Universitat
Autònoma
de Barcelona

Motion Priors for Efficient Bayesian Tracking in Human Sequence Evaluation

A dissertation submitted by **Ignasi Rius** at Universitat Autònoma de Barcelona to fulfil the degree of **PhD in Computer Science.**

Barcelona, June 2010

Director: **Jordi González i Sabaté**
Universitat Autònoma de Barcelona
Computer Vision Centre
Co-director: **Xavier Roca Marvà**
Universitat Autònoma de Barcelona
Computer Vision Centre

Centre de Visió per Computador

This document was typeset by the author using LaTeX2ε.

The research described in this book was carried out at the Computer Vision Centre, Universitat Autònoma de Barcelona.

ISBN-13: 978-84-937261-6-3
ISBN-10: 84-937261-6-2

Printed by Edicions Gràfiques Rey, S.L.

"Mind the gap."

—Anònim

Agraïments

Ja fa temps que penso que hi ha moltes coses a la vida que són fruit d'encadenar una sèrie de casualitats impossibles de predir. És una idea que em fascina, perquè implica que en cada acció, cada pensament i cada decisió que un pren, per banal que sembli, es decideix el destí d'un mateix sense ni tan sols adonar-se'n. Sovint em sorprenc fent memòria de com he anat a parar a un lloc determinat a la vida per mirar de desentrellar aquesta cadena de casualitats. Avui em toca fer memòria de com he arribat a fer aquest llibre que representa la culminació d'una etapa que tenca el meu pas per la universitat com a estudiant de doctorat.

Ha estat una etapa plena de reptes que va començar un bon dia quan em vaig presentar al Centre de Visió per Computador amb la idea al cap de provar sort amb això de la investigació sense saber massa bé on em ficava. Un cop allà, en Juanjo Villanueva em va introduir els diferents projectes de recerca en marxa, i sense saber com, vaig acabar al despatx d'en Xavi Roca fent una trucada a un tal Poal, que llavors es trobava a Karlsruhe amb el professor Nagel, per demanar-li com ho veia això de dirigir-me una tesis i de començar la seva pròpia línia d'investigació al CVC. Ara sé que la meva pregunta va ser tan inesperada per ell com per mi, però, coses del destí, a en Poal no li va semblar malament la proposta. I aquí ens trobem, al cap de set llargs anys, presentant-vos aquest llibre.

Aquest llibre no s'ha fet sol, i tot i que, tal i com diu a la coberta, en sóc l'autor material, compta amb moltíssims més autors morals, coautors d'idees, col.laboradors esporàdics i fins i tot guies espirituals que l'han fet possible. Per això voldria dedicar aquesta secció a tot un seguit de gent que m'han acompanyat durant aquest procés, ja sigui guiant-me, animant-me, treient-me la son de les orelles als matins, o fent-la petar quan ha convingut.

En primer lloc, vull agrair molt especialment a en Poal haver dit que sí el dia que va rebre la meva trucada, i haver-me fet de supervisor, tutor i company durant els transcurs de la meva tesis. Voldria dir també, que sento una gran admiració per la seva valentia, força de voluntat i dedicació per haver tirat endavant, partint de zero, un grup d'investigació consolidat com és avui

l'ISE. Gràcies també per cuidar-me durant tot aquest temps, per guiar-me i per animar-me quan m'ha convingut i per haver-me donat les empentetes necessàries que han fet que finalment aquest llibre sigui possible.

També voldria agrair al professor Juanjo Villanueva la seva capacitat emprenedora per haver dedicat gran part dels seus esforços a tirar endavant el CVC en el sí de la Universitat Autònoma de Barcelona, i haver-me fet confiança el dia que em vaig presentar al seu despatx sent un complet desconegut. Voldria també estendre el meu agraïment a en Xavi Roca per haver-me apadrinat a l'aterrar al centre, perquè sempre que l'he necessitat ha estat allà i perquè és una persona que té la capacitat de fer fàcils les coses difícils.

Per res del món voldria deixar-me el meu company de batalla " sevillano", en Dani, amb el qual he compartit gran part de la meva estada al CVC, i que sens dubte ha sigut una font d'inspiració constant en aquest treball fruit del seu esperit crític i inconformista. Estiguis en el país que estiguis en aquests moments, moltes gràcies per tot i fins aviat!

Tampoc voldria deixar d'esmentar a la resta de companys de despatx i del CVC en general que, sobretot, han contribuït a fer molt més amena i entretinguda la meva estada allà. A tots vosaltres, Edu, Marçal, Alícia, Joan, Aura, àgata, David, Jaume, Sergio, Xavi, Mari, Eric, Débora, Bashkar, Ariel, Murad, Marco i altres, merci per compartir xerrades, moments, cafès i dinars durant aquests anys i per ser uns bons companys de "feina". Finalment, voldria fer extensiu el meu agraïment a tothom del CVC pels dinars, barbacoes, hores de cafeteria, etc. sense els quals el CVC només seria un edifici.

Seguint en el pla professional, voldria donar les gràcies a en Xavi Varona que em va obrir les portes de la Universitat de les Illes Balears durant la meva estada allà i fer-ho extensiu a en Toni, na Cristina i als altres companys. Voldria destacar les xerrades al voltant de l'edifici Anselm Turmeda discutint articles amb en Xavi i el seu punt de vista fresc de les coses, ja que sens dubte, han sigut vitals per acabar aquesta feina. Tot plegat, va contribuir a que m'endugués un bon record del meu pas per la illa i gaudir durant quatre mesos d'aquest entorn fantàstic que és Mallorca i del tarannà de la seva gent.

Also, I would like to thank Dr. Thomas Moeslund for his valuable advices during my stage at the CVMT laboratory in the University of Aalborg, Denmark. Despite the rain, the time I spent there was very profitable and helped me sketch the outline of the work I am presenting in this book.

This work has been supported by the Catalan Research Agency (AGAUR), by the Spanish Ministry of Education (MEC) under projects AHNA (TIC2003-08865), SYSIPHUS (TIN2006-14606), and DPI-2004-5414, and by the EC under projects HERMES (IST-027110) and Vidi-Video (IST-045547).

