer. In the presented experiments, the *MMTA* score is chosen as it is the most significant.



(a) $s_{\mathcal{V}} = 3$ cm, $N_{\mathrm{C}} = 3$

(b) $s_{\mathcal{V}} = 3$ cm, $N_{\mathrm{C}} = 7$

(c) $s_{\mathcal{V}} = 2$ cm, $N_{\mathrm{C}} = 7$

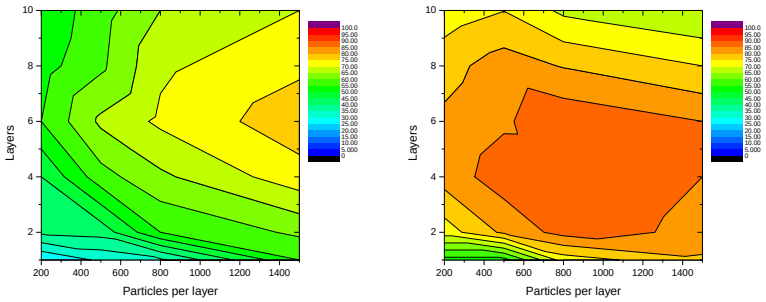
Figure 6.16: *Data resolution influence on the markerless APF HBM tracking algorithm. For every pair of voxel size $s_{\mathcal{V}}$ and number of cameras N_C, we display the voxel reconstruction and the* MMTA *scores for different number of annealing layers L and particles per layer $N_{p,L}$.*

6.7.1.1 Data resolution

Data resolution fed to the APF algorithm drives the quality of the obtained results. Algorithms relying directly on image measurements have the pixel as the minimum resolution[5] but, in our case, input data is generated previously and the voxel size $s_{\mathcal{V}}$, being that minimum data resolution, decided beforehand. Moreover, the employed reconstruction algorithm may affect the quality of the obtained volume. Two experiments have been conducted to assess the influence of the input data to the presented markerless APF HBM tracking algorithm: the number of cameras N_{C} and the voxel size $s_{\mathcal{V}}$.

For the first parameter, we tested the performance when generating a 3D reconstruction only using data provided by the color cameras (3 cameras) since the extraction of foreground regions was better achieved; this led to the results displayed in Figure 6.16(a). This result was compared to the performance of the algorithm when operating on 3D reconstructions obtained from both color and grayscale cameras (7 cameras), see Figure 6.16(b). Despite the foreground segmentation obtained for grayscale images was not as accurate as the one obtained with color images, the spatial redundancy among

[5] In image processing, the concept of sub-pixel precision is employed, specially in the super-resolution field.



(a) Global likelihood (b) Factorized likelihood

Figure 6.17: MMTA *Comparison between global and partitioned likelihood for the APF algorithm using* $N_C = 7$ *and* $s_{\mathcal{V}} = 2$ *cm.*

cameras was better exploited. As a result, the obtained volumes were better defined leading to a noticeable improvement of the overall performance.

When analyzing the influence of the voxel size $s_{\mathcal{V}}$ on the performance of the markerless APF HBM tracking algorithm (with a fixed number of cameras N_{C}), we must compare Figures 6.16(b) and 6.16(c). The smaller the voxel size, the better the capture of the details of the scene hence the improvement of the *MMTA* score (and, obviously of the *MMTP* score due to the improvement of the minimum resolution). However, as it will be further discussed, employing a small $s_{\mathcal{V}}$ leads to an increase of the computational complexity of the overall system.

For both situations where $N_{\mathrm{C}} = 7$, we can observe that the optimal operation point achieving maximum *MMTA* with the minimum number of overall particles is $L = 6$ and $N_{\mathrm{p,L}} = 600$, corresponding to $N_{\mathrm{p}} = 3600$ effective particles.

6.7.1.2 Global vs Partitioned Likelihood

As it has been introduced in §6.5.1.2, the likelihood between a particle $\mathbf{y}_t^j$ and the input data $\mathbf{z}_t$ can be evaluated as a global or factorized likelihood. In order to determine the best choice we tested the APF algorithm under the same conditions for both likelihood proposals as shown in Figure 6.17.

Defining a cost function for every body part and then combining them through a likelihood function based on a multi-variate normal distribution allows a better capture of the match between the pose encoded in a particle with relation with the input data. Particularly, this option generates a more multimodal likelihood function since a per-limb analysis enforces more regions in the state space to be avoided (that is, with a low likelihood). However, APF can cope with such shapes and still perform adequately.

On the other hand, the global likelihood evaluation requires far more particles to properly explore the state space and to obtain comparable performance scores with the fractionate likelihood. In this case, the cost function still tends to be multimodal but with a lower difference between the peaks of the modes in it, thus requiring more particles or annealing layers to search for the main mode. Accumulating all the raw and surface

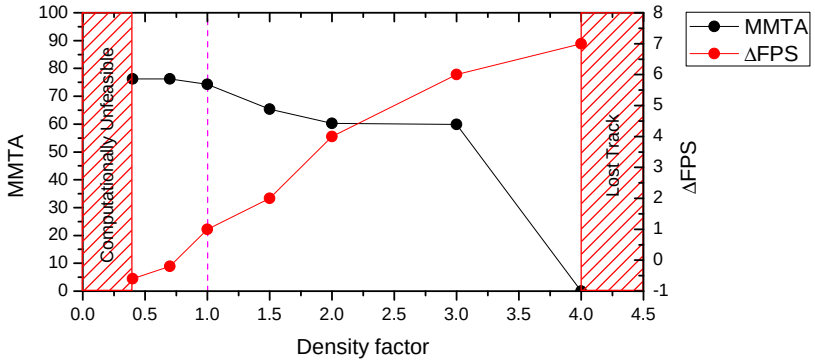


Figure 6.18: *Effect of LUT sub-sampling on the* MMTA *score and the computational complexity (expressed as* FPS*). Density factor stands for coefficient* $\delta_{Density}$.

voxels related scores does not properly model local errors. For instance, the global cost function can not distinguish between a pose with a completely wrongly fitted body part (that is with a zero occupancy score) and a pose with all body part slightly wrongly fitted. We can also compare the global and partitioned likelihood construction with the sum-product duality found in many optimization processes [BV04]. In this case, when finding a maximum of a function that is formed by a product or sum of other functions, the product one tends to present sharp peaks in the location where all subfunctions have a maximum while, in the sum one, the function tends to present several smooth peaks. Indeed, there is a drawback between efficiency and accuracy since particles evaluated over product-based likelihoods will tend to exhibit low values except for those close to a maximum while the sum-based ones will present more uniform values. Empirically, it has been seen that for a good initial distribution the product-based likelihood rapidly converges towards the main peak of the function while the sum-based one requires far more annealing layers to properly locate the main peaks of the likelihood as seen in Figure 6.17.

Factorized likelihood has proved to be a more effective choice than the global likelihood to properly model the relation between the pose encoded in $\mathbf{y}_t^j$ and the input data $\mathbf{z}_t$. It must be noted that the computational cost difference between these two methods is negligible.

6.7.1.3 Model density

As presented in §6.5.1.1, the HBM is fleshed out with truncated cones and the position of the discretization of these cones is stored in a look-up table, described in Eq.6.11. In the initialization phase, the body is set into the neutral position $\boldsymbol{\mathcal{V}}_0^{\mathrm{HBM}}$, the LUT is created and, when requiring the position of the discretized cones into a given pose $\mathbf{y}_t^j$, forward kinematics equations (Eq.6.1) are applied to obtain $\boldsymbol{\mathcal{V}}_{\mathbf{y}}^{\mathrm{HBM}}$. Finally, the occupancy and surface scores are computed based on the input data $\mathbf{z}_t$ and $\boldsymbol{\mathcal{V}}_{\mathbf{y}}^{\mathrm{HBM}}$. However, this process can be speeded up by employing only a fraction of the LUT, that is to use a

sub-sampled version of every discretized limb. This will require less operations when computing $\boldsymbol{\mathcal{V}}_{\mathbf{y}}^{\mathrm{HBM}}$, at the cost of an inaccuracy when computing the scores employed in the likelihood. Figure 6.18 depicts the influence of the LUT sub-sampling in the *MMTA* score compared with the relative increment of processed FPS. Let us denote the density factor as $\delta_{\mathrm{Density}}$ and define it as:

$$\delta_{\mathrm{Density}} = \frac{s_{\mathcal{V}_{HBM}}}{s_{\mathcal{V}}}. \tag{6.26}$$

6.7.2 HumanEva-I Results

The proposed markerless APF system is evaluated using the HumanEva-I framework. As discussed in §6.7.1, we fixed the optimal algorithm operation parameters as follows:

- Employ small voxel sizes, $s_{\mathcal{V}} = 2$ cm.
- Use the highest number of available cameras, $N_{\mathrm{C}} = 7$.
- Partitioned likelihood evaluation is employed since it better captures the similarity between the input data $\mathbf{z}_t$ and the pose encoded in a given particle, $\mathbf{y}_t^j$.
- Data density has been set to $\delta_{\mathrm{Density}} = 1$.
- The optimal number of particles per layer $N_{\mathrm{p,L}}$ and annealing layers L to be used by the APF algorithm was set to $N_{\mathrm{p,L}} = 600$ and $L = 6$ ($N_{\mathrm{p}} = 3600$) according to the results displayed in Figure 6.16(c).

Results are presented in Table 6.4. A plot of the temporal evolution of the position of the virtual markers employed in the calculation of the performance metrics is depicted in Figure 6.19 while in Figure 6.20 there are visual examples of the tracked actions using the proposed APF markerless method.

6.7.3 Computational cost

Computational complexity of the markerless approach to HMC is tightly coupled with the voxel density of the HBM and the size of the voxel side, $s_{\mathcal{V}}$, as explained in §6.7.1.3 and §6.7.1.1. Processing data with a resolution of $s_{\mathcal{V}} = 2$ cm made the system fall below the real-time limit, achieving $FPS = 0.15$ while $s_{\mathcal{V}} = 3$ cm yield to $FPS = 0.29$. However, it must be said that these systems were not optimized for real-time performance.

6.8 Conclusions

This chapter started with a general introduction to Bayesian human motion capture and tracking based on Monte Carlo formulation using multi-camera input data. Within this context, a realistic HBM together with an annealing PF approach is selected to address marker based and markerless HMC using two novel approaches.

First, a real-time marker based HMC is presented, employing the detections onto several image views of a set of distinguishable markers placed on body landmarks of the

	Markerless based APF					
	μ	σ	*MMTP*	*MMTA*	μ_θ	σ_θ
Walking	96.52	41.64	72.05	79.55	7.97	2.51
Jog	130.34	62.01	92.21	68.24	9.91	3.07
Throw/Catch	145.22	52.13	94.69	61.30	11.53	3.47
Gesture	124.87	45.66	90.43	69.17	10.96	4.25
Box	122.27	42.68	92.77	68.38	9.54	4.10
Average	**121.18**	**45.92**	**90.17**	**71.36**	**10.12**	**3.33**

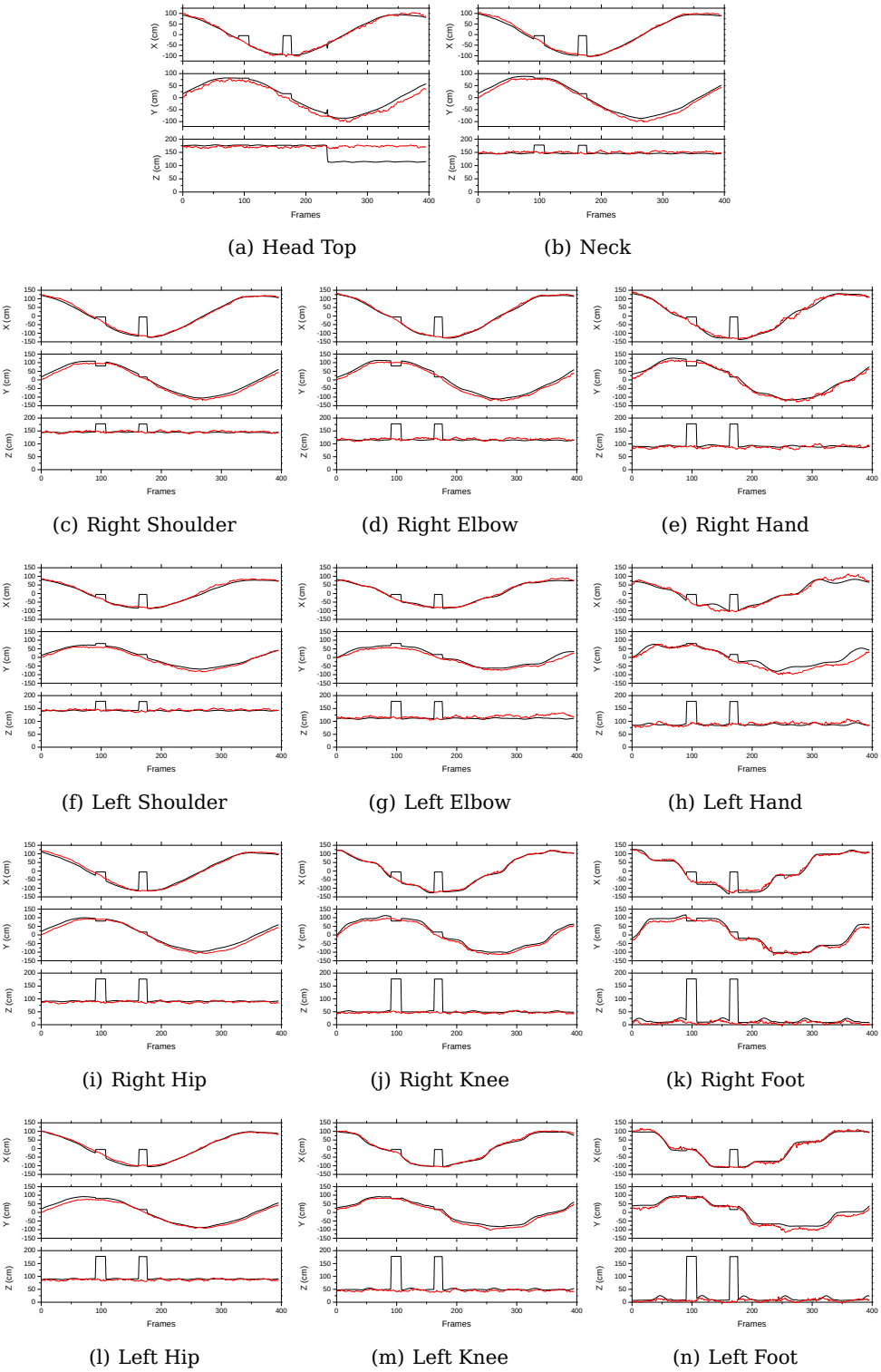
Table 6.4: *Markerless APF tracking results on HumanEva-I dataset with* $\epsilon = 100$ *mm.*

performer. HBM fitting is performed within an annealing PF scheme where multi-camera geometry is exploited by means of the symmetric epipolar distance in the likelihood evaluation. This system is proposed as an economic alternative to commercial HMC systems that require expensive and dedicated hardware.

Secondly, we presented a system for markerless HMC fed by a voxel reconstruction of the scene together with an annealed PF. In this system, every particle encoding a pose of the HBM is rendered into 3D, allowing a match assessment between this pose and the input data encoded by three scores: output, occupancy and double occupancy. These measures, together with a surface distance, conform the employed likelihood function.

Finally, performance of all presented methods is assessed using the previously introduced HumanEva-I database and metrics. Comparing results for marker based and markerless systems, we notice that marker based obtains a better performance in both *MMTP* and *MMTA*. This obvious effect is justified in the *MMTA* by the fact that input data to marker based is well defined and the likelihood evaluation efficiently exploits the multi-view geometry, while in the markerless approach data is easily corrupted and faulty. *MMTP* score is also influenced by this effect, but it should also be considered the fact that, in the marker based technique, we are working with the maximum resolution unit produced by the sensors, the pixel, whereas in the markerless approach, we are dealing with an artificial representation, the voxel.

When examining the 3D voxel reconstructions for the sequences yielding to less accurate results, we noticed that these data was faulty and large parts of the reconstructed person were missing. Indeed, this issue cannot be avoided efficiently with the presented markerless technique since the parts where no data is present cannot be inferred. This data corruption problem was the starting point for Chapter 7 and the motivation to develop robust techniques ables to adapt to any input data quality.



(a) Head Top (b) Neck

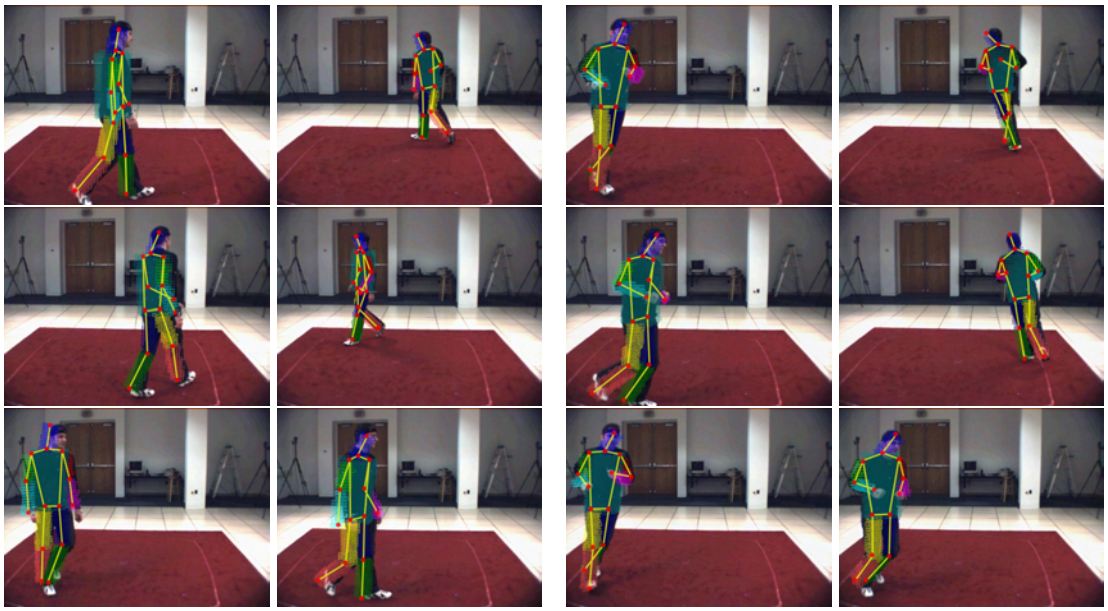
(c) Right Shoulder (d) Right Elbow (e) Right Hand

(f) Left Shoulder (g) Left Elbow (h) Left Hand

(i) Right Hip (j) Right Knee (k) Right Foot

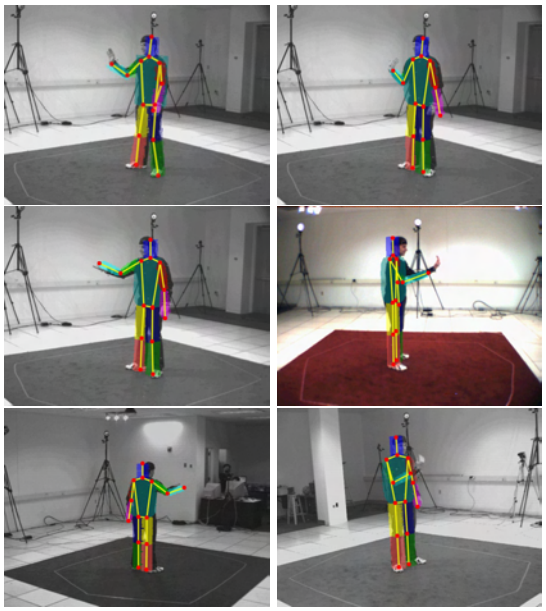
(l) Left Hip (m) Left Knee (n) Left Foot

Figure 6.19: *Position curves for the main joints in the HBM. Black lines stand for the ground truth data while red lines stand for the estimation obtained with the APF algorithm with partitioned likelihood evaluation. Note the glitches in the ground truth data introduced by an error in the annotation.*

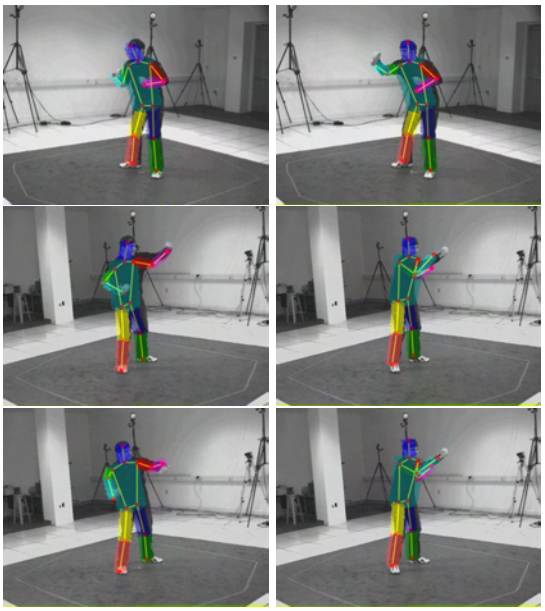


(a) Walking

(b) Jogging



(c) Gesturing



(d) Boxing

Figure 6.20: *Tracking examples of several actions contained in the HumanEva-I database.*

7

Robust Motion Capture with Scalable Human Body Models

A COMMON but unrealistic assumption in most of human motion capture (HMC) algorithms is that available data is free from disturbing elements such as noise, missing data or spurious blobs. Therefore, extracted features are reliable and properly describe the scene to be analyzed. Most of the available algorithms are designed to deal with accurate inputs, hence only facing problems inherent to the fitting process of a highly articulated structure (the human body) using these data. In some cases, some moderate corruption of the data can be handled for a short period of time. Most techniques presented in the literature have not been tested in more severe (and realistic) scenarios and correct performance can not be guaranteed since they were not designed for such operation conditions.