Voldria fer menció també als meus companys actuals de feina a Mediapro

que de ben segur també han "patit" l'etapa final de redacció d'aquesta tesis en forma d'un company de feina a vegades adormit i estressat.

En el lloc més important de tots, vull donar les gràcies a la Meritxell per la paciència que ha tingut i l'ajuda incondicional que m'ha donat al llarg de tots aquests anys. Només ella sap la quantitat d'hores que aquesta tesis li ha robat en forma de caps de setmana perduts davant l'ordinador, mesos d'exili a l'estranger i tots els sacrificis que implica fer un treball com aquest. Perquè en quedi constància per escrit i per mirar de compensar una mica la cosa, un cop sigui doctor prometo que m'ocuparé íntegrament de la cuina durant 3 mesos seguits com a mínim!

A tots els meus amics, Laura, Tià, Xevi, Montse, Raquel, Lè, Tame, Vane, Gerard, i a tots els del Cau, així com als companys de la universitat. També als amics de teatre, Ivan, David, Montse, Fina, Àngels, Txell, Maria, Albert, Dolors i molts d'altres que han contribuït a fer amenes les hores que no he dedicat a la tesis ni a la feina. A tots ells, moltes gràcies i espero que m'ajudin a recuperar la meva vida social un cop acabada aquesta tesis.

I per acabar, voldria agrair a la meva família tot el que ha fet per mi que és impossible de resumir en aquestes pàgines. En especial, gràcies mare, perquè de mare només n'hi ha una, i per haver-me criat, per la teva tendresa i pel teu suport incondicional a la vida. Gràcies pare per satisfer la meva avidesa incessant de voler-ne saber més i per contestar a totes les preguntes d'aquell nen que es va passar mitja infantesa preguntant perquè això i perquè allò. També per haver-me donat sempre un cop de mà quan jo no me'n sortia, i per transmetre'm aquesta afició a "fer invents" tan nostra. Gràcies Ariadna, perquè tot i que no ens veiem tant com voldríem, sempre comptes amb mi i estàs allà quan et necessito. Espero que ara que tindré més temps lliure, i tot i que treballes a la competència, ens puguem veure més sovint encara que no paris de viatjar pel món. Voldria dedicar en especial aquest treball al meu avi Ignasi per haver estat un avi fantàstic i per haver-me transmès els valors de l'esforç en el treball i la honradesa. Amb tu vaig compartir molts dies d'infantesa aprenent a anar amb bicicleta, a nedar, a multiplicar, descobrint el gust de caminar, i ara de gran, he seguit gaudint de la teva companyia i de les nostres xerrades, i per això tinc la certesa que sempre et portaré amb mi allà on vagi. Moltes gràcies a tots per no conformar-se i fer-me anar sempre més enllà, per formar-me com a persona, per estimar-me com a fill, i per compartir amb mi les penes i les alegries.

Moltes gràcies a tots.

Resum

La reconstrucció del moviment humà mitjançant l'anàlisi visual és una àrea de recerca de la visió per computador plena de reptes amb moltes aplicacions potencials. Els enfocs de seguiment basat en models, i en particular els filtres de partícules, formulen el problema com una tasca d'inferència Bayesiana l'objectiu de la qual és estimar seqüencialment la distribució sobre els paràmetres d'un model del cos humà al llarg del temps. Aquests enfocs depenen en gran mesuta d'emprar bons models dinàmics i d'observació per tal de predir i actualitzar les configuracions del cos humà en base a mesures extretes de les dades d'imatge. No obstant, resulta molt difícil dissenyar models d'observació, i en especial pel cas de seguiment a partir d'una sola vista, que siguin capaços d'extreure informació útil de les seqüències d'imatges de manera robusta. Per tant, per tal de superar aquestes limitacions és necessari emprar un fort coneixement a priori sobre el moviment humà i guiar així l'exploració de l'espai d'estats.

El treball presentat en aquesta Tesis està enfocat a recuperar els paràmetres de moviment 3D d'un model del cos humà a partir de mesures incompletes i sorolloses d'una seqüència d'imatges monocular. Aquestes mesures consisteixen en les posicions 2D d'un conjunt reduït d'articulacions en el pla d'imatge. Amb aquesta finalitat, proposem un nou model de moviment humà específic per cada acció, que és entrenat a partir de bases de dades de captures de moviment que contenen vàries execucions d'una acció en particular, i que és utilitzat com a coneixement a priori en un esquema de filtratge de partícules.

Les postures del cos es representen emprant un model articulat simple i compacte que fa ús dels cosinus directors per tal de representar la direcció de les parts del cos en l'espai Cartesià 3D. Llavors, donada una acció, s'aplica l'Anàlisis de Components Principals (PCA) sobre les dades d'entrenament per tal d'aplicar reducció de dimensionalitat sobre les dades d'entrada altament correlacionades. Prèviament al pas d'entrenament del model d'acció, les seqüències de moviment d'entrada són sincronitzades mitjançant un nou algoritme d'adaptació dens basat en Programació Dinàmica. L'algoritme sincronitza totes les seqüències de moviment d'una mateixa classe d'acció i és capaç de trobar una solució òptima en temps real.

Aleshores, s'aprèn un model d'acció probabilístic a partir dels exemples de moviment sincronitzats que captura la variabilitat i l'evolució temporal del moviment del cos sencer durant una acció concreta. En particular, per cada acció, els paràmetres apresos són: una varietat representativa de l'acció que consisteix en l'execució mitjana de la mateixa, la desviació estàndard de l'execució mitjana, els vectors de direcció

mitjans de cada subseqüència de moviment d'una llargada donada i l'error esperat en un instant de temps donat.

A continuació, s'utilitza el model específic per cada acció com a coneixement a priori sobre moviment humà que millora l'eficiència i robustesa de tot l'enfoc de seguiment basat en filtratge de partícules. En primer lloc, el model dinàmic guia les partícules segons situacions similars apreses prèviament. A continuació, es restringeix l'espai d'estats per tal que tan sols les postures humanes més factibles siguin acceptades com a solucions vàlides a cada instant de temps. En conseqüència, l'espai d'estats és explorat de manera més eficient ja que el conjunt de partícules cobreix les postures del cos més probables.