The problem of HMC using variable quality input data can be tackled by means of an adaptive human body model (HBM) that adequates its structure to the characteristics of the analyzed scenario towards ensuring the convergence of the fitting algorithm. We introduce the concept of scalable human body model (SHBM) as a collection of HBMs that fulfill a hierarchical relation among them. This hierarchy is referred to its structure and we devised two scalability criteria: the inclusive and unitive paradigms. This chapter presents several contributions based on the core idea of the SHBM to improve the standard HMC techniques that usually employ a HBM selected beforehand.

Two filtering strategies are introduced towards exploiting the hierarchical properties of the SHBM. First, the SHBM-Annealed Particle Filter (SHBM-APF) performs a progressive model fitting of the SHBM to the input data using a layered scheme inspired in the annealing idea but, instead of using a set of progressively smoothed versions of the likelihood function, we use a set of progressively refined HBMs (that is, those contained in the SHBM). We define this concept as *structural annealing*, and its usage yields an improvement of the robustness of the algorithm and its computational efficiency when compared with the already presented HMC APF algorithms. The second contribution is the Data Driven Adaptive Model-APF (DDAM-APF) where input data is processed towards selecting the most suitable HBM within the SHBM, thus being able to deal with heavily corrupted data. This algorithm is particularly effective when analyzing realistic scenarios where performers might be partially occluded. In this case, a beforehand selected HBM can not properly explain the observed data and eventually leads to loss of track.

The following contributions have been published: [CFCP08].

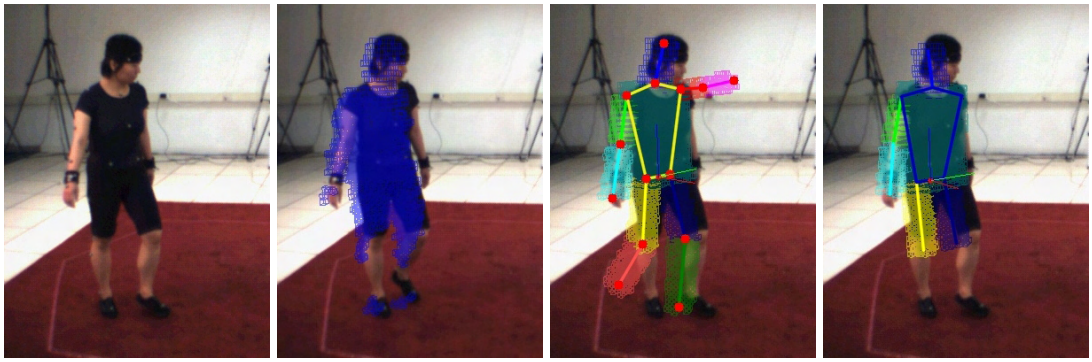
7.1 Problem formulation

Usually, the data employed in human motion capture (HMC) algorithms presented in the literature is clean and of good quality [Mik03, CGH05, RRR08]. In some cases, tests have been conducted to assess the robustness of tracking algorithms when dealing with data corrupted with a certain amount of noise [BSB05]. However, there is a number of situations that have not been covered: heavy and prolonged occlusions, large data misses, spurious blobs, etc. In such cases, available algorithms might tend to fail or diverge and eventually recover track when data is again of tractable quality. In general, HMC algorithms have been tested under benevolent conditions and extreme situations have been avoided or disregarded.

Data fed to HMC algorithms is usually captured in controlled environments [SB06], hence minimizing the risk of interfering factors such as illumination changes, shadows, etc. Moreover, it is assumed that performers wear a determined type of garments [Mik03] and no occluding elements such as tables or chairs are present. If all these conditions are fulfilled, the data obtained will only contain the standard challenges a HMC algorithm should deal with: auto-occlusions and perspective. In this way, the usage of a single beforehand selected HBM is justified and appropriate, since the data faithfully describes the structure of the body. However, in the case where one or some of these conditions are violated, the existing models may not be able to explain the obtained data, thus underperforming and loosing track hereafter. Recently, a contribution to attain some independence to the appearance of the performers when estimating body pose has been presented by Bălan *et al.* [BB08] using very detailed models of the human body together with an energy minimization method. Some other contributions [DBT03] may infer missing data for a period of time but, to the best of of the author's knowledge, none is intended to prolongedly tackle with faulty data.

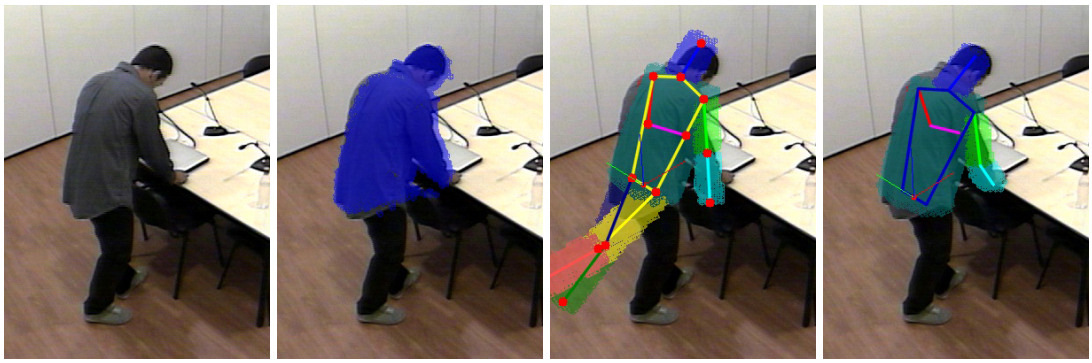
Employing a fixed beforehand selected HBM to perform HMC might not be appropriate to analyze faulty input data yielding to erroneous pose configurations. Two main cases can be considered where a beforehand selected HBM can lead to a faulty analysis. Let us consider Figures 7.1(a) and 7.1(b). In the first case, input data (voxels) are corrupted due to a wrong 3D reconstruction derived from a faulty background/foreground segmentation. When analyzing these data with the markerless APF algorithm presented in §6.5, we obtain wrong pose estimations for those limbs that have no associated data (i.e. the left arm and the legs). Some methods [UFF06] may cope with missing data relying on previously learnt motion periodicity patterns and the assumption that the performed action is known beforehand hence, in our particular case, predicting the missing knee and feet positions. Eventually, when dealing with a long miss of data, the motion prediction may produce wrong pose estimations. The second case depicts, Figure 7.1(b), a similar situation of missing data when occluding elements produce a voxel reconstruction of the subject under study with some missing body parts. In this case, previously learnt motion patterns can not be applied.

Conceptually, taking into account the particularities of the input data may allow se-



Original Image | Voxel Data | Markerless APF | Adaptive SHBM

(a) **Case 1:** Faulty input data



Original Image | Voxel Data | Markerless APF | Adaptive SHBM

(b) **Case 2:** Occluded body parts

Figure 7.1: *Example of the influence of an occlusion of a part of the body in the HMC process. In the first column, input images from HumanEva-I and CLEAR databases. In the second, the 3D voxel reconstruction is overlaid where, in case 1, legs and the left arm are missing due to a wrong data segmentation and, in case 2, legs are not reconstructed since the performer is partly occluded by the table. In the third column, the standard markerless APF algorithm is unable to retrieve the pose of the missing limbs. Finally, the fourth column depicts an adaptive HBM algorithm able to cope with data misses.*

lecting the most suitable HBM from a collection (SHBM) to ensure the convergence of the analysis algorithm. In the case where data is not complete, it might not be possible (or necessary) to estimate the pose of the occluded/missing limbs. When applying this idea, we can obtain results similar to those of the last column of Figure 7.1. In this chapter, we will present several techniques to decide what is the most suitable HBM to analyze a given scene, to decide which limbs are missing and how to gracefully shift from one model to another within a PF framework.

7.2 Scalable Human Body Model

7.2.1 Literature review

The concept of scalability has been widely adopted in many topics within the image processing community. For instance, multi-resolution analysis has been largely employed in image coding to exploit similarity of an images across scale changes [Mal89]. Motion estimation in the field of video analysis and coding has also benefitted from scalability [BAHH92]. Applying the concept of scalability to human motion capture has not yet been addressed thoroughly in the literature. Some techniques have been presented tailored to a specific application or in a very ad hoc fashion, without a wider perspective of the problem or its implications. Exploiting the scalability potential of the human body model has been addressed from two perspectives: model-based and algorithm-based.

In the model-based approach, the employed human body model is modified along the analysis of a given input data to progressively improve the fitting process. Foures *et al.* [FJ06], used several human body models in a scheme to heuristically search body parts in 2D with limited evidence of its performance. In an attempt to simplify the inverse kinematics problem, Theobalt *et al.* [TMSS02] propose a 2-layer kinematic model. A very coarse and unconstrained layer is first fitted onto the tracked body parts. The second layer, containing the correct kinematic constraints, is then adjusted onto the data from the first layer. More specifically, in the first layer, an arm is only represented by the vector linking the shoulder to the hand. The possible positions for the elbow are therefore constrained to a circle in the second layer, and the best solution is found iteratively. The main problem with this approach is that it assumes that specific body parts (hands and feet) can be tracked reliably, which is rarely the case.

Scalability within the fitting process can be addressed from an algorithmic point of view, using the same HBM along all analysis process. These algorithms take advantage of the topology of the HBM towards performing a progressive fitting of it. The already mentioned hierarchical sampling by Mitchelson *et al.* [MH03] has been perhaps the most relevant contribution where the fitting process is performed progressively through the limbs of the body in a layered scheme. However, inter-relations among limbs are not considered.

7.2.2 Definition

A Scalable Human Body Model (SHBM) can be defined as a set of HBMs:

$$\mathcal{M} = \{\mathcal{H}_1, \mathcal{H}_2, \cdots, \mathcal{H}_M\}. \tag{7.1}$$

To achieve scalability, a certain hierarchy among the elements of $\mathcal{M}$ must be defined. This hierarchy is obtained when its elements $\mathcal{H}_i$ fulfill, at least, one of the following conditions:

1. **Inclusion:** Inclusion among elements of the SHBM defined as:

$$\mathcal{H}_i \subset \mathcal{H}_j, \qquad i < j, \tag{7.2}$$

where the inclusion operation can be understood in terms of the scalability criterion. This criterion is a design parameter and can be defined, for instance, as the number of elements in the body parameter, the information encoded in every model, etc. An example of detail scalability of the human body is depicted in Fig.7.2. Under this condition, an order among $\mathcal{H}_i$ is stated.

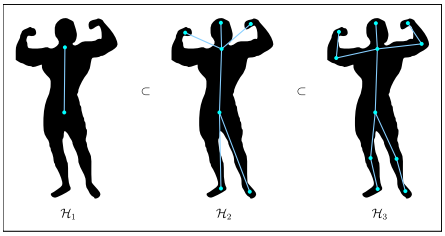


Figure 7.2: *Example of inclusive Scalable Human Body Model in terms of model detail. Most detailed model $\mathcal{H}_{i+2}$ is understood as a refinement of $\mathcal{H}_{i+1}$ since it captures all information and add some more elements of the body model (upper body part).*

2. **Union:** An upper hierarchy element of $\mathcal{M}$ is formed by the union of other elements as:

$$\mathcal{H}_{i_j} \cup \mathcal{H}_{i_k} = \mathcal{H}_{i+1_l}, \qquad j \neq k, \tag{7.3}$$

where union operator is again understood in terms of the scalability criterion. This property leads to a tree hierarchy among elements of $\mathcal{M}$ since two (or more) elements of a given hierarchy level i, $\mathcal{H}_{i_j}$ and $\mathcal{H}_{i_k}$, are united to form another element of a higher level $i+1$, $\mathcal{H}_{i+1_l}$. An example of this condition is shown in Fig.7.3.

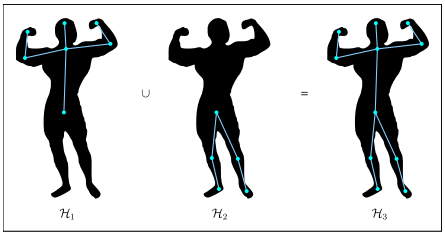


Figure 7.3: *Example of unitive Scalable Human Body Model in terms of model detail. Upper elements in the hierarchy are the result of the union of less detailed models.*

Both hierarchy conditions can be fulfilled in a more elaborate SHBM as depicted in Figure 7.4. Finally, let us define the process of shifting from a lower to a higher hierarchy HBM within the SHBM as an *explicitation*. The opposite shifting from higher to lower hierarchy SHBM will be denoted as *ambiguation*.

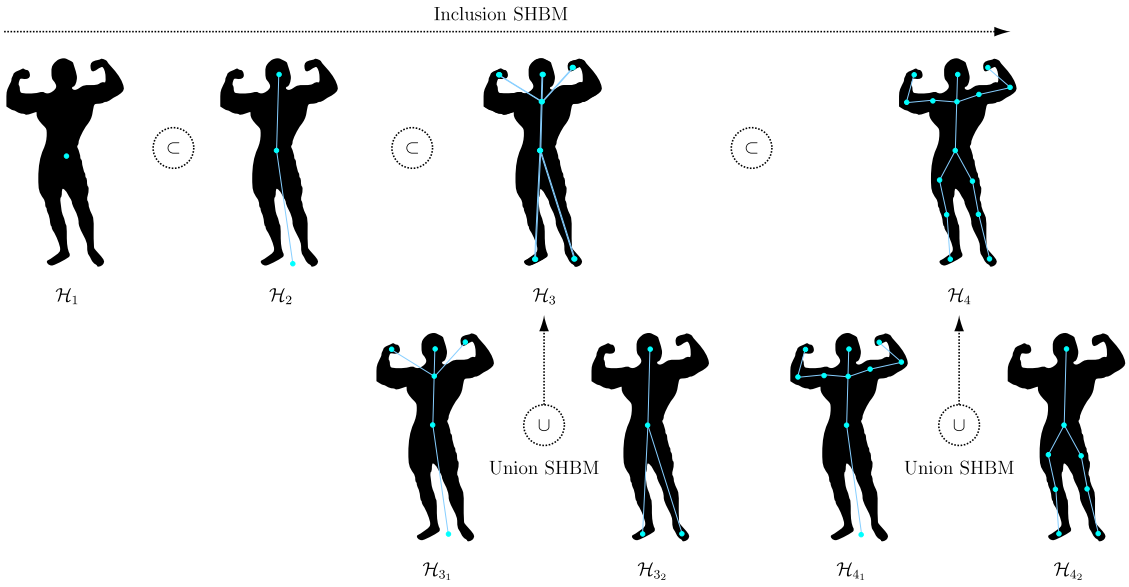


Figure 7.4: *Example of a complex SHBM model including both inclusion and union hierarchy criteria. Both linear and tree structures of inclusive and unitive models allow a broad representation of the human body model with a variable degree of* resolution.

7.3 Scalable Human Body Model Annealed Particle Filter

The underlying hierarchical structure of an articulated object can be exploited towards designing more robust pose estimation and tracking systems. We first propose the scalable human body model annealed particle filter (SHBM-APF) as a filtering technique mimicking the bottom-up and multi-resolution concepts into the HMC field. Assuming that a SHBM $\mathcal{M}$ with a given hierarchy has been stated, we define a sequential fitting process over the several HBMs $\mathcal{H}_i \in \mathcal{M}$. In order to carry out this task, we borrow the idea of annealing [DR05] where particles are placed around the peaks of the likelihood function by means of a recursive search over a set of decreasingly smoothed versions of this function. Our proposal is to use the set of progressively refined HBMs contained in $\mathcal{M}$ instead of a set of smoothed versions of the likelihood function. This process mimics the annealing idea of the coarse-to-fine analysis of the likelihood function thus leading to what we denote as a *structural annealing* process.

For the SHBM-APF, only inclusive SHBMs will be considered. This type of models are the best suited to define an structural annealing process since they progressively refine the model structure until reaching the most detailed model. Although we might define a SHBM-APF filter based on unitive SHBMs, defining variables contained among several models are disjoint hence not fulfilling the refinement idea. In §7.4 it will be shown how unitive SHBMs exhibit some useful properties to design adaptive filtering strategies.

7.3.1 Filter description

Let us have a SHBM $\mathcal{M}$ whose elements $\mathcal{H}_i$ fulfill the inclusive hierarchy criteria and denote the state space associated to HBM $\mathcal{H}_i$ as $\mathcal{X}_{\mathcal{H}_i} = [\theta_1 \cdots \theta_{K_{\mathcal{H}_i}}] \subset \mathbb{R}^{K_{\mathcal{H}_i}}$, where $K_{\mathcal{H}_i}$ is the associated dimension of the model $\mathcal{H}_i$. If the SHBM is properly defined, its elements will fulfill that:

$$\mathcal{X}_{\mathcal{H}_1} \subset \mathcal{X}_{\mathcal{H}_2} \subset \cdots \subset \mathcal{X}_{\mathcal{H}_M}, \tag{7.4}$$

$$K_{\mathcal{H}_1} < K_{\mathcal{H}_2} < \cdots < K_{\mathcal{H}_M}. \tag{7.5}$$

These conditions state the relation between two HMBs $\mathcal{H}_i$ and $\mathcal{H}_j$, $i < j$, as $\mathcal{H}_i$ being a subset of $\mathcal{H}_j$ with strict lower dimension. Concretely, we will design SHBMs following the rule:

$$\mathcal{X}_{\mathcal{H}_i} = \left[f(\mathcal{X}_{\mathcal{H}_{i-1}})\ \theta_{K_{\mathcal{H}_{i-1}}+1}\ \cdots\ \theta_{K_{\mathcal{H}_i}} \right]. \tag{7.6}$$

The state space $\mathcal{X}_{\mathcal{H}_i}$ is designed to contain information directly related to the variables from the previous model $\mathcal{H}_{i-1}$ and, recursively, from all previous models. Function $f(\cdot)$ is intended to perform the mapping among variables between two consecutive state spaces and typically involves a linear (or trivial) mapping. If we define $\mathcal{X}^{\Delta}_{\mathcal{H}_i} = \{\theta_m \in \mathcal{X}_{\mathcal{H}_i} | \theta_m \notin \mathcal{X}_{\mathcal{H}_{i-1}}\}$, Eq.7.6 can be rewritten as:

$$\mathcal{X}_{\mathcal{H}_i} = \left[f(\mathcal{X}_{\mathcal{H}_{i-1}})\ \mathcal{X}^{\Delta}_{\mathcal{H}_i} \right], \tag{7.7}$$

where the associated dimension of $\mathcal{X}^{\Delta}_{\mathcal{H}_i}$ is:

$$\dim\left(\mathcal{X}^{\Delta}_{\mathcal{H}_i}\right) = L \equiv K^{\Delta}_{\mathcal{H}_i} \tag{7.8}$$

Let us have a PF associated to each state space $\mathcal{X}_{\mathcal{H}_i}$, with its associated particle set $\{(\mathbf{y}^j_t, \pi^j_t)\}^{\mathcal{H}_i}$, containing $N_{\mathcal{H}_i}$ particles. The overall operation of the proposed SHBM-APF scheme is to filter the initial distribution associated to the simplest HBM $\mathcal{H}_1$ and then combine the resulting particle set with the initial particle set of the following model, $\mathcal{H}_2$. In this way, information from already filtered variables of $\mathcal{H}_1$ improves the initial particle set associated to $\mathcal{H}_2$. This process is performed for all the models in the SHBM until reaching the last one. Information contained by the particle set of the last model is back-propagated to the models with lower hierarchy rank thus refining their associated particle sets and closing the information filtering loop. The scheme of the proposed technique is depicted in Fig.7.5 for $M = 3$. A fitness function $w_{\mathcal{H}_i}(\mathbf{y}^j_t, \mathbf{z}_t)$ measuring the likelihood between a particle state $\mathbf{y}^j_t$ and the incoming data $\mathbf{z}_t$ is also constructed based on the considerations introduced in §6.5.