Finalment, es duen a terme experiments emprant seqüències de test de vàries bases de dades. Els resultats assenyalen que el nostre esquema de seguiment és capaç d'estimar la configuració 3D aproximada d'un model de cos sencer, a partir tan sols de les posicions 2D d'un conjunt reduït d'articulacions. També s'inclouen proves separades sobre el mètode de sincronització de seqüències i de la tècnica de comparació probabilística de les subseqüències de moviment.

Abstract

Recovering human motion by visual analysis is a challenging computer vision research area with a lot of potential applications. Model-based tracking approaches, and in particular particle filters, formulate the problem as a Bayesian inference task whose aim is to sequentially estimate the distribution of the parameters of a human body model over time. These approaches strongly rely on good dynamical and observation models to predict and update configurations of the human body according to measurements from the image data. However, it is very difficult to design observation models which extract useful and reliable information from image sequences robustly. This results specially challenging in monocular tracking given that only one viewpoint from the scene is available. Therefore, to overcome these limitations strong motion priors are needed to guide the exploration of the state space.

The work presented in this Thesis is aimed to retrieve the 3D motion parameters of a human body model from incomplete and noisy measurements of a monocular image sequence. These measurements consist of the 2D positions of a reduced set of joints in the image plane. Towards this end, we present a novel action-specific model of human motion which is trained from several databases of real motion-captured performances of an action, and is used as a priori knowledge within a particle filtering scheme.

Body postures are represented by means of a simple and compact stick figure model which uses direction cosines to represent the direction of body limbs in the 3D Cartesian space. Then, for a given action, Principal Component Analysis is applied to the training data to perform dimensionality reduction over the highly correlated input data. Before the learning stage of the action model, the input motion performances are synchronized by means of a novel dense matching algorithm based on Dynamic Programming. The algorithm synchronizes all the motion sequences of the same action class, finding an optimal solution in real-time.

Then, a probabilistic action model is learnt, based on the synchronized motion examples, which captures the variability and temporal evolution of full-body motion within a specific action. In particular, for each action, the parameters learnt are: a representative manifold for the action consisting of its mean performance, the standard deviation from the mean performance, the mean observed direction vectors from each motion subsequence of a given length and the expected error at a given time instant.

Subsequently, the action-specific model is used as a priori knowledge on human motion which improves the efficiency and robustness of the overall particle filtering

tracking framework. First, the dynamic model guides the particles according to similar situations previously learnt. Then, the state space is constrained so only feasible human postures are accepted as valid solutions at each time step. As a result, the state space is explored more efficiently as the particle set covers the most probable body postures.

Finally, experiments are carried out using test sequences from several motion databases. Results point out that our tracker scheme is able to estimate the rough 3D configuration of a full-body model providing only the 2D positions of a reduced set of joints. Separate tests on the sequence synchronization method and the subsequence probabilistic matching technique are also provided.

Keywords: *Human Motion Modeling; Particle filtering; Monocular Full Body 3D Tracking.*

Topics: *Image Processing; Computer Vision; Scene Understanding; Machine Intelligence; Machine Vision Applications; Video-Sequence Evaluation*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human motion capture is the problem of recording the motion of the human body for immediate or delayed analysis and playback. The first known works using video data to analyze human body motion date from the 19th century and were carried out by Eadweard James Muybridge and Etienne-Jules Marey. In 1878, Muybridge became the first person to record animal motion over time by taking a series of photographs of a horse in motion. His pictures from animal and human motion are still available today in his books [55]. In addition, Marey published pictures of birds in flight in the early 1880's made with his "photographic gun" [10]. His invention was a forerunner of the motion picture camera, and consisted of a sight and a clock mechanism which allowed to take 12 exposures of 1/72th of a second each. His observations concerning the changes in the shape of birds' wings in relation to air resistance were a great revolution in understanding the phenomenon of flight. He also extended his work to human motion analysis in questions such as fatigue minimization related to how soldiers march while carrying a heavy pack.

However, it hasn't been until the last 20 years that automatic video-based motion analysis appeared as an active research problem. Nowadays, there are many commercial Motion Capture (MoCap) systems available based on visual analysis techniques. They can provide very fast, generic and accurate recordings of the motion performed by any subject, which makes this systems very suitable for applications such as sports performance enhancement, medical analysis or animation of virtual characters from MoCap data, which is becoming more and more popular in the video game and film production industry.

Unfortunately, such systems require multiple cameras recording the scene from different points of view in a controlled illumination environment. As a result, the capturing volume is usually very limited and involve expensive and complex setups. In addition, they require that the recorded subject wears non-natural markers such as special clothing, reflective makeup, reflective balls or active LED-based markers, among others. This fact limits the range of suitable applications, making it impractical for TV footage-based MoCap, automatic content annotation, video-surveillance or scene understanding among others, which usually involve an uncontrolled environment and there is only one simultaneous camera recording the scene. Therefore, researchers

have been studying more flexible solutions for motion capture which do not require any special set up, and that could work from any number of cameras or video sources. Additionally, in many scenarios, the action can only be seen from a single camera at a time.

In consequence, a great number of literature has been devoted to the problem of full body 3D tracking from a monocular image sequence over the past years [53, 83, 3]. Nowadays, this is still a very active research area [53, 72, 60], as many related difficulties remain as an open problem. The work presented in this Thesis is all part of these efforts. Hence, it tackles the problem of recovering the 3D human body motion parameters from a monocular image sequence, as a previous step to produce qualitative descriptions suitable for scene understanding applications such as automatic content annotation or smart video surveillance.

1.1 Motivation

Visual human motion analysis attempts to understand from image sequences what is happening in the scene in terms of human actions, their evolutions and their interactions. Although it concerns a lot of hard issues ranging from human detection and tracking, people identification, body parts detection and action recognition among others, one of the most challenging task relies in modeling, analyzing, and recovering human motion from image sequences which is the aim of this work. This domain is referred as *Human Sequence Evaluation* (HSE) in the framework presented by Gonzàlez in [27], and provides a general scheme for producing human motion descriptions from image sequences suitable to be used for scene understanding applications.

Fig. 1.1 presents an overview of the architecture required to perform HSE. Hence, the HSE framework divides the task of evaluating sequences of images involving human motion in several layers or modules, each one encapsulating different domains of knowledge.

Hence, the interpretation of human motion is treated as a transformation process of knowledge from level to level. Therefore, each module has its own associated models, and the transforming processes between levels must be defined. From this point of view, we can see the process of evaluating sequences of human motion, as a proper parametrization of the models involved in each level of abstraction. Notice that all levels are interconnected between them in order to cooperate and successfully describe human motion: higher levels may help lower levels by performing reasoning about the most plausible actions of a subject given a certain situation. On the other hand, lower levels may help the higher ones by correcting and updating their degree of belief about what is happening on the scene.

The first level is called *Signal Level*. It deals with raw video data signals and information about the camera parameters. Hence, this level has control about the acquisition procedure of images from the reality and can control the viewing conditions. The next level is the *Image Level* . At this stage, we aim to process the raw video signals from the previous level frame by frame in order to segment the regions of interest (ROI) of the image. As a result, a a first representation of the data is provided to the following layer, namely the *Picture Domain Level*. Here, pos-

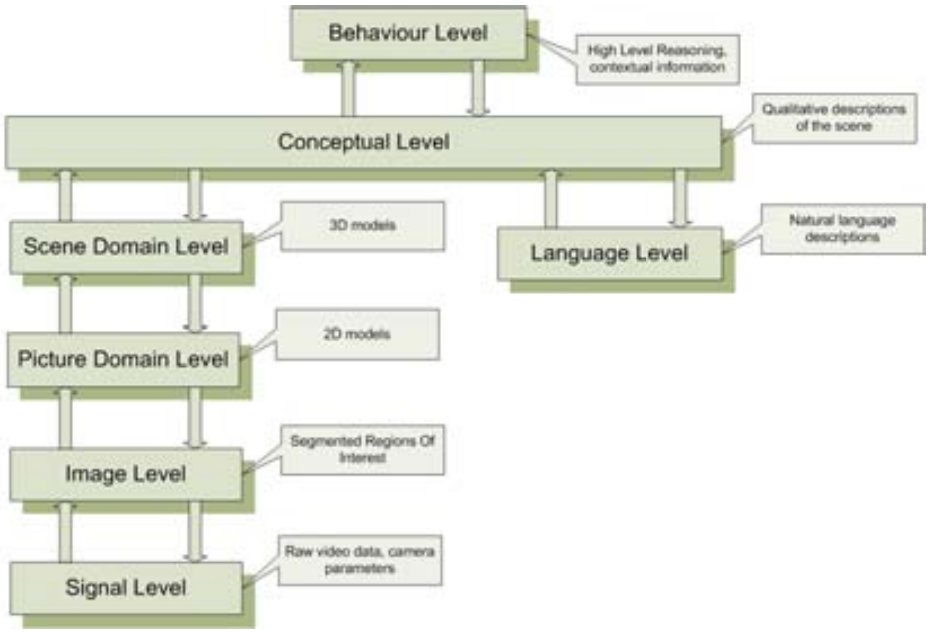


Figure 1.1: An overview of the HSE framework.

sible segmentation errors from the previous level are handled and 2D analysis of the segmented regions is carried out in order track them. Therefore, labeled targets are supplied to the *Scene Domain Level*. This level deals with 3D models of the human body and a priori knowledge on human dynamics in order to perform model-based tracking of a 3D human body model. The quantitative knowledge extracted at this level is forwarded to the *Conceptual Level* in order to deduce qualitative descriptions of the scene at each time step. As a result, plausible descriptions are inferred and updated over time. Then, the *Behavioural Level* performs spatio-temporal reasoning on the observed scene in order to predict expected behaviours and provide contextual information to the overall analysis process. Finally, the *Language Level* attempts to use this information in order to produce natural language descriptions of what is actually happening within the scene.

The work of this Thesis is focused on the transformation process between the *Picture Domain Level* and the *Scene Domain Level*. In other words, the computation of the 3D human body poses from the 2D image sequences, enabling the production of qualitative descriptions of the human motion present in a scene to be used within the HSE scheme. For instance, possible applications include advanced video surveillance systems which could react while a situation is occurring, rather than being used as a forensic tool as nowadays. Therefore, as opposed to commercial MoCap systems, the goal of this work is not to achieve the best possible accuracy when reconstructing highly realistic body models, but to obtain an adequate estimate of the motion parameters of a simple body model, robustly from a monocular and restricted view of the scene.

1.2 Applications

The application domains related to visual human motion analysis are wide and broad. For instance, to name a few, one may want to recover and reconstruct the 3D body poses from 2D image sequences for enhancing sports performance. Alternatively, it might be interesting to synthesize realistic image sequences from 3D models of motion learnt from existing TV-footage. Also, smart video surveillance applications could benefit from having a better comprehension of what is happening in the scene.

In the following, we state some of the most interesting applications related to visual human motion analysis within the HSE framework. In addition, in an attempt to organize the enumerated applications, we have grouped them into 3 related technologies, namely, activity and gesture recognition, motion capture and motion synthesis.