When a new measurement $\mathbf{z}_t$ is available, a structural annealing iteration is performed. The SHBM-APF can be summarized as follows:

1. Starting from model $\mathcal{H}_1$, its associated particle set $\{(\mathbf{y}^j_{t-1}, \pi^j_{t-1})\}^{\mathcal{H}_1}$ is resampled with replacement. Then the filtered state $\{(\tilde{\mathbf{y}}^j_t, \tilde{\pi}^j_t)\}^{\mathcal{H}_1}$ is constructed by applying a propagation model $P(\mathbf{y}^j_t, \Sigma_{\mathcal{H}_1})$ and the weighting function $w_{\mathcal{H}_1}(\tilde{\mathbf{y}}^j_t, \mathbf{z}_t)$ to every particle as:

$$\tilde{\mathbf{y}}^j_t = P(\mathbf{y}^j_t, \Sigma_{\mathcal{H}_1}) = \mathcal{N}^{\star}(\mathbf{y}^j_t, \Sigma_{\mathcal{H}_1}) \tag{7.9}$$

$$\tilde{\pi}^j_t = w_{\mathcal{H}_1}(\tilde{\mathbf{y}}^j_t, \mathbf{z}_t), \tag{7.10}$$

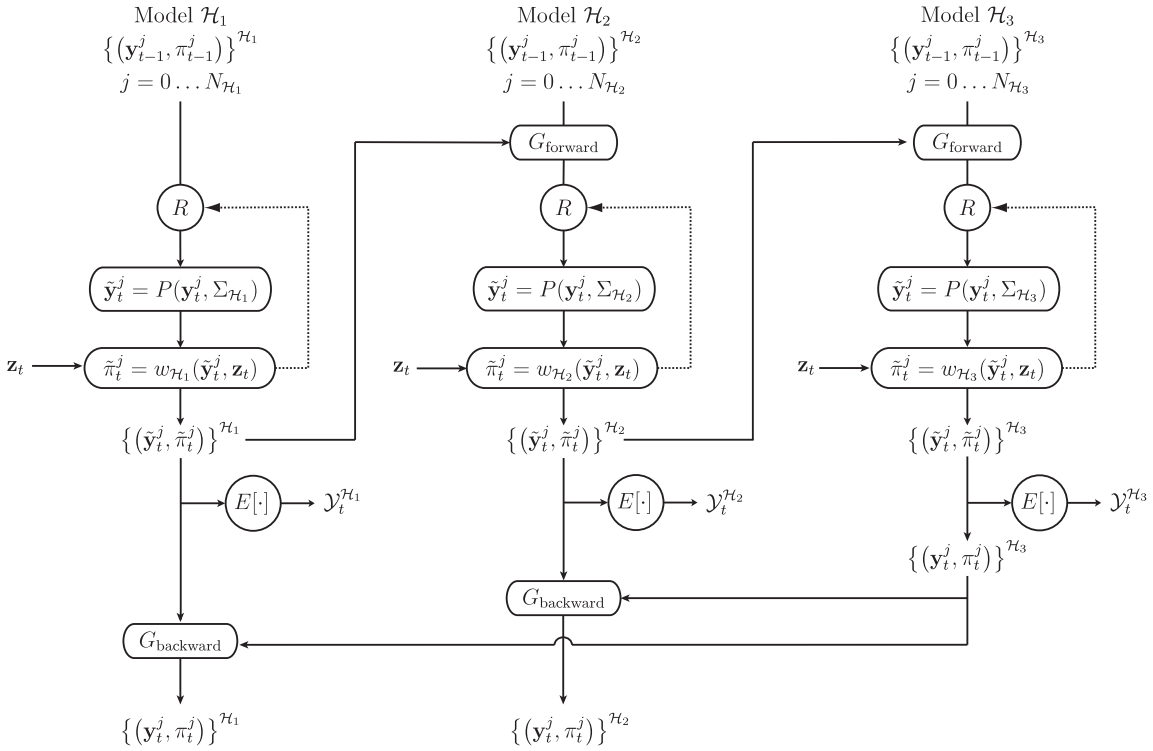


Figure 7.5: *Scalable Human Body Model Annealing Particle Filter (SHBM-APF) scheme for $M = 3$ elements in the SHBM fulfilling the inclusive criteria.*

where $\mathcal{N}^\star$ is the truncated multivariate Gaussian random variable centered at $\mathbf{y}^j_t$ with diagonal covariance matrix $\Sigma = \text{diag}\{\boldsymbol{\sigma}_{\mathcal{H}_1}\}$ presented in §6.3.4. Weights are normalized such that $\sum_j \tilde{\pi}^j_t = 1$. At this point, the output estimation of this model $\mathcal{Y}^{\mathcal{H}_1}_t$ can be computed by applying

$$\mathcal{Y}^{\mathcal{H}_1}_t = \sum_{j=1}^{N_{\mathcal{H}_1}} \tilde{\pi}^j_t \tilde{\mathbf{y}}^j_t. \tag{7.11}$$

2. For the following HBMs, $i > 1$, the filtered particle set of the previous model in the hierarchy, $\{(\tilde{\mathbf{y}}^j_t, \tilde{\pi}^j_t)\}^{\mathcal{H}_{i-1}}$, is combined through the operator G_{forward} with the particle set associated to model $\mathcal{H}_i$, $\{(\mathbf{y}^j_{t-1}, \pi^j_{t-1})\}^{\mathcal{H}_i}$. State space variables associated to $\mathcal{H}_i$ contain information from model $\mathcal{H}_{i-1}$ due to the imposed hierarchy relation. Since these common variables have been already filtered, the updated information can be transferred to the particles of model $\mathcal{H}_i$ in order to generate an improved initial particle set (a further review of G_{forward} is presented in §7.3.2).

Then, the filtered state $\{(\tilde{\mathbf{y}}_t^j, \tilde{\pi})_t^j\}^{\mathcal{H}_i}$ is constructed as:

$$\tilde{\mathbf{y}}_t^j = P(\mathbf{y}_t^j, \Sigma_{\mathcal{H}_i}) = \mathcal{N}^\star(\mathbf{y}_t^j, \Sigma_{\mathcal{H}_i}) \quad (7.12)$$

$$\tilde{\pi}_t^j = w_{\mathcal{H}_i}(\tilde{\mathbf{y}}_t^j, \mathbf{z}_t), \quad (7.13)$$

where $\mathcal{N}^\star$ is a truncated multivariate Gaussian random variable centered at $\tilde{\mathbf{y}}_t^j$ with a covariance matrix $\Sigma_{\mathcal{H}_i} = \text{diag}\{\alpha^{i-1}\boldsymbol{\sigma}_{\mathcal{H}_1}, \alpha^{i-2}\boldsymbol{\sigma}_{\mathcal{H}_2}^{\Delta}, \ldots, \boldsymbol{\sigma}_{\mathcal{H}_i}^{\Delta}\}$ and $\alpha < 1$. Essentially, the covariance matrix is constructed recursively in the following way:

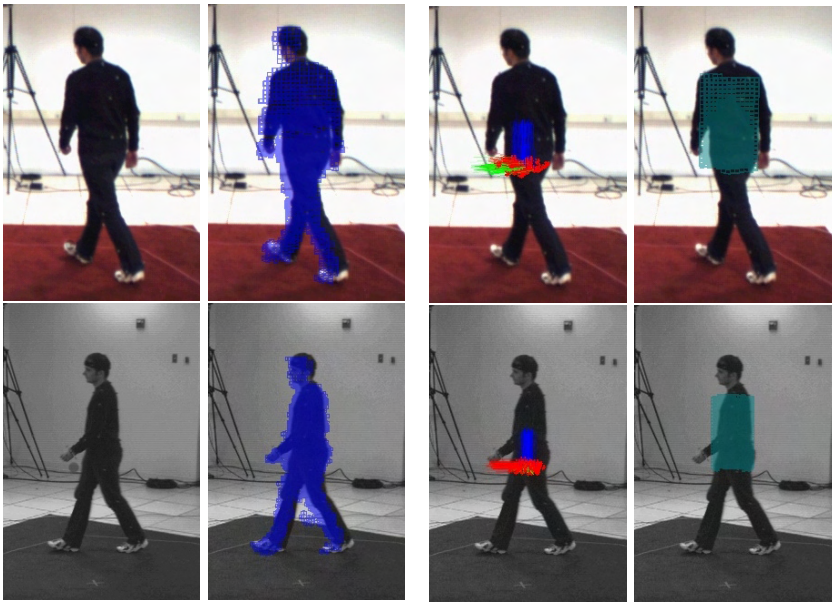
$$\Sigma_{\mathcal{H}_i} = \begin{bmatrix} \ddots & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha\boldsymbol{\sigma}_{\mathcal{H}_{i-1}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{\sigma}_{\mathcal{H}_i}^{\Delta} & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots \end{bmatrix}, \quad (7.14)$$

where $\boldsymbol{\sigma}_{\mathcal{H}_i}^{\Delta}$ stands for the variance of the variables associated to the incremental state space with respect to $\mathcal{H}_{i-1}$, $\mathcal{X}_{\mathcal{H}_i}^{\Delta}$. This propagation function assigns a higher drift to the newly added variables of model $\mathcal{H}_i$ while assigning a lower drift to those that have been more recently filtered in the previous layers. At this point, the output estimation of this model $\mathcal{Y}_t^{\mathcal{H}_i}$ can be computed.

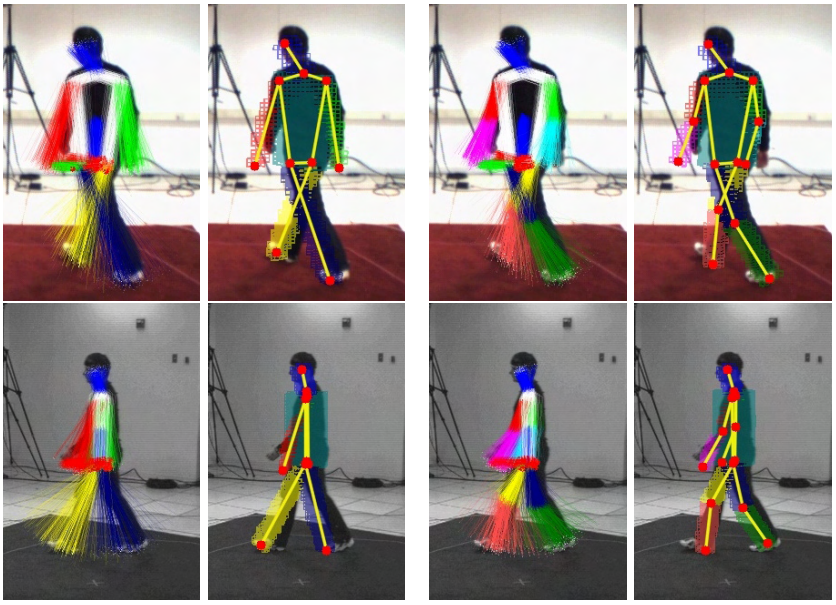
3. Once reaching the highest hierarchy level, that is the most detailed HBM, the information contained in the particle set $\{(\tilde{\mathbf{y}}_t^j, \tilde{\pi}_t^j)\}^{\mathcal{H}_M}$ is back-propagated to the other models in the hierarchy by means of the operator $G_{\text{backwards}}$. In this way, the particle sets of every model are refined thus closing the filtering loop.

An example of the execution of this scheme is depicted in Fig.7.6.

In order to refine the particle set $\{(\tilde{\mathbf{y}}_t^j, \tilde{\pi}_t^j)\}^{\mathcal{H}_i}$ that will be transferred to layer $i+1$, we added a standard annealing loop represented as a dashed line in the overall scheme. This will concentrate the particles around the main modes of the likelihood function at layer i before delivering them to layer $i+1$. It must be noted that an accurate likelihood estimation at a lower layer will benefit the subsequent estimation layers. For further discussion, let us denote as $L_{\mathcal{H}_i}$ the number of annealing layers associated to the filtering thread associated to model $\mathcal{H}_i$. The presented configuration can be seen as a filtering scheme with a double annealing loop: one in the model complexity, benefiting from the hierarchical properties of the SHBM, and the second associated to the filtering branch associated to each model, benefiting from the already mentioned likelihood annealing properties. Let us denote as $\alpha \equiv \alpha_{\text{S}}$ the variance decrease rate among two consecutive HBMs from Eq.7.14 and $\alpha_{\mathcal{H}_i}$ as the variance decrease rate in the inner likelihood annealing loop associated to a given HBM $\mathcal{H}_i$.

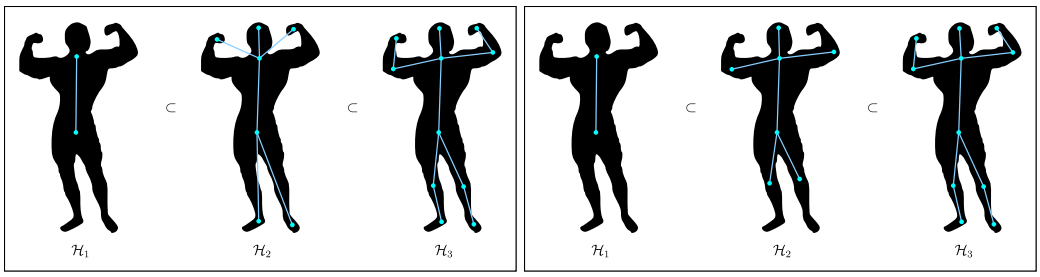


(a) Input data (b) Model $\mathcal{H}_0$



(c) Model $\mathcal{H}_1$ (d) Model $\mathcal{H}_2$

Figure 7.6: *Example of SHBM-APF algorithm operation from two camera views. In (a), the input data ($s_{\mathcal{V}} = 3$ cm) while, in (b)-(d), the successive filtering threads associated to every HBM $\mathcal{H}_i$. Particles $\{(\tilde{\mathbf{y}}_t^j, \tilde{\pi}_t^j)\}^{\mathcal{H}_M}$ and estimation $\mathcal{Y}_t^{\mathcal{H}_i}$ for every HBM are displayed.*



(a) Model 1 (b) Model 2

Figure 7.7: *Two SHBM analysis models employed in the SHBM-APF algorithm.*

7.3.2 Filter implementation

7.3.2.1 SHBM choice

In order to define the structure of the SHBM-APF filter, the mapping functions $f(\cdot)$ from Eq.7.7 and the number of layers $L_{\mathcal{H}_i}$, we should define the analysis SHBM. Two proposals of inclusive SHBM are presented, depicted in Figure 7.7 (note that there is a slight difference between the two models in the $\mathcal{H}_2$ element). In model 1, the overall limbs orientation is first estimated and then the remaining joints whereas, in model 2, the limbs orientation is progressively estimated. Each model has its advantages and drawbacks. In motions where limbs are mostly straight (i.e. walking or running), model 1 was proved to be more adequate capturing the overall orientation of each limb (hips and shoulders joints) and then refining the estimation (knees and elbows). Other types of motion like gesturing are better captured using model 2.

The associated state spaces to both models are identical:

$$\mathcal{X}_{\mathcal{H}_1} = [\mathbf{x}\ \mathbf{R}], \tag{7.15}$$

$$\mathcal{X}_{\mathcal{H}_2} = \big[\mathcal{X}_{\mathcal{H}_1}\ \theta_x^{\text{Neck}}\ \theta_y^{\text{Neck}}\ \theta_x^{\text{R.Should.}}\ \theta_z^{\text{R.Should.}}\ \theta_x^{\text{L.Should.}}\ \theta_z^{\text{L.Should.}}\ \ldots \\ \ldots\ \theta_x^{\text{R.Hip}}\ \theta_y^{\text{R.Hip}}\ \theta_x^{\text{L.Hip}}\ \theta_y^{\text{L.Hip}}\big], \tag{7.16}$$

$$\mathcal{X}_{\mathcal{H}_3} = \big[\mathcal{X}_{\mathcal{H}_2}\ \theta_y^{\text{R.Should.}}\ \theta_z^{\text{R.Elbow}}\ \theta_y^{\text{L.Should.}}\ \theta_z^{\text{L.Elbow}}\ \theta_z^{\text{R.Hip}}\ \theta_y^{\text{R.Knee}}\ \theta_z^{\text{L.Hip}}\ \theta_y^{\text{L.Knee}}\big] \tag{7.17}$$

where $\mathbf{x}$ and $\mathbf{R}$ stand for the global translation and rotation w.r.t. origin of coordinates, respectively. Note that the mapping functions $f(\cdot)$ in our case are trivial. Regarding the associated dimensions $K_{\mathcal{H}_i}$ introduced in Eq.7.5, we have that:

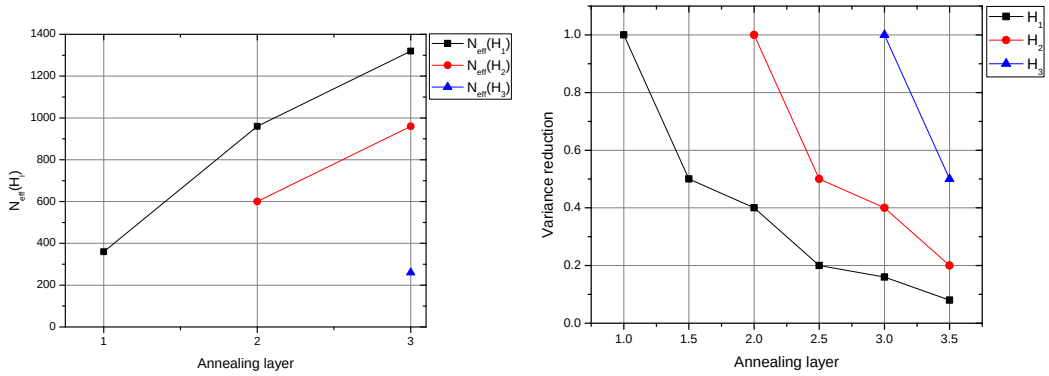
$$K_{\mathcal{H}_1} = 6, \qquad K_{\mathcal{H}_2} = 16, \qquad K_{\mathcal{H}_3} = 22, \tag{7.18}$$

with the following associated incremental state space dimension:

$$K^{\Delta}_{\mathcal{H}_1} = 6, \qquad K^{\Delta}_{\mathcal{H}_2} = 10, \qquad K^{\Delta}_{\mathcal{H}_3} = 6. \tag{7.19}$$

7.3.2.2 Particle assignment

Let us review how to assign values to the number of particles per model thread, $N_{\mathcal{H}_i}$, and the number of layers per model, $L_{\mathcal{H}_i}$. Indeed, an empirical set up obtained by randomly



(a) (b)

Figure 7.8: *Evolution of the number of effective particles, N_{eff}, and relative variance reduction of the different variables associated to every HBM, $\mathcal{H}_i$. The fractional values in the x axis in subfigure (b) stand for the internal annealing layers at every structural layer.*

testing several configurations might be computationally expensive and time consuming, and can not even lead to an intuitive assignation rule. We first define the effective number of particles associated to a given state space as:

$$N_{\text{eff}}\left(\mathcal{X}_{\mathcal{H}_i}\right) = \sum_{j=i}^{M} N_{\mathcal{H}_j} \cdot L_{\mathcal{H}_j}. \tag{7.20}$$

From this equation, it can be seen that variables from a given HBM, $\mathcal{H}_i$, are filtered by all following structural annealing layers. For instance, variables from the first model $\mathcal{H}_1$ associated to the global position and orientation of the torso will be filtered by all structural annealing layers since they are a set of relevant variables. Variables associated to the last model of the hierarchy, being less important, are only filtered by their layers. The figure N_{eff} is displayed in Figure 7.8.