1.2.1 Activity and Gesture Recognition

Activity recognition aims to identify which action/activity a human agent is performing during a given motion sequence. According to Kurtenbach et al. [42], "A gesture is a motion of the body that contains information". Hence, gesture recognition rather than treating an action as a whole, is aimed at recognizing the meaningful gestures involved in the articulation of a physical action. Interesting application areas for these technologies comprise:

1. **Smart video surveillance:** Current surveillance systems have poor, or non-existing qualitative data available to make high-level decisions. In most cases they consist in simple video recording systems, and they are used as a "forensic" tool when the fact to analyze has already happened. Even more sophisticated systems are based in simple motion detection, and in most cases they require human supervision when some movement appears in the scene not taking into consideration what is generating that motion, or which kind of motion is. The incorporation of qualitative information into a smart surveillance system could be used to build a system that describes where, when, and what an agent is performing in the monitored scene. Hence, such a system would be able to make high-level reasoning about what is happening in the scene, make decisions according to it, and minimize the intervention of a human supervisor by only requiring its attention when suspicious or ambiguous behaviours are being detected. For instance, a surveillance system monitoring a parking lot could generate an alarm if it detects a subject approaching a parked vehicle and not entering in it, and repeats the same behaviour with other vehicles. Another suspicious situation where such a system could be useful would be a pedestrian approached by someone who was walking and starts running suddenly.

2. **Video safety:** Similarly to surveillance applications, particular dangerous situations could be detected automatically by a system and react appropriately to prevent the danger. For instance, such a system could help in elderly care by detecting if a subject has fallen into the ground and automatically raising an alarm accordingly. Kindergarten surveillance applications could also benefit

from such technologies by defining a security area where babies can stay, and raise an alarm if a baby is detected outside these premises. In a similar fashion, one could think of an automated monitoring system for swimming pool safety which could help in detecting drowning situations, thus reducing the intervention time. Additionally, a driving assistance system could detect if a driver is falling sleep and warn him to prevent a car accident.

3. **Advanced human-computer interfaces (HCI):** Advanced user interfaces could take into consideration human gestures and actions in order to provide control and command from the user to the computer, or develop machines able to interpret human behaviour, leading to a more natural and intelligent communication between machines and humans. Existing systems with early advanced interfaces may use facilities such as speech recognition, but they could also take advantage of action and gesture recognition by adding context information as a rich cue for understanding human communication, thus helping in solving some of the ambiguities inherent to human natural language. Specific applications include: automatic sign-language translation, gesture-driven control of graphical objects, or signaling in high-noisy environments, among others. Domotics could also benefit from advanced HCI as one could command a series of functionalities using simple gestures.

4. **Automatic content annotation:** Nowadays, with the appearance of a huge number of lightweight integrated cameras into portable devices and the universal availability of high internet bandwidth, the growth of digital media content is becoming unmanageable. Hence, there is a need to make accessible and searchable the huge amount of media generated each day by users, professionals and also the already existing footage. As a result, a lot of effort is required to label, classify and annotate it. Activity recognition algorithms could leverage this effort by automatically annotating the activities detected into the media without human intervention.

1.2.2 Markerless Motion Capture

Motion capture is the process of recording live movement and translating it into usable mathematical terms by tracking a number of key points or regions/segments in space over time and combining them to obtain a three-dimensional representation of the performance [62]. Applied to human actions, motion capture aims to perform full 3D reconstruction and identification of the different body limbs and its movement while performing an action. Notice, that this is not strictly necessary for action recognition, where the body can be treated as a whole without requiring accurate 3D data.

Nowadays, there exist different commercial motion capture systems which are based on different technologies, such as electromagnetic fields, magnetic, ultrasound or mechanical tracking systems among others. However, optical Motion Capture systems provide the higher accuracy and ease of use, and have become extremely popular on the entertainment industry. Such systems use a set of calibrated cameras to digitize different views of a predefined region of interest and usually, a set of easy-to-segment markers are taped or glued to specific points all over the object or actor's

body whose movement will be recorded. As a result, accurate 3D information about the motion performed by the actor is finally obtained.

However, despite all the achievements of optical motion capture systems, they still require a numerous multicamera setup and the placement of markers attached to the subject body. As a result, they are not suitable for non-intrusive scenarios and the setup is complicated and time consuming. Therefore, markerless motion capture based on computer vision techniques would enable new applications and facilitate the usage of the existing ones which currently use optical MoCap systems. In addition, it would also be of interest to use ideally only one camera, or at least a reduced number of them.

Some of the application areas which could benefit from markerless motion capture are:

1. **Sports Biomechanics:** Motion capture is currently used to understand athlete's body motion and enhance its performance and prevent injury. Markerless approaches would ease this process allowing to capture the athlete's motion in its real environment, e.g. during a soccer match, rather than on a special setup.

2. **Medical analysis:** The diagnosis of some diseases which affect body motion could be improved by such systems. In addition, applications for the assisted recovery after injuries or surgery would also benefit of using markerless motion capture.

3. **Arts & Entertainment industry**: Currently, motion capture is being widely used for video-games and film production. The motion performed by an actor is extracted and used to animate virtual characters. Markerless motion capture would extend current applications easing its setup and bringing new possibilities. For instance, MoCap could be applied on existing TV footage and extract motion models for famous dancers. Alternatively, systems using a reduced camera setup for markerless motion capture would make this technology available to independent artists or dancers which could explore body motion in more innovative and creative ways.

4. **Security:** Security devices could benefit from markerless MoCap by detecting body movement and extracting characteristic and unique signatures per each subject linked to his identity. This could be used as an alternative or complement to biometric techniques such as fingerprints or iris scan.

1.2.3 Motion Synthesis

Motion synthesis is a fairly broad term that refers to the automatic creation of animation data. In particular, human motion synthesis aims to animate virtual actors or characters. An important challenge remains in making the subject's motion to appear realistic, smooth and natural-looking. This is not easy to achieve, since human body motion is conditioned to a huge number of constraints very difficult to be determined and modeled, i.e. physics laws such as gravity, constraints on the body topology or the effect of the wind among others. From the scope of human motion analysis, some interesting synthesis applications comprise:

1. **HCI:** More realistic and intuitive user interfaces could be developed using synthetic human models. Hence, the information presented to the user could be rendered in real-time in the form of a human avatar which moves and reacts naturally. Early examples within this application area, comprise the work carried out by Activa Multimèdia on automatic TV content generation, with their SAM system[1]. SAM is a fully automatic system which presents the national weather forecast by means of a synthetic avatar which moves and speaks naturally in a number of different languages.

2. **Virtual Reality:** Accurate visual feedback could be provided to the user by mimicking his motion inside a virtual world. As a result, the overall interactivity of such systems would improve resulting in a much better immersive experience.

3. **Character reanimation:** Visual gesture recognition and markerless motion capture in combination with motion synthesis techniques can be used to recreate the captured scenes for different targets and audiences. For instance, a virtual puppet application for a kids show, could recreate in real-time the body motion and gestures performed by an actor appearing as performed by the main character of a cartoon. Another interesting application would be to generate accurate multiple character 3D reanimation streams from real world sports settings. For instance, the user could watch a recreation of a soccer match from a completely novel perspective by allowing him to choose the viewpoint from where to watch the match.

4. **Video compression:** The required storage or transmission bitrate could be lowered by exploiting body motion information. For instance, the background of the scene could be encoded separately and use higher bitrates on areas of interest of the scene where the motion is occurring.

1.3 Why is it difficult?

Despite its promising applications, recovering the underlying 3D full-body motion parameters by visual analysis of video sequences still constitutes an open problem. The main challenges involved arise from 4 main issues, namely, the 2D-3D projection ambiguities, the occlusions and self-occlusions, changes in the shape and appearance of the human body within the scene and the high-dimensionality and non-linearity of human motion.

In the first place, we are dealing with 2D images which are projections of the original 3D scene given a particular camera configuration. Hence, all the depth information from the scene that originated such images is lost in the projection process leading to **2D-3D ambiguities**. In other words, there are many human body configurations from the real world resulting in the similar 2D projected images, and vice-versa.

This situation is depicted in Figure 1.2, where typical walking 3D body postures are shown with their corresponding 2D silhouettes obtained from different viewpoints. As we may observe, it is very hard to infer a univocal mapping between the 2D

[1]SAM is available under `http://www.activamultimedia.com/sam/`

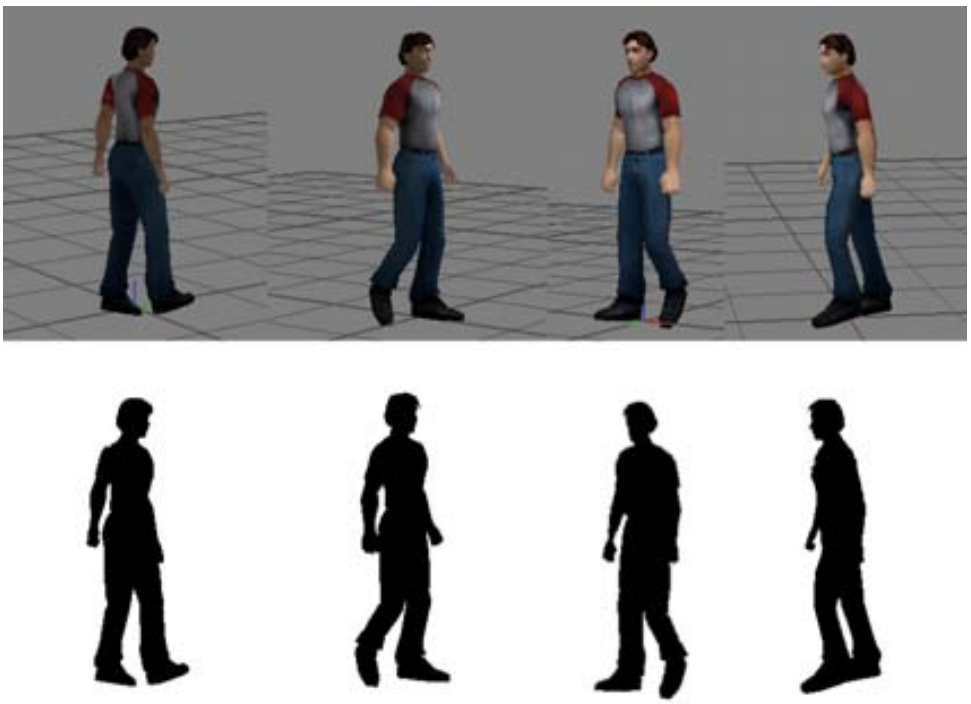


Figure 1.2: Example of 2D-3D ambiguities.

silhouettes and the 3D body postures, since some silhouettes are very similar although they were originated by 4 different postures.

In the second place, most of the times the 2D position of the human body joints are not observable in the images due to **self-occlusions** and **occlusions** with other objects. For instance, depending on the viewpoint used to record the scene, one limb may occlude the other one during a walking or running cycle, or the human body could be temporarily occluded by other subjects or objects located between the subject and the camera. Figure 1.3, shows three subjects suffering both occlusions and self-occlusions in the context of a soccer match[2].

Third, the **shape and appearance** of the human body may also **change drastically** from one subject to another, and even between different frames from the same subject due to changing illumination conditions, shadows, rotations in-depth of body limbs, loosely fitting clothing, and background clutter. This phenomenon is depicted in Figure 1.4, where different frames of the same subject have been taken from a sequence where the subject goes to a vending machine, buys a bottle of water, drinks it and sits down in a table. As one may observe, there are many light sources in the scene, self-rotations and background clutter, causing severe appearance changes as the subject moves across the room.

[2]image courtesy of Mediapro S.L.

Figure 1.3: Example of occlusions and self-occlusions.

Figure 1.4: Changing subject's appearance due to varying lighting conditions, self-rotations and background clutter during a performance of an action.

Finally, another challenging issue is the **non-linearity** of human motion and the **high dimensionality** of human body models. Hence, the human body is composed of many articulated limbs which suffer large accelerations while performing an action.

As a result, even very simple body modeling approximations easily result in very high dimensional models with non linear dynamics.

1.4 Contributions and Thesis Outline

The work presented in this Thesis is all part of the efforts for recovering the 3D human body motion parameters from a monocular image sequence. The full-body tracking problem is faced following a model-based approach, i.e. the process of sequentially estimating the parameters of the target's model given the observations available at each moment. In particular, the estimation process is formulated as a recursive Bayesian filter implemented by a Particle Filter (PF). The Bayesian formulation results in a probabilistic inference framework whose aim is to estimate the *posterior probability density function* (pdf) at each time step over the parameters of a human body model given the evidences (image data) available up to each time step. Hence, the *posterior* pdf represents the sum of knowledge regarding the state of the tracked object from frame to frame.