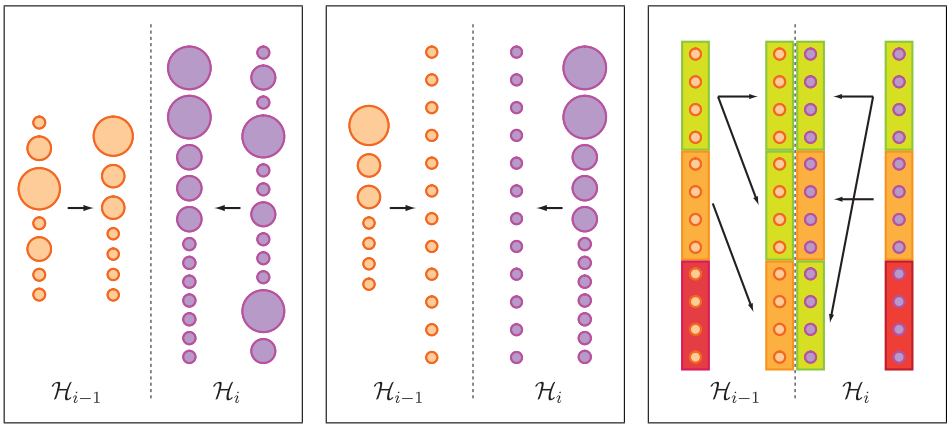
We devised the following method to assign the number of particles per structural annealing layer as the increment of dimensionality between state spaces:

$$N_{\mathcal{H}_i} \propto K_{\mathcal{H}_i}^{\Delta}. \tag{7.21}$$

This methodology will be tested with experiments in §7.5.1.

7.3.2.3 G_{forward} operator: Adaptive Resampling and Genetic Crossing

When the particle set associated to the HBM $\mathcal{H}_{i-1}$, $\{(\tilde{\mathbf{y}}_t^j, \tilde{\pi}_t^j)\}^{\mathcal{H}_{i-1}}$, has been filtered (with annealing), the encoded *pdf* is transferred to the next model, $\mathcal{H}_i$, to improve its associated initial particle set $\{(\mathbf{y}_t^j, \pi_t^j)\}^{\mathcal{H}_i}$. This information delivery is carried out by the operator G_{forward} that has to deal with two problems: the difference of the number of particles associated to each filtering thread ($N_{\mathcal{H}_i} \neq N_{\mathcal{H}_{i-1}}$) and the dimension difference between $\mathcal{X}_{\mathcal{H}_i}$ and $\mathcal{X}_{\mathcal{H}_{i-1}}$ ($K_{\mathcal{H}_i} > K_{\mathcal{H}_{i-1}}$). The first issue is addressed by a new technique



(a) Sorting (b) Adaptive resampling (c) Genetic crossing

Figure 7.9: *Combination process of particles from two different state spaces corresponding to two different HBMs following the sorting, adaptive resampling and genetic crossing methodology.*

called *sorting and adaptive resampling* while the second is tackled by means of *genetic crossing*.

A first step to combine information from two different particle sets is to have the same number of elements in each set. Moreover, these two sets, $\{(\tilde{\mathbf{y}}_t^j, \tilde{\pi}_t^j)\}^{\mathcal{H}_{i-1}}$ and $\{(\mathbf{y}_{t-1}^j, \pi_{t-1}^j)\}^{\mathcal{H}_i}$, are both weighted (that is $\pi_t^j \neq N_{\mathcal{H}_i}^{-1}$, $\forall j, i$). We address this problem following the sorting and adaptive resampling procedure described as follows:

1. **Sorting:** Internal order within the particle set has not been considered previously since it does not affect the filtering operation. For the presented strategy, this order will be taken into account hence particles will be sorted decreasingly according to their associated weights. This operation will be applied to both particle sets as shown in Figure 7.9(a).

2. **Adaptive resampling:** Systematic resampling presented in §3.1.2.1 is designed to produce an output set of $N_{\mathrm{p}}^{\mathrm{output}}$ resampled particles from an input set of the same number of elements, $N_{\mathrm{p}}^{\mathrm{input}}$. However, the problem of generating an output set with $N_{\mathrm{p}}^{\mathrm{output}} \neq N_{\mathrm{p}}^{\mathrm{input}}$ has not been addressed in the literature, as it is a very unusual requirement. For our purposes, we designed a variant of Algorithm 2 to do so, reported in Algorithm 4 and depicted in Figure 7.9(b). Particles associated to model $\mathcal{H}_i$ are not affected by this adaptive resampling since $N_{\mathrm{p}}^{\mathrm{output}} = N_{\mathrm{p}}^{\mathrm{input}}$ and, in such cases, our resampling proposal performs as in the standard case.

   Note that resampling does not alter the order of the input and output vector. According to the previously applied sorting, particles with lower index in the output vector are resampled from particles with a higher weight in the input vector. Hence, despite all particles have the same weight, we can still distinguish which ones are the most relevant, according to their index value.

Merging information from the two particle sets once they have the same number of elements requires a criteria to combine their defining state space variables, $\mathcal{X}_{\mathcal{H}_{i-1}}$ and $\mathcal{X}_{\mathcal{H}_i}$. According to the SHBM construction stated in Eqs.7.15, 7.16 and 7.17, a model $\mathcal{H}_i$ includes all variables from the preceding models. Hence, since variables from model $\mathcal{H}_{i-1}$ have been already filtered, we propose to combine these two particle sets by disregarding information from model $\mathcal{H}_{i-1}$ in model $\mathcal{H}_i$ and replacing these variables with the already filtered ones from the preceding filtering thread. That is:

$$\tilde{\mathbf{y}}_t^j \in \mathcal{X}_{\mathcal{H}_{i-1}}, \qquad \mathbf{y}_{t-1}^k \in \mathcal{X}_{\mathcal{H}_i}, \qquad \mathbf{y}_{t-1}^{k,\Delta} \in \mathcal{X}_{\mathcal{H}_i}^{\Delta},$$

$$\mathbf{y}_{t-1}^k = \left[\tilde{\mathbf{y}}_t^j \; \mathbf{y}_{t-1}^{k,\Delta}\right]. \tag{7.22}$$

As we will see with the definition of the G_{backward} operation, no information is lost by overwriting variables from one model to the next one. Note that indices i and k in Eq.7.22 are not set to be the same and indeed, the procedure to associate these two indices will conform the particle combination algorithm.

Once the adaptively resampled particles belonging to two consecutive HBM have been generated (see Figure 7.9(b)), we might design a way to combine them in order to produce a resulting particle set, $\{(\mathbf{y}_{t-1}^j, \pi_{t-1}^j)\}^{\mathcal{H}_i}$, benefiting from this already filtered information. We first considered a direct association, $i \rightarrow j$, combining the best particles of each model. However, this procedure proved unable to cope with fast and unexpected motion since weak hypothesis from both models were rarely considered. Instead, inspired on genetic algorithms [Mit98] (although not following its methodology), we defined a more versatile particle combination method. Recently, Ye *et al.* [YZG08] exploited genetic algorithms together with PF to perform HMC by efficiently exploring the HBM's state space.

Genetic or biologically inspired algorithms are based on the breeding mechanisms present in Nature. In our case, we applied such ideas to develop the following particle cross-over technique. Let us first define a partition over our sorted and resampled particle set of equal size denoted as $\mathcal{S}_{\mathcal{H}_i}^n$, $1 \leq n \leq N$, being N the number of partitions. Again, the lower the n index, the more relevant the partition (in terms of resample particles originated from particles with higher weights). Then we define an association rule between the sets $\mathcal{S}_{\mathcal{H}_{i-1}}^n$ and $\mathcal{S}_{\mathcal{H}_i}^m$ based on generating combinations of sets with high indices but also allowing combinations of sets with high and low indices. In this way, some variability is introduced thus becoming more robust to rapid motion and sudden pose changes. The rule to generate such index correspondence has been set empirically as shown in Figure 7.9(c).

7.3.2.4 Propagation

Propagation of particles has been addressed in §7.3.1 and basically depends on the initial variances associated to each model filtering thread, $\Sigma_{\mathcal{H}_i}$, the structural annealing variance reduction, α_{S}, and the annealing variance reduction within the same filtering thread, $\alpha_{\mathcal{H}_i}$. It must be remarked that the variance associated to a given HBM variable should always decrease as it is processed either through an inner or a structural annealing loop. Hence, for the models presented in §7.3.2, we can state the initial $\Sigma_{\mathcal{H}_i}$ variances

Algorithm 4: Systematic Adaptive Resampling Algorithm

$c_1 = \pi_t^1$
for $j = 2$ **to** N_p^{input} **do**
  $c_j = c_{j-1} + \pi_t^j$
end
Draw a starting point $u_1 \sim \mathbf{U}[0, 1/N_{\mathrm{p}}^{\mathrm{output}}]$
$j = 1$
for $i = 1$ **to** N_p^{output} **do**
  $u_i = u_1 + (i-1)/N_{\mathrm{p}}^{\mathrm{output}}$
  while $u_i > c_j$ & $j < N_p^{input}$ **do**
    $j = j + 1$
  end
  $\{\mathbf{x}_t^i, \pi_t^i\} = \{\mathbf{x}_t^j, 1/N_{\mathrm{p}}^{\mathrm{output}}\}$
end

as:

$$\Sigma_{\mathcal{H}_1} = \mathrm{diag}\left\{\sigma_{\mathcal{H}_1}\right\}, \tag{7.23}$$

$$\Sigma_{\mathcal{H}_2} = \mathrm{diag}\left\{\left(\alpha_{\mathrm{S}}\alpha_{\mathcal{H}_1}^{L_{\mathcal{H}_1}}\right)\sigma_{\mathcal{H}_1}, \sigma_{\mathcal{H}_2}^{\Delta}\right\}, \tag{7.24}$$

$$\Sigma_{\mathcal{H}_3} = \mathrm{diag}\left\{\left(\alpha_{\mathrm{S}}^2\alpha_{\mathcal{H}_1}^{L_{\mathcal{H}_1}}\alpha_{\mathcal{H}_2}^{L_{\mathcal{H}_2}}\right)\sigma_{\mathcal{H}_1}, \left(\alpha_{\mathrm{S}}\alpha_{\mathcal{H}_2}^{L_{\mathcal{H}_2}}\right)\sigma_{\mathcal{H}_2}^{\Delta}, \sigma_{\mathcal{H}_3}^{\Delta}\right\}. \tag{7.25}$$

Or, in a more general fashion:

$$\Sigma_{\mathcal{H}_i} = \mathrm{diag}\left\{\left(\alpha_{\mathrm{S}}^{i-1}\prod_{p=1}^{i-1}\alpha_{\mathcal{H}_j}^{L_{\mathcal{H}_j}}\right)\sigma_1, \underbrace{\left(\alpha_{\mathrm{S}}^{q-1}\prod_{p=q}^{i-1}\alpha_{\mathcal{H}_j}^{L_{\mathcal{H}_j}}\right)\sigma_q^{\Delta}}_{q=1\ldots(i-1)}, \sigma_i^{\Delta}\right\} \tag{7.26}$$

7.3.2.5 G_{backward} operator

Once reaching the last HBM model, $\mathcal{H}_M$, the filtered particle set $\{(\mathbf{y}_t^j, \pi_t^j)\}^{\mathcal{H}_M}$ contains the most accurate and detailed estimation of the HBM. Taking into account that state variables of $\mathcal{H}_M$ are also represented in $\mathcal{H}_i$, $i < M$, by means of Eq.7.7, we might use this information to update the already filtered particle sets $\{(\tilde{\mathbf{y}}_t^j, \tilde{\pi}_t^j)\}^{\mathcal{H}_i}$. In this way, the initial particle set to be filtered at time $t+1$ at each HBM analysis thread will be derived from the best estimation at time t. Basically, operator G_{backward} will first sort and generate $M-1$ adaptively resampled sets from $\{(\mathbf{y}_t^j, \pi_t^j)\}^{\mathcal{H}_M}$ with input dimension $K_{\mathcal{H}_M}$ and output dimensions $K_{\mathcal{H}_i}$, $1 \leq i < M$ (note that filtered sets $\{(\tilde{\mathbf{y}}_t^j, \tilde{\pi}_t^j)\}^{\mathcal{H}_i}$ are already ordered from the G_{forward} operation). Then the variables associated to the state space subset $\mathcal{X}_{\mathcal{H}_i}^{\Delta}$ of each HBM are replaced with the values from the adaptively resampled set derived from $\{(\mathbf{y}_t^j, \pi_t^j)\}^{\mathcal{H}_M}$.

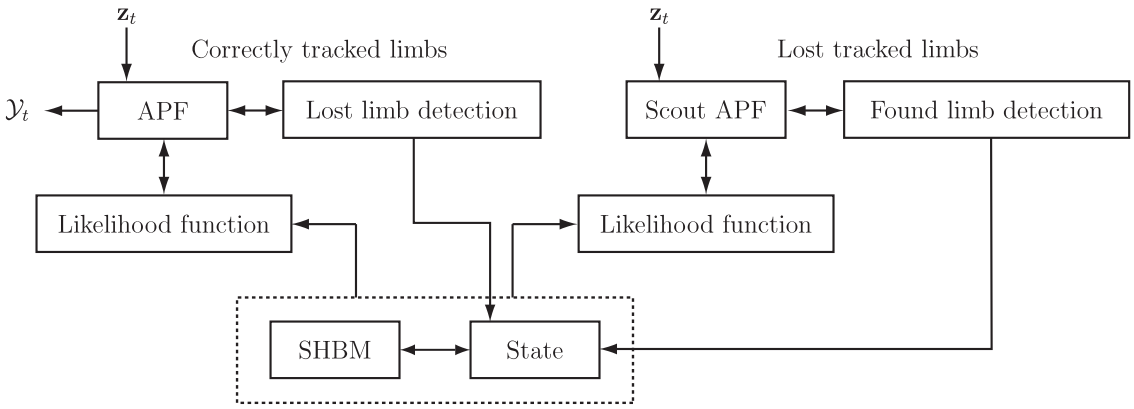


Figure 7.10: *Data Driven Model Adaptive Particle Filter scheme.*

7.4 Data Driven Model Adaptive Particle Filter

Analysis of data corrupted by noise with a given distribution is a classical problem in signal processing and its extension to HMC has been the standard working paradigm addressed in the literature. It has been assumed that input data is moderately corrupted [TMSS02, Mik03] and the structure of the body is faithfully represented by the data. Empty scenarios are the typical analysis environment disregarding other more complex scenarios including occlusive elements such as furniture. Multi-camera HMC research relies on the fact that redundancy among cameras will allow resolving occlusions towards estimating the HBM pose. However, more realistic scenarios will not fulfill these assumptions, thus rendering most of HMC algorithms unsuitable for such task.

The concept of SHBM has been presented in the previous section as an efficient and robust analysis tool to develop the SHBM-APF algorithm. Inclusive models have been employed due to their state space nesting properties (see Eq.7.7). Now we are dealing with a problem of a different nature: data is not only noisy but can also include large missing parts, thus not representing all body parts as depicted in Figure 7.1. Hence, neither the markerless APF nor the SHBM-APF would be able to properly deal with such data. Instead, properties of the unitive HBM introduced in §7.2.2 render them suitable for such task.

7.4.1 Filter description

Let us have a tracking scenario where the subject under analysis is partially occluded like in Figure 7.1(b). When analyzing such scenario with one of the already presented methods, the state variables (that is, the angles) associated with the limbs that are not represented into data will be wrongly estimated. Moreover, the likelihood function may be biased by this limb missing effect. Instead, we could design a tracking algorithm that is aware of the quality of data and employs the most adapted HBM to analyze these data as previously depicted in the fourth column of Figure 7.1. The Data Driven Model Adaptive Particle Filter (DDM-APF) is proposed as a tracking technique to automatically

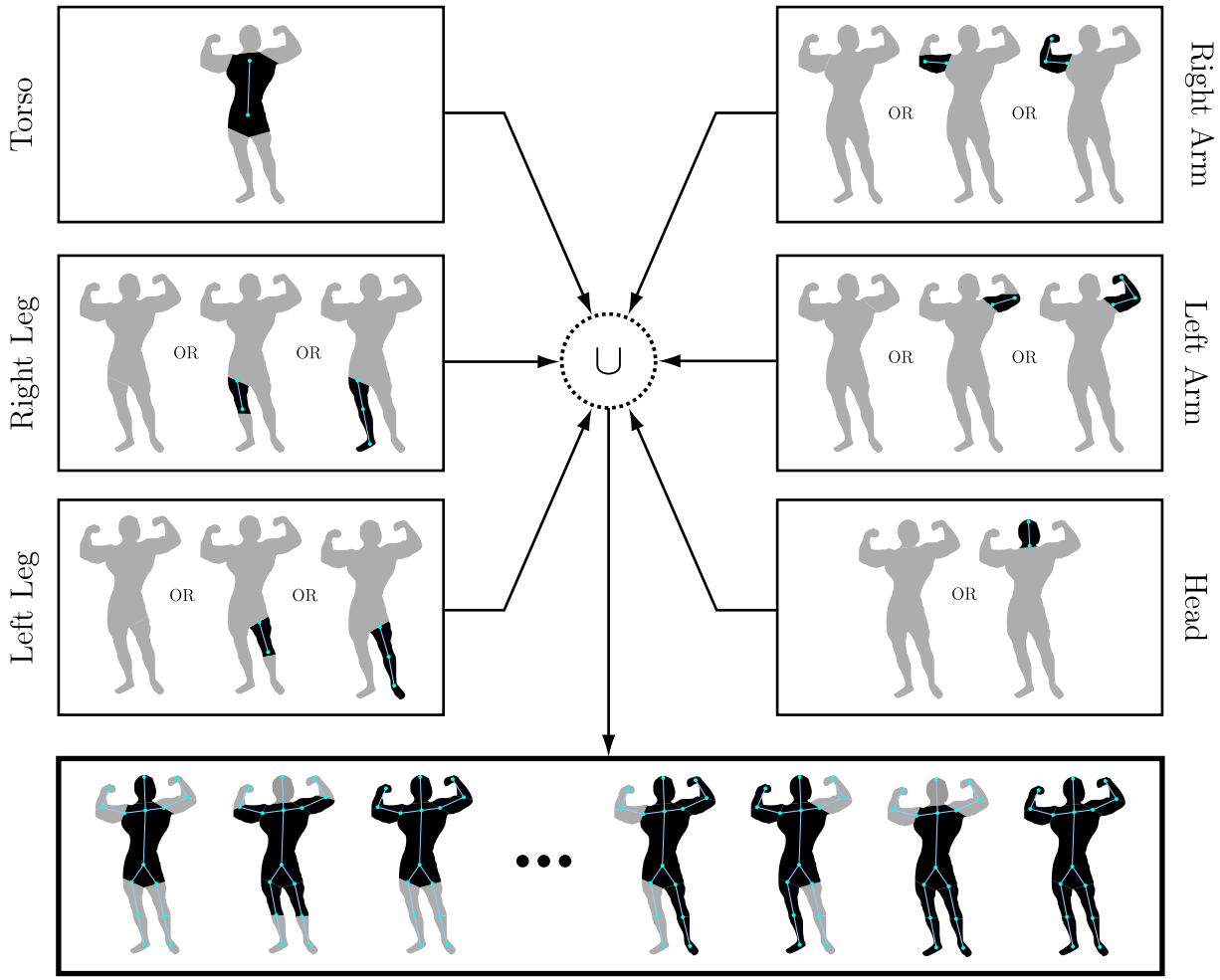


Figure 7.11: *Complex unitive model employed by the DDM-APF algorithm.*

adapt the HBM employed to analyze the input data by selecting the model that better suits these data. The overall processing pipeline is shown in Figure 7.10.

Let us have a SHBM $\mathcal{M}$ based on unitive relationships among its HBMs, $\mathcal{H}_i$, that can describe the structure of the human body with different degrees of completeness. In our case, we propose the model depicted in Figure 7.11 as the working SHBM. Basically, each limb is represented as missing or with a number of sub-parts (legs, forelegs, arms and forearms). The torso is enforced to be always present in our model. Finally, the union of all these sub-models give the family of HBMs shown in the bottom part of the figure. Let us have an associated state vector $\mathbf{s}_t$ that will encode the employed model as a collection of binary states denoted whether a limb part is used or not. Note that there is a number of combinations that have been omitted since the produced HBMs might be rare or might lead to awkward body poses. For instance, we do not consider a HBM with an arm but without the forearm simultaneously or similar configurations.

From the system diagram let us consider the section related with the correctly tracked

limbs, so those that have an active state in the vector $\mathbf{s}$. The operation scheme is as follows:

1. Input data $\mathbf{z}_t$ is fed to an annealed particle filter (APF) described in §6.5. The likelihood function, $w(\mathbf{z}_t, \mathcal{H}_i)$, is constructed based upon the state vector $\mathbf{s}_t$, where the associated volume $\boldsymbol{\mathcal{V}}^{\text{HBM}}$ only contains the active limbs. This process can be observed in Figure 7.12 in two different situations. In case 1, Figure 7.12(a), the likelihood function takes into account all limbs whereas in case 2, Figure 7.12(b), some elements in $\mathbf{s}_t$ are set to false thus denoting that some limbs (legs and left arm) may not be tracked properly. In this case, the likelihood function is constructed based only on the still properly tracked limbs.

2. Once data has been filtered by the APF, we analyze the output data to detect whether a limb has disappeared due to an occlusion or a quality reduction of the input data $\mathbf{z}_t$. In this case, the state vector is modified accordingly to not account for the newly missing limb in the next time processing $t + 1$.

3. Finally, the output $\mathcal{Y}_t$ is produced (together with vector state $\mathbf{s}_t$) following the standard weighted averaging procedure.

The second section of the scheme deals with missing limbs in order to detect whether they have re-appeared into the scene after an occlusion has ended or data is no longer faulty. The seminal idea to build up this section was introduced by Bernardin *et al.* [BGS07] in the context of person tracking and was denoted as *scout* particle filters (SPF). This type of filter places particles following a very broad prior to "discover" a zone of relevant likelihood to initialize a tracking filter in that state space region. SPF were employed to initialize a track in problems related to person tracking. The lost tracked limb section can be summarized as:

1. Every missing limb has an associated SPF that will broadly explore the likely zones where that limb might be found. This effect is achieved by setting a high variance in the propagation noise of the filter. Since we are in an annealing framework, we refined the SPF by using its annealed version, ASPF. We may understand this process as having several ASPF in parallel where each of them try to find a lost limb in the region where this limb should be found. An example of this procedure is depicted in Figure 7.12(b). Legs and the left arm are not properly represented by the input data and a ASPF is associated to each of them, representing their particles in red in the figure. Observe how this ASPF technique is only applied to broadly search for the missing limbs but not affecting the final estimation.

2. Two options may be devised: the lost limb is still not captured by the input data hence the state vector $\mathbf{s}_t$ will not be updated or the limb has re-appeared. In this case, we should change the employed analysis HBM. It must be noted, that when computing the lost limb likelihood associated to every ASPF, interactions with the already tracked limbs should be taken into account. The contrary effect, taking into account the lost limbs when computing the $\boldsymbol{\mathcal{V}}^{\text{HBM}}$ derived scores, is not carried out. Indeed, by doing so, the double occupancy score associated to $\boldsymbol{\mathcal{V}}^{\text{HBM}}$ might

be affected by a limb that is not even present. To achieve this effect, the lost tracked limb section is always computed after the correctly tracked limb section is processed, that is sequentially.

7.4.2 Filter implementation

APF considerations

The APF block in Figure 7.10 is based on the technique presented in §6.5. The already discussed propagation and particle assignment considerations are applied in this framework, as well as the likelihood evaluation and construction of the $\boldsymbol{\mathcal{V}}^{\mathrm{HBM}}$ set.

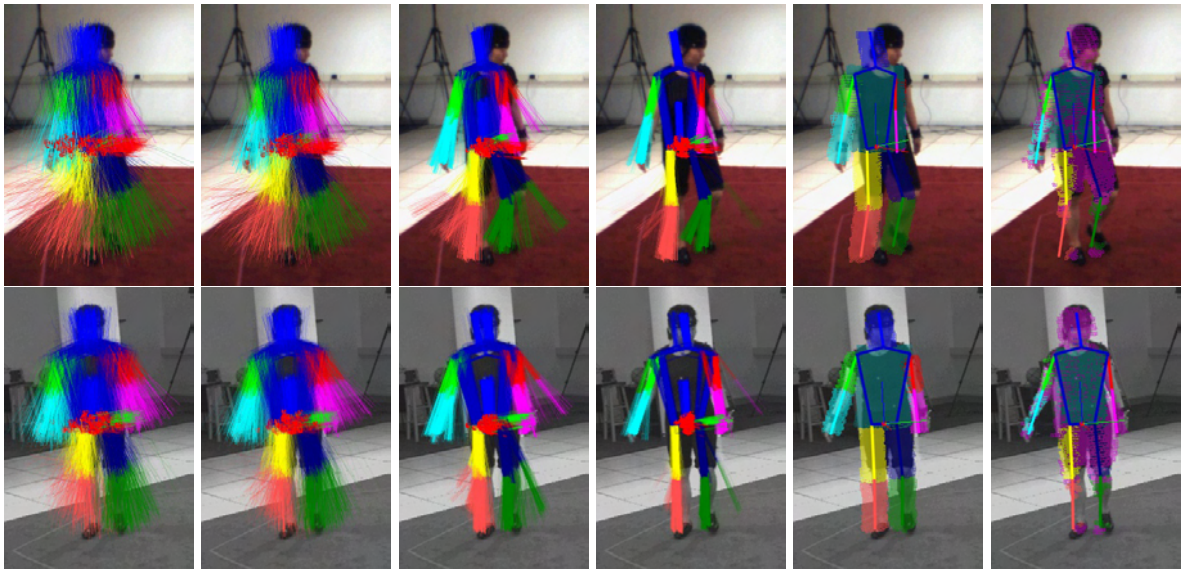
Lost/Found Limb Detection

The criteria to detect whether a limb has ceased to be properly represented by the input data $\mathbf{z}_t$ is based on two measurements. First, we compute the variance of each variable in the state space $\theta_l \in \mathcal{X}_{\mathcal{H}_i}$ after the last annealing iteration has been executed:

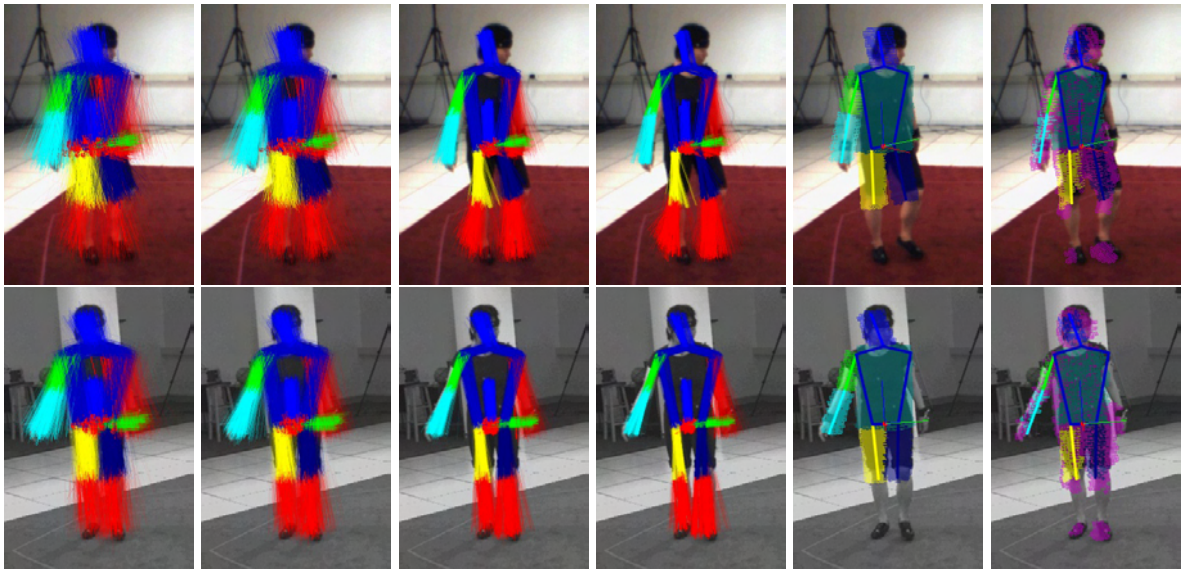
$$\sigma_{\theta_l} = \frac{1}{N_{\mathrm{p,L}}} \sum_{j=1}^{N_{\mathrm{p,L}}} \left(\theta_l^j - \bar{\theta}_l^j \right)^2 . \tag{7.27}$$

A large value of σ_{θ_l} demonstrates that the tracker could not lock onto any mode of the likelihood function thus indicating that there are no data to be assigned to such limb. Since the location of a limb is usually governed by a number of angles associated to an articulation, we will set a limb to be missing if all these angles present a large variance. The second criterion is employed to verify the decision taken based on the variance analysis. We compute the occupancy score (see Eq.6.15) associated to the average of all particles in the analyzed limb and check that this figure attains a low value. If both conditions are fulfilled, the limb is marked in the state vector $\mathbf{s}_t$ as lost. Thresholds associated to both conditions are set by hand where $\sigma_{\theta_l} > 15^{\mathrm{o}}$ and $\rho^{\mathrm{Occ}} < 0.05$ have been found to produce satisfactory results. Nonetheless, the system is not extremely sensitive to these values since the two measurements attain extreme values (that is $\sigma_{\theta_l} \gg 15^{\mathrm{o}}$ and $\rho^{\mathrm{Occ}} \approx 0$) when a limb is missing.

The mechanism employed to detect if a limb has again been captured by the input data is similar to the lost limb detection procedure. Assuming that every lost limb has an ASPF associated, we compute the same two scores for each of them: variance and occupancy of the $\boldsymbol{\mathcal{V}}^{\mathrm{HBM}}$ part associated to that limb after computing the average of all particles in the SAPF. In this case, a low variance might indicate that the ASPF have locked onto data located in the spatial region more likely to contain the searched limb. Moreover, a high occupancy of the particle average for this limb confirms this assumption. In this case, we follow a more conservative criteria thus setting the thresholds to be more discriminative and only modify the state vector $\mathbf{s}_t$ when we have a high confidence of having found the limb. In this way, we avoid oscillations in the state vector. For this module, we set $\sigma_{\theta_l} < 5^{\mathrm{o}}$ and $\rho^{\mathrm{Occ}} > 0.7$ as our working decision point.

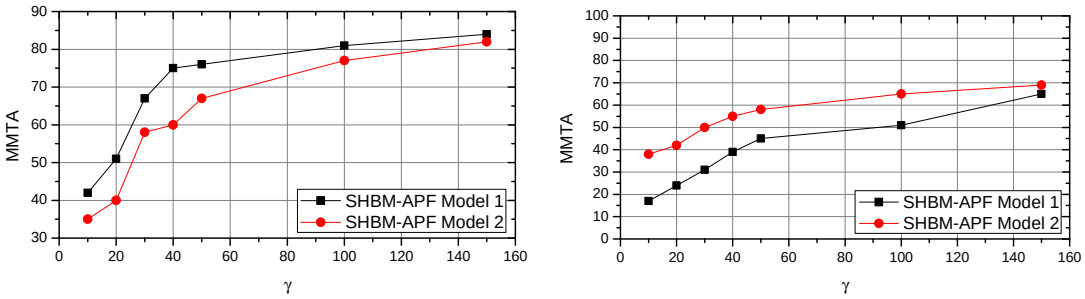


(a) Case 1: Normal operation



(b) Case 2: Lost limbs

Figure 7.12: *DDM-APF operation examples. In both cases, two camera views are displayed containing the follow information at each row: the first four, the four annealing layers of the algorithm, the next two: the final estimated pose and the pose overlaid onto the voxel data.*



(a) Jogging (b) Boxing

Figure 7.13: *Two examples of the SHBM-APF algorithm operation with the two proposed analysis models.*

7.5 Results

7.5.1 SHBM-APF Tracking

The SHBM-APF system is evaluated using HumanEva-I database taking the design considerations already presented in §7.3.2. Particle assignment has been done following Eq.7.21 as:

$$N_{\mathcal{H}_i} = \gamma \cdot K^{\Delta}_{\mathcal{H}_i}, \tag{7.28}$$

thus obtaining:

$$N_{\mathcal{H}_1} = 300, \qquad N_{\mathcal{H}_2} = 500, \qquad N_{\mathcal{H}_3} = 300. \tag{7.29}$$

An exploratory analysis over a fraction of the dataset for the two proposed HBM is depicted in Figure 7.13. Selecting $\gamma = 50$ particles as the working point, adding up to $N_{\text{p}} = 1100$ efficient particles. The internal annealing has been designed to have $L_{\mathcal{H}_i} = 2$ with a variance reduction rate of $\alpha_{\mathcal{H}_i} = 0.5$, $\forall i$. Variance reduction among HBMs, has been set to $\alpha_S = 0.8$. Initial variance values are set in the same way as in all HMC presented systems: to be half of the maximum variation expected in each joint angle. In Figure 7.8(b), we displayed the relative variance reduction associated to every HBM associated variables as they pass through all system steps. Considerations discussed in §6.7.1 have been taken into account hence selecting the minimum voxel resolution, $s_{\mathcal{V}} = 2$ cm, the maximum number of cameras, $N_{\text{C}} = 7$, the partitioned likelihood approach and the model density to $\delta_{\text{Density}} = 1$.

Results for the SHBM-APF algorithm using the two proposals of SHBM are reported in Table 7.1. When comparing the performance of the SHBM-APF using the two aforementioned analysis models, we corroborated that motions where limbs are mostly straight are well captured when using model 1 (Figure 7.7(a)) as in the case of walking or jogging. Activities with a high flexion of limbs such as gesturing, boxing or throwing/catch are better captured using model 2 (Figure 7.7(b)). Both cases are depicted in Figure 7.13 where we noticed that for two specific actions, the most suitable model produces better *MMTA* results than the other although, when employing a large value of γ both tend to converge.

	SHBM-APF Model 1					
	μ	σ	$MMTP$	$MMTA$	μ_θ	σ_θ
Walking	42.11	24.95	39.27	83.19	5.37	2.48
Jog	46.90	26.71	42.62	75.08	7.52	2.98
Throw/Catch	64.22	32.17	51.84	68.11	9.95	3.39
Gesture	53.55	30.81	50.40	71.77	8.26	2.83
Box	58.53	27.96	48.89	72.16	8.91	3.51
Average	**54.06**	**31.71**	**47.46**	**76.85**	**8.02**	**3.03**

	SHBM-APF Model 2					
	μ	σ	$MMTP$	$MMTA$	μ_θ	σ_θ
Walking	43.07	26.12	40.21	82.53	5.24	2.98
Jog	46.51	27.18	43.09	73.85	7.17	3.41
Throw/Catch	58.85	26.79	50.05	72.52	8.74	3.19
Gesture	48.10	29.12	42.91	76.14	6.72	2.58
Box	55.19	26.54	45.31	77.21	5.63	2.80
Average	**51.34**	**28.51**	**45.31**	**76.42**	**6.73**	**2.97**

Table 7.1: *SHBM-APF tracking results on HumanEva-I dataset.* $\epsilon = 100$ *mm.*

Another effect observed in the operation of the SHBM-APF algorithm is its ability to deal with corrupted data. In cases where there is a sudden missing of a part of the data (typically, in the legs part), the simplest model, $\mathcal{H}_0$, is able to keep tracking the torso part regardless of the poor accuracy of the system in the affected limbs. When the data quality is back to normal, the adaptation is much faster than the markerless APF.

Some results showing the SHBM-APF operation are given in Figure 7.14.

7.5.2 DDMA-PF Tracking

A quantitative evaluation of the performance of the DDMA-PF tracking algorithm using the presented metrics is not straightforward. Since the structure of the analysis model depends on the quality of the input data, the obtained result is an output vector with variable length. Although it might be possible to adapt the presented metrics to only quantize the state space variables of the employed final model, its usefulness is limited since it does not allow a direct comparison with the already presented and evaluated methods using HumanEva dataset.

Instead, we opt for a visual comparison of the DDMA-PF versus the APF method, as shown in Figure 7.15. In this case, a person passes near a table and her legs are not well reconstructed. Moreover, the quality of the input data is low, including missing data due to the proximity of the wall and spurious volumes due to a wrong segmentation of the input images. The DDMA-PF is able to detect that the legs part cannot be properly analyzed and the model dissimises them; afterwards, one arm is also affected by missing data, thus being removed from the employed analysis HBM. Finally, when data quality is

back to a tractable level, legs and arms are used. When analyzing the output produced by the APF algorithms, it can be seen that missing data affects the algorithm and awkward poses are produced. Eventually, the algorithm is unable to cope with such changes in the data quality and loses track.

7.6 Conclusions

In this chapter, two algorithms to exploit the underlying hierarchical structure of the human body have been presented: the Scalable Human Body Model Annealead Particle Filter (SHBM-APF) and the Data Driven Model Adaptive Particle Filter (DDMA-PF). These two algorithms can deal with faulty input data and, specifically, with missing data. In the SHBM-APF, a progressive fitting of the HBM is performed thus hierarchicaly exploring the state space and avoiding getting trapped in local minima. In this way, noisy data can be handled more efficiently as proved by the obtained results. On the other hand, when the data exhibits large missing parts, a beforehand selected HBM can be unable to properly analyze it. In this case, the DDMA-PF algorithm adapts the employed analysis model to the quality of the input data by adding or removing limbs and/or their parts to more faithfully explain these data.

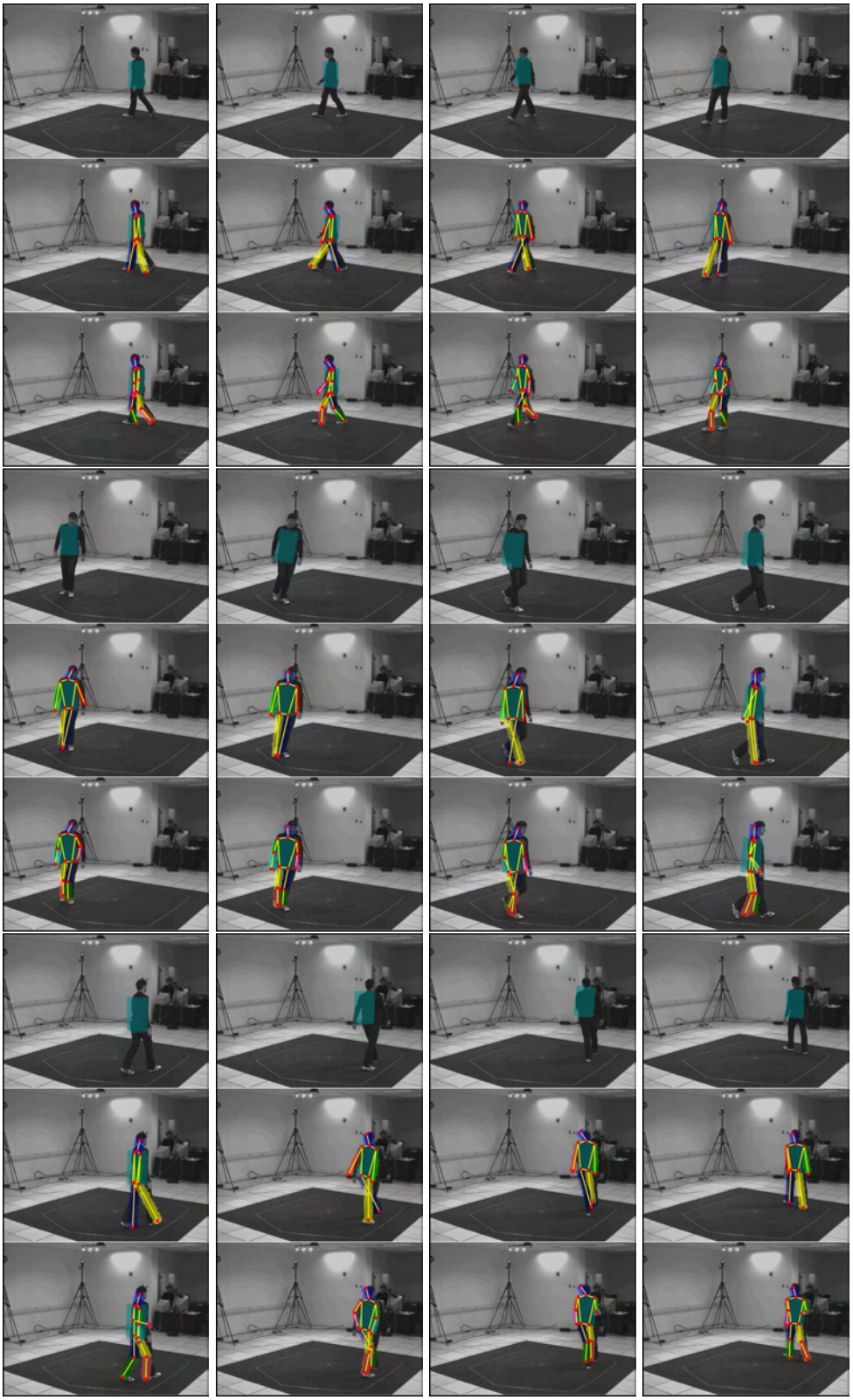


Figure 7.14: *SHBM-APF operation example for action walking. The three involved HBM are stacked for every frame.*

(a) DDMA-PF Tracking

(b) APF Tracking

Figure 7.15: *DDMA-PF vs APF tracking results.*

8

Overall Comparison and Discussion

THIS CHAPTER presents the discussion about the performance of the presented human motion capture (HMC) algorithms when measured within the framework of the HumanEva-I dataset.