In general terms, PF tracking approaches are good for keeping multiple hypotheses about the state of the tracked object, and can incorporate a priori knowledge by means of their motion and observation models, which are responsible of propagating the particle set over time and determining the fitness of the particles to the evidences available at each time step. However, full body PF tracking approaches suffer from well known problems related to their discrete nature, and to the fact that it is very difficult to define robust observation models, resulting in drift on the posterior pdf estimate over time. Therefore, a strong motion prior is needed to guide the movement and dispersion of the particle set and avoid tracking failures, which constitutes the framework of the approach presented in this Thesis.

Therefore, this work is aimed to address common drawbacks for full-body 3D tracking using particles filters. This is achieved by using a strong motion prior to improve the robustness and efficiency of this PF framework for monocular 3D full-body tracking of a given set of actions. Towards this end, an action-specific dynamic model of human motion is introduced to avoid particle wastage within the prediction step of the PF. Hence, particles are propagated taking into account their motion history, and previously learnt motion directions from real training data. Next, the state space is constrained by filtering out those body configurations which are not likely according to our motion model. As a result, as long as the truly performed motion lies within the bounds of our motion model, robustness is added to the whole tracker against non-reliable measurements from the image sequence, i.e. in case of occlusions and/or background clutter. In fact, experimental results show that the tracker allows the reconstruction of the 3D motion parameters of a full body stick figure model, using only the 2D positions of a very reduced set of observable joints, namely the head, one hand, and one foot, which constitute a set of joints which are feasible to be automatically extracted from images [85, 51, 46, 47, 74, 65]. Hence, this work assumes an external detection or 2D tracking stage which provides the 2D position of some joints as input data for our tracking scheme.

Regarding the human body model used, human postures are represented by means

of a full body 3D model composed of 12 limbs. Limbs' orientations are represented within the kinematic tree using their direction cosines [87] thus avoiding singularities and abrupt changes due to the representation. Moreover, near configurations of the body limbs account for near positions in our representation at the expense of extra parameters to be included in the model. Principal Component Analysis (PCA) is applied to the training data to perform dimensionality reduction over the highly correlated input data, leading to a coarse-to-fine representation of human motion which relates the precision of the model with its complexity by means of the main modes of gait variation, i.e. the principal components found.

In addition, we introduce an action-specific model of human motion suitable for motion tracking applications. The parameters of this model are learnt from examples of motion-captured data, both from our own dataset and from a publicly available one, i.e. the Carnegie Mellon University's (CMU) Graphics Lab Motion capture database. These datasets were acquired using a commercial optical Motion Capture system which provides very accurate motion data from a set of reflective markers carefully attached on the human body. Given that motion performances have different speeds and durations, a method for synchronizing similar motion sequences has been developed in order to allow comparison between them.

Summarizing, the main contributions of the presented approach are the following:

- A stick figure human body model well-suited for 3D motion capture applications based on PCA and direction cosines representation.
- A method based on Dynamic Programming (DP) for synchronizing pre-recorded motion datasets which have been performed at different speeds by different subjects.
- A probabilistic action model based on examples which captures the variability and temporal evolution of full-body motion within a particular action. Furthermore, the motion model allows to predict feasible 3D body postures given a small motion history of a particular action.
- An improvement of the efficiency and robustness of a PF framework for full-body tracking by introducing a strong motion prior based on the learnt action model.

The outline of this document is organized as follows. Chapter 2 covers the state of the art regarding visual human motion analysis paying special attention to the use of particle filters for 3D full body tracking within the literature. Additionally, an overview on most commonly used human body models is given.

In Chapter 3, we present the body model used in this work to represent 3D human postures. Subsequently, in Chapter 4, the overall action modeling approach is detailed. In particular, we first depict the composition of the motion capture training datasets and introduce the representation space used. Second, the synchronization method applied to the input dataset is detailed. Finally, the model itself and the learning procedure are fully described.

Chapter 5 develops the tracking framework used. Hence, we first review the particle filtering framework for human motion tracking, and then, we detail the use of the learnt action models to improve the overall tracking performance and robustness.

Chapter 6 is related to experimental results. Tests are carried out regarding the dataset synchronization method, the probabilistic matching technique for matching motion subsequences, and the overall tracking approach including the action-specific motion priors.

Lastly, Chapter 7 summarizes the main contributions of this Thesis, outlines the conclusions and discusses some future research lines beyond the scope of this work.

Chapter 2

Related Work

To overcome the issues mentioned in the previous chapter related to full body tracking, many approaches make use of *a priori* knowledge within the estimation process. This knowledge usually comes in the form of a geometrical model of the human body whose parameters are to be estimated given its projections on the 2D images [53]. However, given the complexity of such models, exhaustively searching for the best match within the state-space is not feasible for full-body posture estimation. Therefore, some approaches use machine learning techniques to learn view-dependant mappings between features extracted from the 2D images and the full state space in order to solve the 3D-2D ambiguities [78, 33, 9]. Alternatively, a common approach is to explore only a part of the state-space by temporal filtering, a.k.a. model-based tracking. Model-based tracking approaches aim to sequentially estimate the parameters of a human body model, by comparing the image data from a video stream to *a priori* knowledge about the subject's observable properties, in order to estimate the best fit between the model and the images available at each time instant. Therefore, such approaches are typically composed of a human body model, a human dynamical model, an observation model and a search strategy.