8.1 Results comparison

Several HMC algorithms have been proposed along this thesis, ranging from the marker based one to the several proposals for markerless tracking. Results obtained over the HumanEva-I dataset allow a direct comparison of the evaluations performed for all them, summarized in Table 8.1. Some remarks can be given based on these results. When comparing among all methods, it can be seen that the marker based approach outperforms all the other markerless algorithms. This effect is given by the fact that features employed in the marker based APF are far more discriminative. Moreover, likelihood functions associated to marker based APF tend to be sharp and well localized, thus producing accurate results in the *MMTA*. Nonetheless, *MMTP* score is not always below other markerless approaches (SHBM-APF for instance) and this is caused by the data generation procedure introduced in §6.6.1 where a Gaussian noise is added to the position of the markers, driven by a fixed variance. The aim of testing the marker based algorithm under hard operation conditions lead perhaps to an unreal scenario. Therefore, we suppose that, in a real scenario, this algorithm may attain lower *MMTP* values.

When perusing the results of markerless methods, we notice that the APF approach is the one that yields to worst results in comparison with the SHBM-APF alternatives. Although using a human body model (HBM) to generate the 3D volumetric instance of the pose encoded in a given particle to evaluate the likelihood between this particle and the input data, the markerless APF algorithm does not fully exploit the structure of the HBM. Despite the fact that a HBM imposes a number of kinematic restrictions on the angular span of its joints, and this is exploited in the propagation step of the APF, no other knowledge of the HBM is taken into account. Nonetheless, some of the improvements introduced in the APF algorithm such as the partitioned likelihood evaluation (see §6.5.1.2) contribute to give more importance to the end parts of limbs thus implicitly exploiting some HBM hierarchy. This effect is reflected in the obtained results, being the lowest in comparison with the SHBM-APF.

SHBM-APF algorithm is indeed the most efficient markerless approach presented

Method		Walk	Jog	Box	Gesture	Throw/Cach	Average
Marker based APF	*MMTP*	45.81	47.77	46.12	42.42	47.13	45.85
	MMTA	96.15	90.12	87.03	97.46	91.72	95.32
	μ_θ	6.02	7.85	10.55	5.89	9.22	7.09
	σ_θ	2.55	2.75	7.04	2.83	6.17	4.21
Markerless based APF	*MMTP*	72.05	92.21	92.77	90.43	94.69	90.17
	MMTA	79.55	68.24	68.38	69.17	61.30	71.36
	μ_θ	7.97	9.91	9.54	10.96	11.53	10.12
	σ_θ	2.51	3.07	4.10	4.25	3.47	3.33
Model 1 SHBM-APF	*MMTP*	39.27	42.62	48.89	50.40	51.84	47.46
	MMTA	83.19	75.08	72.16	71.77	68.11	76.85
	μ_θ	5.37	7.52	8.91	8.26	9.95	8.02
	σ_θ	2.48	2.98	3.51	2.83	3.39	3.03
Model 2 SHBM-APF	*MMTP*	40.21	43.09	45.31	42.91	50.05	45.31
	MMTA	82.53	73.85	77.21	76.14	72.52	76.42
	μ_θ	5.24	7.17	5.63	6.72	8.74	6.73
	σ_θ	2.98	3.41	2.80	2.58	3.19	2.97

Table 8.1: *Human motion capture results summary from results already presentedin Chapters 6 and 7. Note that, for the sake of readability, μ and σ have been omitted (but available in the aforementioned chapters.*

in this thesis. The progressive exploration of the state space contributed to an efficient and robust HBM fitting. When comparing the performance improvement between SHBM-APF (model 1, for instance) with the markerless APF, we get $\Delta(\mathit{MMTP}, \mathit{MMTA}) = (47.36, 7.69)\%$. Although the increment in the precision, *MMTP*, is notable the improvement of the accuracy, *MMTA*, is fair. This effect is given by the threshold ϵ set in the computation of these two scores, that considers all landmark ground truth-estimation pairs below 100 mm to be matched. If we decrease this threshold to $\epsilon = 70$ mm, we observed $(\mathit{MMTP}, \mathit{MMTA})_{\text{APF}} = (68.19, 54.74)$ and $(\mathit{MMTP}, \mathit{MMTA})_{\text{SHBM-APF}} = (45.22, 71.06)$ yielding to $\Delta(\mathit{MMTP}, \mathit{MMTA}) = (33.68, 22.96)\%$. Hence, if we set our metrics to be more restrictive in the correct pair estimation, we observe how, apart of the improvement in the *MMTP* score, there is a noticeable accuracy improvement when employing the SHBM-APF algorithm.

When comparing the two SHBM-APF algorithms, it can be seen that, depending on the executed action, there is a performance difference. Actions involving straight movement of the limbs, such as walking or jogging, are better captured by model 1 while actions involving a high flexion, such as boxing, gesturing or throw/catch, are better captured by model 2. However, there is not a significant overall performance difference among these two methods. In a real scenario, we might choose model 2, since, although there is no prior information about the motion executed by the performer, it is likely that some gesturing or joint flexion will be involved.

Comparison between the DDMA-PF and the rest of the presented methods is not straightforward. Due to the state space adaptation, *MMTP* and *MMTA* results can not be given for the HumanEva-I. However, the presented data proved the concept of the adaptive model, in comaprison with the non adaptive APF algorithm.

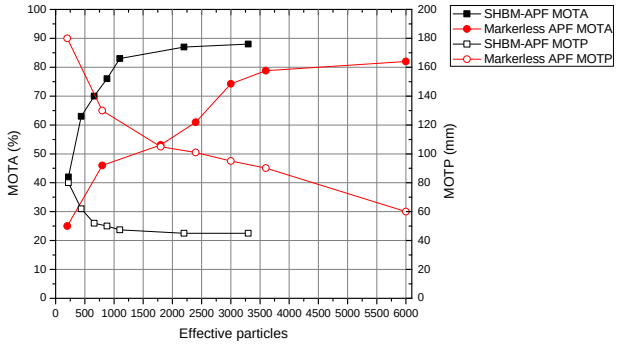


Figure 8.1: *Computational complexity comparison between the markerless APF and the SHBM-APF algorithms.*

When analyzing the performance of the markerless APF and SHBM-APF methods taking into account the complexity of each algorithm, we encountered an interesting result. Let us examine Figure 8.1 where we plot the *MMTP* and *MMTA* scores related with the number of effective particles of both algorithms. It can be seen how, by exploiting the hierarchical structure of the human body, we can obtain better results with a lower computational complexity. For example, for a fixed computational load, $N_{\mathrm{p}} = 2200$, there is a improvement of $\Delta(\mathit{MMTP}, \mathit{MMTA}) = (58.0, 51.2)\%$.

8.2 State of the art comparison

A number of algorithms in the literature have been evaluated using HumanEva-I and their results have been reported in Table 8.2. However, some algorithms presented results only using the HumanEva-II database, which is significantly smaller (only 2 sequences) and involves a very reduced set of motions (walking and jogging). In this thesis, we have not dealt with this dataset since no ground truth information was provided to compute *MMTP* and *MMTA* and, furthermore, the data contained in HumanEva-I was far more challenging in terms of types of motion. Nonetheless, and only for qualitative comparison, those state of the art methods that reported results based on the HumanEva-II dataset have been reported in Table 8.2.

Among the studied state of the art methods we find two main trends those methods based on a tracking formulation of the problem and methods based on statistical classification. The methods presented within this thesis fall into the first category where some comparisons can be made. Among the reported methods, we find the expectation-maximization (EM) kinematically constrained GMM method presented by Cheng and Trivedi [CT07] as the continuation of the techniques already presented by Mikič [Mik03]. Addressing a complex problem such as human motion capture using EM is perhaps manageable in a benevolent scenario with well learnt constrains but, as suggested by Caillete *et al.* [CH04, CGH05] in his comparison of EM and PF based methods, Monte Carlo based techniques clearly outperform those based in minimization algorithms (nonetheless, some efforts have been made in tackling HMC in this fashion [KG06]). Other con-

Method	Walk	Jog	Box	Gesture	Throw/Cach	Average
EM+Kinematically constrained GMM [CT07]	–	–	–	–	–	150.9
Hierarchical Partitioned PF [HW07]	101.9	–	–	–	–	–
PF+Dynamic model [BFH06]	100.4	–	–	–	–	–
ICP+Naïve classification [MCA06]	53.1	–	45.4	–	–	–
Example-based pose estimation [Pop07a]	45.3	43.8	94.3	–	–	–
Example-based pose estimation [OS08]	–	–	–	–	–	37.98
Sparse probabilistic regression [UD08]	32.7	31.2	38.5	–	–	
Marker based APF	56.01	62.51	77.89	44.70	58.31	59.88
Markerless APF	96.52	130.34	145.22	124.87	122.27	121.18
SHBM-APF (Model 1)	42.11	49.90	58.53	53.55	64.22	54.06
SHBM-APF (Model 2)	43.07	46.51	55.19	48.10	58.85	51.34

Table 8.2: *Result comparisons with state of the art methods evaluated over HumanEva dataset. The presented score corresponds to the mean of the error estimation μ, as reported by the compared authors in their respective contributions.*

tributions reported over HumanEva-I are based on the seminal idea of PF. Husz *et al.* [HW07] included a particle propagation step relying on learnt information on the structure of the executed motion thus facing the already mentioned problem of lack of adaptivity to unseen motions. A very detailed dynamic model of the human kinematics is employed by Brubaker *et al.* [BFH06]. Although these two methods report results comparable to our markerless APF algorithm, they only evaluate sequences where motion can be well modeled (both by learning or using a dynamic model) such as walking. Motion involving a more complex pattern such as boxing or gesturing may not cope well with these two methods.

The other family of human motion capture algorithms is based on learning and classification instead of tracking. Basically, these techniques examine the ground truth data and extract a number of features from them. Afterwards, when a new test frame is processed, these same features are extracted and the best match between them and the already learnt ones is outputed. In other words, human motion capture is no longer posed as a tracking problem but as a pattern recognition one. Results obtained with these techniques, specially those of Urtasun *et al.* [UD08] and Poppe *et al.* [Pop07a], outperform the tracking based ones. However, these techniques are constrained to track a beforehand selected action and their applicability to unknown motion patterns is limited. It is notable the technique presented by Munderman *et al.* [MCA06] where a 3D reconstruction is performed before computing the features to be learnt, that is using a data fusion approach in the same fashion as this thesis.

9

Conclusions, Contributions and Perspectives

IN ORDER to conclude this thesis, a review of the contributions is presented. Indeed, the objectives targeted in our research plan have been fulfilled and a number of algorithms are presented for the task of human motion capture. However, a large amount of perspectives, future research lines and derived applications are still to be addressed.

9.1 Contributions

In this thesis, a number of contributions to the task of person tracking and human motion capture into the context of a multi-camera setup have been presented. The achievements of this thesis are listed in the following.

9.1.1 Contributions to multi-person/multi-camera tracking

- **Voxel based analysis.** Multi-person tracking in the context of multi-camera image processing has been posed as the previous step to human motion capture in Chapter 4. We opted for a data fusion strategy previous to any analysis process in contraposition with other approaches based on a per-camera analysis and a feature fusion afterwards. This data fusion was performed by aggregating information from all camera views into a single and unified representation of the 3D space by means of colored voxels. We investigated the impact of the creation and deletion of tracks into the performance scores and proposed a Bayesian approach to this creation/deletion process based on extracting a number of object features and deciding whether an object is a person or not by means of a binary decision tree classifier.

- **Sparse Sampling filtering.** The filtering of the state of a track (that is, its centroid) using the colored 3D voxel data was achieved by means of two proposals: particle filtering (PF) and sparse sampling (SS). While PF was found to produce fair results it turned out to be computationally expensive and experienced some problems when managing mergings between the tracked blob (a person) and other spurious blobs. SS, being perhaps the most notable contribution of that chapter, is proposed as a low complexity solution able to cope with the mergings problem and being robust to noisy data as well. Experiments conducted over the CLEAR 2007 database assessed the performance in both accuracy/precision and computational load of both algorithms.

- **Publications.** The following publications have been produced related with this topic: [ACFS$^+$06, LCFC07, CFSC07a, CFSCP08, CFSC$^+$08, CFCPM09a, NPS$^+$09].

9.1.2 Contributions to human body motion tracking

- **Marker-based HMC.** First, a marker based approach together with an annealed particle filter (APF) has been introduced in Chapter 6 as an economic alternative to available commercial systems. On the other hand, markerless HMC has deserved most of the novelties presented in this work. Input data was set to be a 3D reconstruction of the analyzed space following a data fusion approach as the starting point to all algorithms.
- **Markerless HMC.** Markerless APF has been presented in Chapter 6 as a first approach to markerless HMC with several contributions to the design of the APF such as an efficient likelihood formulation and a hard kinematic restriction in the particle propagation step.
- **Scalable HBM markerless HMC.** Analyzing heavily corrupted input data towards extracting the HBM pose has been addressed in Chapter 7 where two systems have been presented. First, the SHBM-APF exploits the hierarchical and scalable structure of the HBM into a double annealing loop: structure and likelihood function. This strategy, being the main contribution of this thesis, allows an efficient and robust data processing with an affordable complexity. In the case where parts of data are missing, we proposed a model adaptive system, the DDM-APF.
- **Evaluation methods.** Finally, all these methods are evaluated through a set of new metrics presented in Chapter 5, designed in such a way that they do not present the bias present in standard HMC performance metrics.
- **Publications.** Several publications have been provided in this field: [CFCP05b, CFCTP05, CFCP06b, CFCP$^+$06c, CFCPM09b, CFCP09a, CFCP09b, CFCP09c].

9.1.3 Side Contributions

During the execution period of this thesis, several side research topics have been covered due to requirements of some of the several projects the Image Processing Group at UPC was involved with. Research performed within the framework of a Smart Room yield to the following contributions:

- **Multi-camera head orientation estimation:** [CFCP05a, CFCP06a, VGCF$^+$09].
- **Multimodal head orientation estimation:** [SCFCH07, CFCP07, CFSC$^+$07b].
- **Focus of attention analysis:** [CFSC$^+$08].
- **Multimodal human motion analysis:** [OCFT$^+$08b, ODCF$^+$08, OCFT$^+$08a].
- **Multimodal acoustic event classification:** [BTNCF08b, BTNCF08a, CFBS$^+$09, BCFS$^+$09].

Parts of the contributions and investigations conducted in this dissertation have been undertaken in answer to the challenges raised by some of the projects where the Image Processing Group of the UPC has been involved. In particular, this work has been supported by the EU through the Integrated Project CHIL (Computers in the Human Interaction Loop) and by the Networks of Excellence SIMILAR (Human-machine interfaces SIMILAR to human-human communication) and MUSCLE (Multimedia Understanding Through Semantics, Computation and Learning). In addition, this work has also been developed within the framework of the Spanish projects TEC2004-01914 (Analysis, coding and semantic indexing in controlled environments), HESPERIA (Homeland sEcurity: tecnologíaS Para la sEguridad integRal en espacIos públicos e infrAestructuras) and Vision (Comunicaciones de Vídeo de Nueva Generación).

9.2 Future work

Despite the work presented in this thesis is usable and potentially useful for real applications, there is a number of research lines that unfold after it. Future research directions may be summarized as follows:

- Once robust techniques for HMC are available, the obtained output can be analyzed for a number of applications. For instance, action recognition can be addressed based on the temporal evaluation of HBM defining parameters. Gait can be also recognized based on this information yielding to biometric and person recognition applications.

- Combining the Scalable Human Body Model-Annealed Particle Filter (SHBM-APF) together with the Data Driven Adaptive Particle Filter (DDA-PF) may be a future research line. This combination may lead to more versatile filtering schemes when analyzing noisy data with occlusions. It must be said that, during this Ph.D. thesis, we have not encountered situations so hard that required further adaptive filtering strategies. However, for more detailed HBM, this SHBM-APF/DDA-PF combination may be of interest.

- Multi-person tracking with multiple sensors is a topic where we have already produced some results employing audio-visual inputs. Fusing this information within a PF framework has been well researched, but the fusion of audio-visual information using SS is yet to be explored.

- Problems (not necessarily in the image processing field) involving a structure with some hierarchical properties can benefit from the considerations and algorithms introduced in Chapter 7 for robust human motion capture. For instance, multi-resolution shape analysis and tracking in the domain of oceanography [MEM+08] falls in this category leading to a possible research path.

- Human motion capture is indeed a fertile research field due to the number of challenges derived from fitting a structured model with a high number of defining parameters to noisy data. In Chapter 6 we addressed this problem using what we

though to be the most appropriate technique: the annealed particle filter. However, there is still work to do towards automatic adjustment of the involved operation parameters such as the annealing scheduling or the propagation noise (without restricting ourselves to a beforehand learnt motion pattern). Recent advances on Monte Carlo techniques [Vas08] are still to be applied to the field of human motion capture.

- Automatically establishing the size of the body parts in a HBM for tracking purposes is still an open problem. Although there might be a quasi-linear dependency of the limb lengths with the height of a person, no relation can be derived regarding its perimeter. Some researchers solve this problem by means of an initialization protocol where subjects must adopt a pre-definite pose before the system starts the tracking process. Some efforts have been presented by [Mik03] using a Bayesian network to estimate such length and size dimensions.

- The markerless approach to human motion capture presented in this thesis has been mainly focused on using a voxel reconstruction as the input of the algorithms. Nonetheless, other input data have to be explored together with the scalable filtering proposals and, perhaps, combinations of several modalities. Among them, we count image measurements [DR05] or depth information derived from stereo vision [ZNS06].

- Real-time implementation of some algorithms presented in this thesis can be addressed in the near future using already available hardware such as GPU's.

- This thesis has been written from an engineering point of view, perhaps lacking of mathematical rigour in some sections where clarity and practical examples were preferred to formalisms and in-depth demonstrations. Recently, the standard and annealed particle filter convergence was analyzed in [GPS$^+$07] providing the mathematical insight that these methods required. Assessing the mathematical inners of sparse sampling algorithm from Chapter 4 and the filtering strategies based on scalable human body models in Chapter 7 with mathematical eye is an open issue.

A
Exponential Maps

Exponential maps are an efficient and singularity-free way to encode rotations and translations. This representation has been used in our research and some further details are provided in this appendix. Let us consider Figure A.1 where $\omega \in \mathbb{R}^3$ is the axis of rotation and $\mathbf{q} \in \mathbb{R}^3$ is the center of rotation of a rigid solid. Assuming that the object rotates with unit velocity, the velocity of a point $\mathbf{p}(\theta)$ on the object is given by:

$$\dot{\mathbf{p}}(\theta) = \omega \times (\mathbf{p}(\theta) - q)\,. \tag{A.1}$$

In homogeneous coordinates, this expression can be written as:

$$\begin{bmatrix} \dot{\mathbf{p}} \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{\omega} & -\omega \times \mathbf{q} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \tag{A.2}$$

$$\dot{\bar{\mathbf{p}}} = \hat{\xi}\bar{\mathbf{p}}, \tag{A.3}$$

where $\bar{\mathbf{p}}$ is the homogeneous representation of point $\mathbf{p}$, and $\hat{\omega}$ is the skew symmetric matric such that $\omega \times \mathbf{q} = \hat{\omega}\mathbf{q}$, $\forall \mathbf{q} \in \mathbb{R}^3$:

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_z & -\omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{A.4}$$

and

$$\hat{\xi} = \begin{bmatrix} \hat{\omega} & -\omega \times \mathbf{q} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \hat{\omega} & \mathbf{v} \\ 0 & 0 \end{bmatrix} \tag{A.5}$$

is defined as the associated twist with the rotation about the axis defined by ω and $\mathbf{q}$. Then, the solution to the differential equation posed in Eq.A.1 is:

$$\bar{\mathbf{p}}(\theta) = e^{\hat{\xi}t}\bar{\mathbf{p}}(\theta_0), \tag{A.6}$$

where $e^{\hat{\xi}\theta}$ is the exponential map associated with the twist $\hat{\xi}$. Despite this exponential notation, $e^{\hat{\xi}\theta}$ is a matrix with a number of properties[1]. This operator maps the initial location $\mathbf{p}(0)$ to its new location, $\mathbf{p}(t)$, after rotation θ radians about the axis defined by ω and $\mathbf{q}$. It can be proved that:

$$e^{\hat{\xi}\theta} = \begin{bmatrix} e^{\hat{\omega}\theta} & (\mathbf{I} - e^{\hat{\omega}\theta})(\omega \times \mathbf{v}) + \omega\omega^T\mathbf{v}\theta \\ 0 & 1 \end{bmatrix}, \tag{A.7}$$

[1] For more details about the properties of the exponential map, check [MSZ94, Gra98, HZ04].

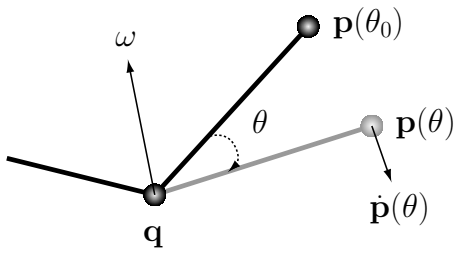


Figure A.1: *Rotation and translation scheme.*

where $e^{\hat{\omega}\theta}$ is the rotation matrix associated with the rotation of θ radians about an axis ω. The closed form of this matrix can be obtained through Rodrigues' formula [HZ04]:

$$e^{\hat{\omega}\theta} = \mathbf{I} + \frac{\hat{\omega}}{\|\omega\|}\sin(\|\omega\|\theta) + \frac{\hat{\omega}^2}{\|\omega\|^2}\left(1 - \cos(\|\omega\|\theta)\right). \tag{A.8}$$

Exponential map $e^{\hat{\xi}\theta}$ can be alternatively written in the form:

$$e^{\hat{\xi}\theta} = \begin{bmatrix} \mathbf{R}(\theta) & \mathbf{t}(\theta) \\ 0 & 1 \end{bmatrix}, \tag{A.9}$$

where $\mathbf{R}(\theta)$ corresponds to a rotation of θ radians on the rotation axis and $\mathbf{t}(\theta)$ is the translation associated with the distance between the rotation center $\mathbf{q}$ and the studied point. In some domains, this mapping is denoted as the roto-translation matrix [CPF03].

The main advantage of using exponential maps is the simplicity when concatenating rotations and translations, by multiplying the associated exponential maps. Moreover, it can be shown[MSZ94] that the result is independent of the order of the products. For an open kinematic chain containing K axes of rotation, the transformation between the base of the chain and the last point on the last link of the chain, is given by:

$$\Lambda_1^K = e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}\cdots e^{\hat{\xi}_K\theta_K} = \prod_{k=1}^{K} e^{\hat{\xi}_k\theta_k}. \tag{A.10}$$

Given a point $\mathbf{p}(\theta_0) = (x, y, z)$, its coordinates after applying a rotation in a given axis can be derived. Indeed, expressions of the exponential map for each rotation axis can be

analitically derived from Eq.A.7 and A.8:

$$\text{Axis } x \quad e^{\hat{\xi}_x \theta_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & z\sin\theta_x + y(1-\cos\theta_x) \\ 0 & \sin\theta_x & \cos\theta_x & -y\sin\theta_x + z(1-\cos\theta_x) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.11}$$

$$\text{Axis } y \quad e^{\hat{\xi}_y \theta_y} = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & -z\sin\theta_y + x(1-\cos\theta_y) \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & x\sin\theta_y + z(1-\cos\theta_y) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.12}$$

$$\text{Axis } z \quad e^{\hat{\xi}_z \theta_z} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & y\sin\theta_z + x(1-\cos\theta_z) \\ \sin\theta_z & \cos\theta_z & 0 & -x\sin\theta_z + y(1-\cos\theta_z) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.13}$$

We can obtain the coordinates of a point $\bar{\mathbf{p}}(K)$ at the end of the last link, knowing the position of the point at the origin of the first link $\bar{\mathbf{p}}(0)$ and the lenght, axis of rotation and angle of each link as:

$$\bar{\mathbf{p}}(K) = \Lambda \bar{\mathbf{p}}(0). \tag{A.14}$$

B

Discrete Rotation Considerations

Proposition: Let us consider compact region $\mathcal{D}$ described by a set of positions $\mathbf{x}_j = (x_j, y_j) \in \mathbb{R}^2$ encoded into the matrix $A = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_n]^\top$. Consider the rotation operation of region $\mathcal{D}$ described as $A_\theta = [\mathbf{x}_{\theta,1}\ \mathbf{x}_{\theta,2}\ \cdots\ \mathbf{x}_{\theta,n}]^\top = AR_\theta$, where R_θ is a rotation transformation matrix in $\mathbb{R}^2$. If we try to map A_θ onto a discrete grid (i.e. an image), we must discretize A_θ by mapping every position $\mathbf{x}_{\theta,j} = (x_{\theta,j}, y_{\theta,j})$ onto this discrete grid as:

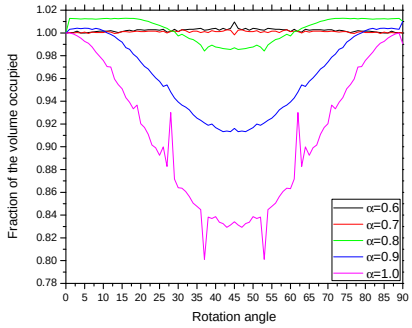
$$\tilde{\mathbf{x}}_{\theta,j} = (\lfloor x_{\theta,j} \rfloor, \lfloor y_{\theta,j} \rfloor)\,. \tag{B.1}$$

Taking into account the described operation, the set $\mathcal{D}_\theta$ is not compact.

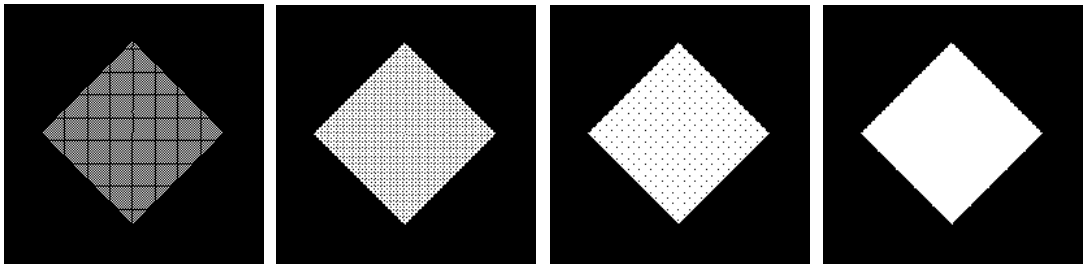
Demonstration: Although this problem is related with sampling theory and a formal demonstration can be derived, we devised an empirical method to prove the validity of the proposition and give a more visual effect of its implications. We defined a compact region (a square) onto an image and computed the fraction of area of this region that disappear w.r.t. to the initial area, when applying a given rotation. Results are depicted in Figure B.1(a) (curve with $\alpha = 1$), where it can be seen that for the critical angle $\theta = 45^\text{o}$, the number of pixels decreased to the 84% of the initial area. The visual example is shown in Figure B.1(c).

Compactness of region $\mathcal{D}_\theta$ can be achieved by enlarging the number of elements in matrix A with non-unitary increments among $\mathbf{x}_j$, $\forall j$. That is, to uniformly over-sample the initial region $\mathcal{D}$ by reducing the pixel size as $\tilde{s}_p = \alpha s_p$. Hence, after a rotation, when applying Eq.B.1, these extra coordinates will map the empty regions that broke the compactness property of $\mathcal{D}_\theta$. However, as a consequence, some $\tilde{\mathbf{x}}_j$ will map onto the same pixel thus decreasing the computational performance of the algorithm.

By exploring the influence of α onto the compactness of $\mathcal{D}_\theta$ we obtain the results shown in Figure B.1. The approximate optimal value is $\alpha = 0.7$. Once we have stated that the direction that presents the minimal compactness is $\theta = 45^\text{o}$, we can alternatively estimate the optimal α value by means of trigonometric considerations leading to $\alpha = 1/\sqrt{2} \approx 0.7$.



(a)



(b) $\alpha = 1.5$ (c) $\alpha = 1.0$ (d) $\alpha = 0.8$ (e) $\alpha = 0.7$

Figure B.1: *Rotation considerations of an object on a discrete grid. In (a), the influence of the over-sampling parameter α on the compactness of the region $\mathcal{D}$. In (b)-(e), examples of compactness of a region when applying the critical rotation $\theta = 45^o$, for different values of α.*

Bibliography

[ACFS+06] A. Abad, C. Canton-Ferrer, C. Segura, J.L. Landabaso, D. Macho, J.R. Casas, J. Hernando, M. Pardàs, and C. Nadeu. UPC audio, video and multimodal person tracking systems in the CLEAR evaluation campaign. In *Proceedings of Classification of Events, Activities and Relationships Evaluation and Workshop*, volume 4122 of *Lecture Notes on Computer Science*, pages 93–104, 2006. 29, 140

[AMGC02] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb 2002. 15, 17, 19

[AT04] A. Agarwal and B. Triggs. 3D human pose from silhouettes by relevance vector regression. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 9–16, 2004. 67, 72, 78

[ATS06] E. Aguiar, C. Theobalt, and H. Seidel. Automatic learning of articulated skeletons from 3D marker trajectories. In *Proceedings of 2nd International Symposium on Advances in Visual Computing*, volume 4291 of *Lecture Notes on Computer Science*, pages 485–494, 2006. 86

[BAHH92] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of European Conference on Computer Vision*, volume 588 of *Lecture Notes on Computer Science*, pages 237–252, 1992. 112

[BB08] A.O. Bălan and M.J. Black. The naked truth: estimating body shape under clothing. In *Proceedings of Euroepan Conference on Computer Vision*, volume 2, pages 15–29, 2008. 110

[BCFS+09] T. Butko, C. Canton-Ferrer, C. Segura, X. Giró, C. Nadeu, J. Hernando, and J.R. Casas. Improving detection of acoustic events using audiovisual data and feature level fusion. In *Proceedings of Interspeech*, 2009. 140

[BD01] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001. 74, 76

[BD02] E. Borovikov and L. Davis. 3D shape estimation based on density driven model fitting. In *Proceedings of 1st International Symposium on 3D Data Processing Visualization and Transmission*, pages 116–125, 2002. 90

[BER02] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. In *Proceedings of Workshop on Motion and Video Computing*, pages 169–174, 2002. 30

[BES06] K. Bernardin, A. Elbs, and R. Stiefelhagen. Multiple object tracking performance metrics and evaluation in a smart room environment. In *Proceedings of IEEE International Workshop on Visual Surveillance*, 2006. 30, 47, 63

[BFH06] M. Brubaker, D.J. Fleet, and A. Hertzmann. Physics-based human pose tracking. In *Proceedings of Workshop on Evaluation of Articulated Human Motion and Pose Estimation*, 2006. 138

[BFOS93] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Chapman and Hall, 1993. 36

[BGS07] K. Bernardin, T. Gehrig, and R. Stiefelhagen. Multi-level particle filter fusion of features and cues for audio-visual person tracking. In *Proceedings of Classification of Events, Activities and Relationships Evaluation and Workshop*, volume 4625 of *Lecture Notes on Computer Science*, pages 70–81, 2007. 32, 51, 52, 126

[BHP05] N.V. Boulgoris, D. Hatzinakos, and K.N. Plataniotis. Gait recognition: a challenging signal processing technology for biometric identification. *IEEE Signal Processing Magazine*, 22(6):78–90, 2005. 77

[BK01] C. Barron and I.A. Kakadiaris. Estimating anthropometry and pose from a single uncalibrated image. *Computer Vision and Image Understanding*, 81(3):269–284, 2001. 72

[BM98] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, number 1, pages 8–15, 1998. 81

[BMP04] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, 2004. 78, 81

[Bou04] J.Y Bouget. Camera calibration toolbox for Matlab. http://www.vision.caltech.edu/bouguetj, 2004. 10, 60

[Bre65] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965. 13

[BSB05] A.O. Bălan, L. Sigal, and M.J. Black. A quantitative evaluation of video-based 3D person tracking. In *Proceedings of 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 349–356, 2005. 20, 96, 110

[BTNCF08a] T. Butko, A. Temko, C. Nadeu, and C. Canton-Ferrer. Fusion of audio and video modalities for detection of acoustic events. In *Proceedings of Interspeech*, 2008. 29, 58, 140

[BTNCF08b] T. Butko, A. Temko, C. Nadeu, and C. Canton-Ferrer. Inclusion of video information for detection of acoustic events using fuzzy integral. In *Proceedings of 5nd Joint Workshop on Machine Learning and Multimodal Interaction*, volume 5237 of *Lecture Notes on Computer Science*, pages 74–85, 2008. 29, 58, 140

[Bŏ3] A.O. Bălan. Voxel carving and coloring - constructing a 3D model of an object from 2D images. 2003. 12

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 104

[CCU+05] P. Correa, J. Czyz, T. Umeda, F. Marqués, X. Marichal, and B. Macq. Silhouette-based probabilistic 2D human motion estimation for real-time applications. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 836–839, 2005. 76

[CFBS+09] C. Canton-Ferrer, T. Butko, C. Segura, X. Giró, C. Nadeu, J. Hernando, and J.R. Casas. Audiovisual event detection towards scene understanding. In *Proceedings of Workshop on Human Communicative Behavior Analysis within the IEEE International Conference on Computer Vision and Pattern Recognition*, 2009. 140

[CFCP05a] C. Canton-Ferrer, J. R. Casas, and M. Pardàs. Fusion of multiple viewpoint information towards 3D face robust orientation detection. In *Proceedings of IEEE International Conference on Image Processing*, volume 2, pages 366–369, 2005. 140

[CFCP05b] C. Canton-Ferrer, J. R. Casas, and M. Pardàs. Towards a Bayesian approach to robust finding correspondences in multiple view geometry environments. In *Proceedings of 4th International Workshop on Computer Graphics and Geometric Modelling*, volume 3515 of *Lecture Notes on Computer Science*, pages 281–289, 2005. 29, 30, 71, 75, 87, 88, 140

[CFCP06a] C. Canton-Ferrer, J. R. Casas, and M. Pardàs. Head pose detection based on fusion of multiple viewpoint information. In *Proceedings of Classification of Events, Activities and Relationships Evaluation and Workshop*, volume 4122 of *Lecture Notes on Computer Science*, pages 305–310, 2006. 140

[CFCP06b] C. Canton-Ferrer, J.R. Casas, and M. Pardàs. Human model and motion based 3D action recognition in multiple view scenarios. In *Proceedings of European Signal Processing Conference*, 2006. 71, 75, 76, 80, 140

[CFCP+06c] C. Canton-Ferrer, J.R. Casas, M. Pardàs, M.E. Sargin, and A.M. Tekalp. 3D human action recognition in multiple view scenarios. In *Proceedings of 2es Jornades de Recerca en Automàtica, Visió i Robòtica*, pages 134–138, 2006. 71, 76, 140

[CFCP07] C. Canton-Ferrer, J. R. Casas, and M. Pardàs. Head orientation estimation using particle filtering in multiview scenarios. In *Proceedings of Classification of Events, Activities and Relationships Evaluation and Workshop*, volume 4625 of *Lecture Notes on Computer Science*, pages 317–327, 2007. 140

[CFCP08] C. Canton-Ferrer, J.R. Casas, and M. Pardàs. Exploiting structural hierarchy in articulated objects towards robust motion capture. In *Proceedings of 5th Conference on Articulated Motion and Deformable Objects*, volume 5098 of *Lecture Notes on Computer Science*, pages 82–91, 2008. 73, 110

[CFCP09a] C. Canton-Ferrer, J. R. Casas, and M. Pardàs. Towards a low cost multi-camera marker based human motion capture system. In *Proceedings of IEEE International Conference on Image Processing (submitted)*, pages 2644–2647, 2009. 71, 73, 75, 140

[CFCP09b] C. Canton-Ferrer, J. R. Casas, and M. Pardàs. Voxel based annealed particle filtering for markerless 3d articulated motion capture. In *Proceedings of IEEE Conference on 3DTV*, pages 2644–2647, 2009. 71, 140

[CFCP09c] C. Canton-Ferrer, J.R. Casas, and M. Pardàs. Monte carlo marker-based real-time robust human motion capture in multi-camera environments (submitted). *Image and Vision Computing*, 2009. 71, 140

[CFCPM09a] C. Canton-Ferrer, J.R. Casas, M. Pardàs, and E. Monte. Multi-camera multi-person voxel based Monte Carlo 3D tracking strategies. *Computer Vision and Image Understanding (submitted)*, 2009. 29, 140

[CFCPM09b] C. Canton-Ferrer, J.R. Casas, M. Pardàs, and E. Monte. Towards a fair evaluation of video-based 3d human pose estimation algorithms (submitted). In *Proceedings of IEEE International Conference on Computer Vision*, 2009. 59, 140

[CFCTP05] C. Canton-Ferrer, J.R. Casas, A.M. Tekalp, and M. Pardàs. Projective Kalman filter: multiocular tracking of 3D locations towards scene understanding. In *Proceedings of 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, volume 3869 of *Lecture Notes on Computer Science*, pages 250–261, 2005. 71, 87, 140

[CFSC07a] C. Canton-Ferrer, J. Salvador, and J.R. Casas. Multi-person tracking strategies based on voxel analysis. In *Proceedings of Classification of Events, Activities and Relationships Evaluation and Workshop*, volume 4625 of *Lecture Notes on Computer Science*, pages 91–103, 2007. 29, 30, 52, 140

[CFSC$^+$07b] C. Canton-Ferrer, C. Segura, J.R. Casas, M. Pardàs, and J. Hernando. Audiovisual head orientation estimation with particle filters in multisensor scenarions. *EURASIP Journal on Advances in Signal Processing*, 2007. 15, 140

[CFSC+08] C. Canton-Ferrer, C. Segura, J.R. Casas, M. Pardàs, and J. Hernando. Multimodal real-time focus of attention estimation in smartrooms. In *Proceedings of Workshop on Human Communicative Behavior Analysis within the IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1–8, 2008. 15, 29, 30, 140

[CFSCP08] C. Canton-Ferrer, R. Sblendido, J. R. Casas, and M. Pardàs. Particle filtering and sparse sampling for multi-person 3D tracking. In *Proceedings of IEEE International Conference on Image Processing*, pages 2644–2647, 2008. 29, 30, 75, 140

[CGH05] F. Caillette, A. Galata, and T. Howard. Real-time 3D human body tracking using variable length Markov models. In *Proceedings of British Machine Vision Conference*, volume 1, pages 469–478, 2005. 20, 21, 73, 76, 79, 80, 83, 110, 137

[CGPV05] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani. Probabilistic posture classification for human-behavior analysis. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 35(1):42–54, 2005. 1, 77

[CH04] F. Caillette and T. Howard. Real-time markerless human body tracking with multi-view 3-D voxel reconstruction. In *Proceedings of British Machine Vision Conference*, volume 2, pages 597–606, 2004. 137

[Che03] G. Cheung. *Visual hull construction, alignment and refinement for human kinematic modeling, motion tracking and rendering*. PhD thesis, Carnegie Mellon, 2003. 11, 12, 76, 93

[Che05] Z. Chen. Bayesian filtering: From Kalman filters to particle filters and beyond. Technical report, McMaster University, 2005. 17, 18

[CHI07] CHIL - Computers in the Human Interaction Loop, fp-6 european integrated project. http://chil.server.de, 2004-2007. 1, 11, 30, 47, 54, 77

[CKBH00] G.K.M. Cheung, T. Kanade, J.Y. Bouguet, and M. Holler. A real time system for robust 3D voxel reconstruction of human motions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 714–720, 2000. 5, 12, 30, 31

[CLE07] CLEAR - Classification of Events, Activities and Relationships Evaluation and Workshop. http://www.clear-evaluation.org, 2007. 2, 7, 29, 46, 47, 52, 59, 60, 64

[CM98] J.J. Crisco and R.D. McGovern. Efficient calculation of mass moments of intertia for segmented homogeneous three-dimensional objects. *Journal of Biomechanics*, 31(1):97–101, 1998. 41

[Con80] W.J. Conover. *Practical Nonparametric Statistics*. 1980. 67

[Cox93] I.J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal on Computer Vision*, 10(1):53–66, 1993. 32

[CPF03] P. Cerveri, A. Pedotti, and G. Ferrigno. Robust recovery of human motion from video using Kalman filters and virtual humans. *Human Movement Science*, 22:377–404, 2003. 1, 5, 75, 77, 81, 86, 144

[CS06] J.R. Casas and J. Salvador. Image-based multi-view scene analysis using conexels. In *Proceedings of HCSNet workshop on use of vision in human-computer interaction*, pages 19–28, 2006. 11, 31

[CT07] S. Cheng and M. Trivedi. Articulated body pose estimation from voxel reconstructions using kinematically constrained Gaussian mixture models: algorithm and evaluation. In *Proceedings of 2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation*, 2007. 137, 138

[DBT03] S.L. Dockstader, M.J. Berg, and A.M. Tekalp. Stochastic kinematic modeling and feature extraction for gait analysis. *IEEE Transactions on Image Processing*, 12(8):962–976, 2003. 1, 76, 77, 82, 110

[DF99] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with silhouettes. In *Proceeding of International Conference on Computer Vision*, volume 2, pages 716–721, 1999. 74, 90

[DF01] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with physical forces. *Computer Vision and Image Understanding*, 81(2):328–357, 2001. 5, 90

[DFG01] A. Doucet, N. Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001. 17, 18, 78

[DHS00] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2000. 36

[DKZ+03] P.M. Djuric, J.H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, and J. Miguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, 2003. 15

[DM06] L. Ding and A.M. Martinez. Three-dimensional shape and motion reconstruction for the analysis of American sign language. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 146–152, 2006. 1

[DR05] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, 2005. 5, 20, 21, 23, 26, 61, 74, 78, 92, 93, 94, 114, 142

[FJ06] T. Foures and P. Joly. Scalability in human shape analysis. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 2109–2112, 2006. 112

[FK02] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. In *Proceedings of 5nd IEEE International Summer School on Biomedical Imaging*, 2002. 11, 31

[FL01] O. Faugeras and Q.T. Luong. *The geometry of multiple views.* MIT Press, 1st edition, 2001. 7

[FLD08] M. Fontmarty, F. Lerasle, and P. Danès. Towards real-time markerless human motion capture from ambiance cameras using and hybrid particle filter. In *Proceedings of IEEE International Conference on Image Processing*, pages 709–712, 2008. 75, 94

[FS02] D. Focken and R. Stiefelhagen. Towards vision-based 3-D people tracking in a Smart Room. In *Proceedings of IEEE International Conference on Multimodal Interfaces*, pages 400–405, 2002. 30

[Gar04] O. Garcia. Mapping 2D images and 3D world objects in a multicamera system. Master's thesis, Image Processing Department, Technical University of Catalonia, 2004. 10

[GBC02] V. Girondel, L. Bonnaud, and A. Caplier. Hands detection and tracking for interactive multimedia applications. In *Proceedings of International Conference on Computer Vision and Graphics*, 2002. 74

[GD96] D.M. Gavrila and L.S. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 73–80, 1996. 5, 76, 89, 93

[GF05] G. Guerra-Filho. Optical motion capture: theory and implementation. *Journal of Theoretical and Applied Informatics*, 12(2):61–89, 2005. 86

[GG04] R.D. Green and L. Guan. Quantifying and recognizing human movement patterns from monocular video images-Part I: a new framework for modeling human motion. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(2):179–190, 2004. 72

[GGTS01] N. Grammalidis, G. Goussis, G. Troufakos, and M.G. Strintzis. 3-D human body tracking from depth images using analysis by synthesis. In *Proceedings of IEEE International Conference on Image Processing*, volume 2, pages 185–188, 2001. 73

[GMD08] A. Gupta, A. Mittal, and L.S. Davis. Constrain integration for efficient multiview pose estimation with self-occlusions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):493–506, 2008. 95

[GPS+07] 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*, 28(1):1–18, 2007. 25, 142

[Gra98] F.S. Grassia. Practical parameterization of rotations using the exponential map. *Journal of Graphic Tools*, 3(3):29–48, 1998. 80, 143

[GRS95] W.R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. CRC Press, 1995. 18

[GSS93] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140(2):107–113, 1993. 18

[HFP+01] L. Herda, P. Fua, R. Plankers, R. Boulic, and D. Thalmann. Using skeleton-based tracking to increase the reliability of optical motion capture. *Human Movement Science*, 20(3):313–341, 2001. 86

[HHD00] I. Haritaoglu, D. Harwood, and L.S. Davis. W^4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000. 1, 30, 77

[HLGB03] J.M. Hasenfratz, M. Lapierre, J.D. Gascuel, and E. Boyer. Real-time capture, reconstruction and insertion into virtual world of human actors. *Vision, Video and Graphics*, pages 49–56, 2003. 1, 12

[HM04] D.L. Hall and S.A.H. McMullen. *Methematical Techniques in Multisense Data Fusion*. Artech House, 2004. 31

[Hor87] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, 1987. 80

[HUF05] L. Herda, R. Urtasun, and P. Fua. Hierarchical implicit surface joint limits for human body tracking. *Computer Vision and Image Understanding*, 99(2):189–209, 2005. 80, 83, 85, 92

[HW07] Z.L. Husz and A.M. Wallance. Evaluation of a hierarchical partitioned particle filter with action primitives. In *Proceedings of 2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation*, 2007. 84, 95, 138

[HZ04] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 7, 9, 10, 73, 88, 143, 144

[IB98] M. Isard and A. Blake. CONDENSATION–conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. 17, 21, 78

[IS03] J. Isidoro and S. Sclaroff. Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 1335–1342, 2003. 11, 30, 31

[JBY96] S.X. Ju, M.J. Black, and Y. Yacoob. Cardboard people: a parameterized model of articulated image motion. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 38–44, 1996. 72, 74

[JW07] R.A. Johnson and D.W. Wichern. *Applied multivariate statistical analysis*. 2007. 63, 67

[KBD03] Z. Khan, T. Balch, and F. Dellaert. Efficient particle filter-based tracking of multiple interacting targets using an MRF-based motion model. In *Proceedings of International Conference on Intelligent Robots and Systems*, volume 1, pages 254–259, 2003. 30, 40

[KG06] R. Kehl and L.V. Gool. Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding*, 104(2):190–209, 2006. 5, 73, 76, 90, 137

[KGV83] S. Kirkpatrick, C.D. Gellatt, and M.P. Vecchi. Optimisation by simulated annealing. *Science*, 220(4598):671–680, 1983. 22

[Kit96] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian non-linear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996. 19

[KM00] L. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 22(12):1453–1459, 2000. 5, 74, 90

[KOF05] A.G. Kirk, J.F. O'Brien, and D.A. Forsyth. Skeletal parameter estimation from optical motion capture data. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 782–788, 2005. 75, 85, 86

[KS00] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000. 11, 31

[KTPP07] N. Katsarakis, F. Talantzis, A. Pnevmatikakis, and L. Polymenakos. The AIT 3D audio-visual person tracker for CLEAR 2007. In *Proceedings of Classification of Events, Activities and Relationships Evaluation and Workshop*, volume 4625 of *Lecture Notes on Computer Science*, pages 35–46, 2007. 30, 52

[KZ04] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004. 31

[Lan06] O. Lanz. Approximate bayesian multibody tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1436–1449, 2006. 15, 30, 31, 74, 76

[Lan07] J.L. Landabaso. *A Unified Framework for Consistent 2D/3D Foreground Object Detection*. PhD thesis, Technical University of Catalonia, 2007. 11, 12

[LC98] J.S Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998. 18

[LC03] M.W. Lee and I. Cohen. Human body tracking with auxiliary measurements. In *Proceedings of IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 112–119, 2003. 75

[LCB07] O. Lanz, P. Chippendale, and R. Brunelli. An appearance-based particle filter for visual tracking in smart rooms. In *Proceedings of Classification of Events, Activities and Relationships Evaluation and Workshop*, volume 4625 of *Lecture Notes on Computer Science*, pages 57–69, 2007. 30, 52

[LCFC07] A. López, C. Canton-Ferrer, and J.R. Casas. Multi-person 3D tracking with particle filters on voxels. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 913–916, 2007. 29, 30, 140

[Leu91] J.G. Leu. Computing a shape's moments from its boundary. *Pattern Recognition*, 24(10):116–122, 1991. 41

[LH08] K. Lien and C. Huang. Multiview-based cooperative tracking of multiple human objects. *EURASIP Journal on Image and Video Processing*, 8(2), 2008. 30

[LP05] J.L. Landabaso and M. Pardàs. Foreground regions extraction and characterization towards real-time object tracking. In *Proceedings of 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, volume 3869 of *Lecture Notes in Computer Science*, pages 241–249, 2005. 11

[LPC08] J.L. Landabaso, M. Pardàs, and J.R. Casas. Shape from inconsistent silhouette. *Computer Vision and Image Understanding*, 112(2):210–224, 2008. 12

[LRH04] J. Lichtenauer, M. Reinders, and E. Hendriks. Influence of the observation likelihood function on particle filtering performance in tracking applications. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 767–772, 2004. 22

[Mal89] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989. 112

[MB06] J. Man and B. Bhanu. Individual recognition using gait energy image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):316–322, 2006. 74, 76

[MB07] J. Madapura and L. Baoxin. 3D articulated human body tracking using KLD-annealed Rao-Blackwellised particle filter. In *Proceedings of IEEE International Conference on Multimedia and Expo*, volume 2, pages 1950–1953, 2007. 78

[MCA06] L. Münderman, S. Corazza, and T.P. Andriacchi. Markerless human motion capture through visual hull and articulated ICP. In *Proceedings of 1st Workshop on Evaluation of Articulated Human Motion and Pose Estimation*, 2006. 138

[MEM+08] J. Marcello, F. Eugenio, F. Marqués, A. Hernandez-Guerra, and A. Gasull. Motion estimation techniques to automatically track oceanographic thermal structures in multisensor image sequences. *IEEE Transactions on Geoscience and Remote Sensing*, 46(9):2743–2762, 2008. 141

[MGPB+05] I. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang. Automatic analysis of multimodal group actions in meetings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):305–317, 2005. 30

[MH03] J. Mitchelson and A. Hilton. Simultaneous pose estimation of multiple people using multiple-view cues with hierarchical sampling. In *Proceedings of British Machine Vision Conference*, 2003. 78, 93, 112

[MHK06] T.B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2–3):90–126, 2006. 72, 76

[MI00] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proceddings of European Conference on Computer Vision*, pages 3–19, 2000. 19, 78

[Mik03] I. Mikič. *Human body model acquisition and tracking using multi-camera voxel data*. PhD thesis, University of California, San Diego, 2003. 17, 68, 73, 76, 80, 110, 124, 137, 142

[Mit97] T. Mitchel. *Machine Learning*. McGraw Hill, 1997. 36

[Mit98] M. Mitchel. *An Introduction to Genetic Algorithms*. MIT Press, 1998. 122

[Mit03] J.R. Mitchelson. *Multiple-camera studio methods for automated measurement of human motion*. PhD thesis, University of Surrey, 2003. 21

[Mov] Moven-Inertial Motion Capture. http://www.moven.com. 85

[MPRC07] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro. Particle PHD filtering for multi-target visual tracking. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 1101–1104, 2007. 32

[MRR+53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953. 22, 23

[MSJ00] I. Mikič, S. Santini, and R. Jain. Tracking objects in 3D using multiple camera views. In *Proceedings of Asian Conference on Computer Vision*, 2000. 30, 81

[MSKS03] Y. Ma, S. Soatto, J. Kosecka, and S. Shankar. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2003. 73

[MSZ94] R.M. Murray, S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994. 81, 143, 144

[NPS+09] K. Nickel, M. Pardàs, R. Stiefelhagen, C. Canton-Ferrer, J.L Landabaso, and J.R. Casas. *Computers in the Human Interaction Loop*, chapter Activity classification, pages 109–119. Springer-Verlag, 2009. 29, 140

[OCFT+08a] F. Ofli, C. Canton-Ferrer, J. Tilmanne, Y. Demir, E. Bozkurt, Y. Yemez, E. Erzin, and A.M. Tekalp. An audio-driven dancing avatar. *Journal on Multimodal User Interfaces*, 2(2):93–103, 2008. 1, 140

[OCFT+08b] F. Ofli, C. Canton-Ferrer, J. Tilmanne, Y. Demir, E. Bozkurt, Y. Yemez, E. Erzin, and A.M. Tekalp. Audio-driven human body motion analysis and synthesis. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2233–2236, 2008. 71, 99, 140

[ODCF+08] F. Ofli, Y. Demir, C. Canton-Ferrer, J. Tilmanne, K. Balci, E. Bozkurt, I. Kizoglu, Y. Yemez, E. Erzin, A.M. Tekalp, L. Akarun, and T.A. Erdem. Analysis and synthesis of multiview audio-visual dance figures. In *Proceedings of IEEE Signal Processing, Communication and Applications Conference*, pages 1–4, 2008. 71, 99, 140

[ODE+07] F. Ofli, Y. Demir, E. Erzin, Y. Yemez, and A.M. Tekalp. Multicamera audio-visual analysis of dance figures. In *Proceedings of IEEE International Conference on Multimedia and Expo*, volume 1, pages 1703–1706, 2007. 78

[OM02] G. Olague and R. Mohr. Optimal camera placement for accurate reconstruction. *Pattern Recognition*, 35(4):927–944, 2002. 7, 73

[Ord05] F. Orderud. Comparison of Kalman filter estimation approaches for state space models with nonlinear measurements. In *Proceedings of Scandinavian Conference on Simulation and Modeling*, 2005. 17

[OS08] R. Okada and S. Soatto. Relevant feature selection for human pose estimation and localization in cluttered images. In *Proceedings of European Conference on Computer Vision*, 2008. 90, 138

[OWS+07] T. Osawa, X. Wu, K. Sudo, K. Wakabayashi, and H. Arai. MCMC based multi-body tracking using full 3D model of both target and environment. In *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 224–229, 2007. 30

[PET07] PETS - Performance Evaluation of Tracking and Surveillance. http://pets2007.net, 2007. 59

[PHRR00] P.J. Phillips, M. Hyeonjoon, S.A. Rizvi, and P.J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000. 59

[Pic04] M. Piccardi. Background subtraction techniques: a review. In *Proceedings IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3099–3104, 2004. 11

[Pin70] M. Pincus. A Monte Carlo method for the approximate solution of certain types of constrained optimization problems. *Operations Research*, 18:1225–1228, 1970. 22

[Pop07a] R. Poppe. Evaluating example-based pose estimation: Experiments on the humaneva sets. In *Proceedings of 2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation*, 2007. 138

[Pop07b] R. Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1–2):4–18, 2007. 72

[PP02] A. Papoulis and S.U. Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw Hill, 2002. 84

[PT08] S. Park and M.M. Trivedi. Understanding human interactions with track and body synergies (TBS) captured from multiple views. *Computer Vision and Image Understanding*, 111(1):2–20, 2008. 30

[RBM05] X. Ren, A.C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *Proceedings of IEEE International Conference on Computer Vision*, volume 1, pages 824–831, 2005. 61

[RMG+02] J. Rickel, S. Marsella, J. Gratch, R. Hill, D. Traum, and W. Swartout. Toward a new generation of virtual humans for interactive experiences. *IEEE Intelligent Systems*, 17(4):32–38, 2002. 1

[Roe06] D. Roetenberg. *Inertial and Magnetic Sensing of Human Motion*. PhD thesis, University of Twente, 2006. 85

[RRR08] L. Raskin, E. Rivlin, and M. Rudzsky. Using Gaussian process annealing particle filter for 3D human tracking. *EURASIP Journal on Advances in Signal Processing*, 2008. 20, 21, 74, 79, 83, 92, 93, 94, 110

[SB01] H. Sidenbladh and M.J. Black. Learning image statistics for Bayesian tracking. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 709–716, 2001. 76

[SB06] L. Sigal and M.J. Black. HumanEva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Department of Computer Science, Brown University, 2006. 2, 7, 59, 60, 61, 62, 74, 96, 110

[SC07] J. Saboune and F. Charpillet. Markerless human motion tracking from a single camera using interval particle filtering. *International Journal on Artificial Intelligence Tools*, 16(4):593–609, 2007. 74, 78

[SCFCH07] C. Segura, C. Canton-Ferrer, J.R. Casas, and J. Hernando. Multimodal head orientation towards attention tracking in smartrooms. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007. 140

[SD99] S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999. 12

[SG99] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 252–259, 1999. 11

[SG00] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000. 74

[SKLM05] C. Sminchisescu, K. Kanauija, Z. Li, and D. Metaxas. Discriminative density propagation for 3D human motion estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 390–397, 2005. 67, 72

[SMP05] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14:407–422, 2005. 73

[SP94] P. Salembier and M. Pardàs. Hierarchical morphological segmentation for image sequence coding. *IEEE Transactions on Image Processing*, 3(5):639–651, 1994. 54

[SPL+05] S. Sarkar, P.J. Phillips, Z. Liu, I.R. Vega, P. Grother, and K.W. Bowyer. The HumanID gait challenge problem: data sets, performance, and analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):162–177, 2005. 59

[Sto01] L.D. Stone. *A Bayesian Approach to Multiple-Target Tracking*. Handbook of Multisensor Data Fusion. CRC Press, 2001. 32

[Sze93] R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics and Image Processing*, 58(1):23–32, 1993. 12

[TMSS02] C. Theobalt, M. Magnor, P. Schuler, and H.P. Seidel. Combining 2D feature tracking and volume reconstruction for online video-based human motion capture. In *Proceedings of 10^{th} Pacific Conference on Computer Graphics and Applications*, pages 96–103, 2002. 112, 124

[TPC08] F. Talantzis, A. Pnevmatikakis, and A.G. Constantinides. Audio-visual active speaker tracking in cluttered indoors environments. *IEEE Transactions on Systems, Man, and Cybernetics (Part B)*, 38(3):799–807, 2008. 32

[Tuc77] J.W. Tuckey. *Exploratory Data Analysis*. Addison-Wesley, 1977. 34

[UD08] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 90, 138

[UFF06] R. Urtasun, D.J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 238–245, 2006. 79, 80, 110

[VAC] VACE-Video Analysis and Context Extraction. http://www.ic-arda.org. 47

[Vas08] N. Vaswani. Particle filtering for large-dimensional state spaces with multimodal observation likelihoods. *IEEE Transactions on Signal Processing*, 56(10):4583–4597, 2008. 142

[VGCF+09] M. Voit, N. Gourier, C. Canton-Ferrer, O. Lanz, R. Stiefelhagen, and R. Brunelli. *Computers in the Human Interaction Loop*, chapter Estimation of Head Pose, pages 33–42. Springer-Verlag, 2009. 140

[Vic] VICON. http://www.vicon.com. 3, 60, 75, 86, 101

[Vin93] L. Vincent. Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2(2):176–201, 1993. 54

[VU05] M.C. Villa-Uriol. *Video-Based Avatar Reconstruction and Motion Capture*. PhD thesis, University of California, 2005. 1, 11, 73, 80

[WB95] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical report, University of Chapel Hill, NC, USA, 1995. 17

[WC04] S.S. Wong and K.L. Chan. Multi-view 3D model reconstruction: Exploitation of color homogeneity in voxel mask. In *Proceedings of IEEE International Conference on Image and Graphics*, pages 142–145, 2004. 12

[Wer07] N. Werghi. Segmentation and modeling of full human body shape from 3-D scan data: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1122–1136, 2007. 72

[WH97] M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models*. Springer-Verlang, New York, 2nd edition, 1997. 16

[YA96] L. Yang and F. Albregtsen. Fast and exact computation of Cartesian geometric moments using discrete Green's theorem. *Pattern Recognition*, 29(7):1061–1073, 1996. 41

[YJS06] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006. 30

[YS05] A. Yilma and M. Shah. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *Proceedings of IEEE International Conference on Computer Vision*, volume 1, pages 150–157, 2005. 1, 73

[YZG08] L. Ye, Q. Zhang, and L. Guan. Use hierarchical genetic particle filter to figure articulated human tracking. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1561–1564, 2008. 122

[Zha02] Z. Zhang. A flexible new technique for camera calibration. Technical report, Microsoft Research, Aug 2002. 10, 73

[ZL00] D. Zhang and G. Lu. Segmentation of moving objects in image sequence: A review. *Circuits, Systems, and Signal Processing*, 20(2):143–183, 2000. 11

[ZNS06] J. Ziegler, K. Nickel, and R. Stiefelhagen. Tracking of the articulated upper body on multi-view stereo image sequences. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 774–781, 2006. 73, 142