Some well-known examples of model-based human body tracking approaches are given in the following. For example, Wachter and Nagel [82] estimated the motion parameters of a human body model composed of right-elliptical cones assuming a constant velocity model, and using an Iterated Extended Kalman Filter (IEKF) inference framework. They assumed a constant velocity dynamic model and use region and edge information from images in order to match the model to the data. Kalman Filter was also used by Drummond and Cipolla [20] to track the limbs of an articulated human body. They extended Harris' previous work [31] on tracking the 6 DOF pose parameters of a rapid moving rigid body to the articulated body case, by adding global consistency kinematic constraints. Similarly, the work carried out by Sidenbladh et al. [71, 70] formulated the problem as a Bayesian inference task implemented as a Particle Filter (PF). They used strong motion priors based on examples and learnt the probability density function (pdf) of filter responses based on edges and ridges detection over human body limbs. Alternatively, Plänkers and Fua's [63] approach could estimate both the motion and size parameters of the human body by modeling

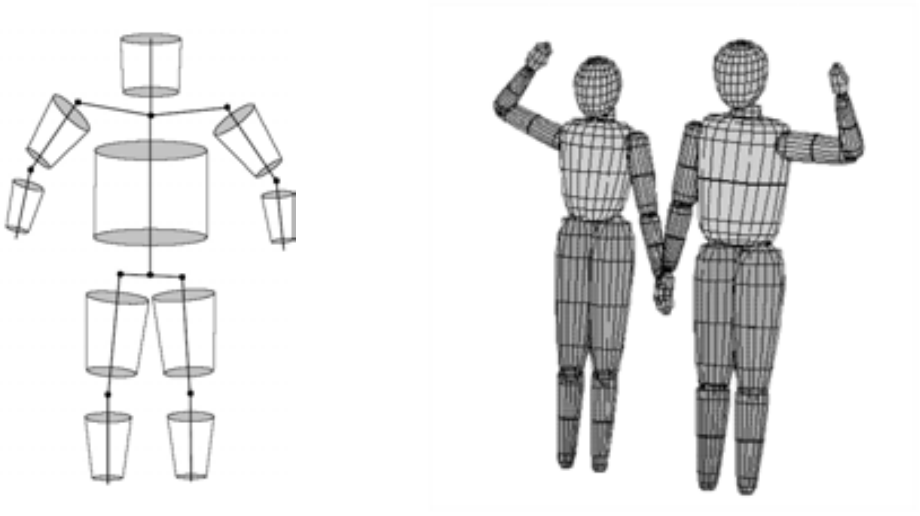
very accurately the shape and appearance of the human body, and the use of an observation model based on binocular disparity maps and silhouettes. Delamarre and Faugeras [16, 17] presented a multi-view human tracking approach based on silhouette contours and 3D volumetric models whose projection onto the image plane was associated with detected 2D contours for improved tracking robustness. Bregler and Malik [12] aimed to track 3D motion at the level of joints by integrating twists and products of exponential maps into region based motion estimation, thus obtaining both image motion and kinematic chain parameters at the same time. Alternatively, Zhao [88] defined the state of the human body as strings, and trained a highly structured motion model similar to a finite state machine for ballet dancing under the minimum description length (MDL) paradigm. It is worth noting that in general, motion models of human dynamics constitute an important part of most tracking processes, being used as *a priori* knowledge to predict motion parameters [88, 15], to interpret and recognize human dynamics [11], or to constrain the estimation of low-level image measurements [70].

The remainder of this chapter is organized into two sections. First, we focus on human body models which are of critical importance within model-based tracking approaches. Hence, the body model is very related to the final applications each tracking approach is facing. As a result, there is a large variety of human body models employed in the tracking literature and thus, we will be reviewing some of them in the following. Second, we discuss on Particle Filtering approaches. PFs [34] have become a very popular model-based scheme for human body tracking because they bring a practical and principled way for estimating non-Gaussian posterior distributions over time, thus multiple hypotheses about the human body posture can be considered and prior knowledge about the human dynamics can be easily integrated into the tracking process. Therefore, we will end this section reviewing the use of particle filtering applied to visual motion analysis paying special attention to the use of human motion priors.

2.1 Human Body Models

Models for representing the human body vary widely in the levels of details. Stick figure models have been frequently used in the literature [27, 45] for motion tracking due to their simplicity and to its ability to represent the relative position and angles between the joints and limbs composing the human body. In such models, the human body is assumed to be a rigid articulated body, where body parts are represented by sticks connected by joints with up to 3 degrees of freedom (DOF). The number of joints, limbs and DOF vary from work to work. For instance in [71] the authors use an articulated body model with 50 DOF. Similarly, 47 DOF are used in [72] to represent all the possible body configurations of a 10 limbed stick figure model, with 9 joints overall, including the fixed parameters of a volumetric model consisting in a set of tapered cones.

The stick figure representation follows a topological organization, i.e. limbs and joints are organized in a hierarchical manner, which enables to express one joint's position relative to another. For instance, one hand's position can be expressed only



(a) (b)

Figure 2.1: From Gavrila et al [24]. (a) Typical truncated cone 3D volumetric model used for 3D human model-based tracking. (b) "ELLEN" and "DARIU": A 3D human model using super-quadrics.

relative to the elbow. Thus, moving only the torso doesn't influence on the relative position of the hands. Notice that this is a very desirable property, since it allows the analysis of motion for specific body parts regardless the rest of the body which might not be interesting for analyzing local motion patterns. This full-body tree-like layout is known in robotics as a *kinematic chain* [69].

Additionally, the stick figure model can be "augmented" by using some kind of 3D volumetric primitives to depict the body shape in more detail [82, 16, 17]. Fig. 2.1 shows two volumetric models that have been used for 3D modeling. In Fig. 2.1.(a) a model consisting of 3D truncated cones is shown. The volume and shape of each limb can be approximated by the conical sections, so that information could be used to predict the region of a limb when performing the matching between the model and the image. More accurate modeling can be achieved by using super-quadrics, see Fig. 2.1.(b). The model was used by Gavrila et al in [24] and consists of generalizations of ellipsoids which have additional "squareness" parameters along each axis. Moreover, they support global deformations such as tapering, bending, etc. Compared to truncated cones, super-quadrics provide a wider scope for better fitting the model to less cylindrical body parts bringing a reasonable degree of accuracy and flexibility at expenses of more parameters to estimate.

In [16, 17] a 3D human model with 20 degrees of freedom is built using truncated cones, spheres and parallelepipeds as shown on Fig. 2.2. Mikic et al. [52] used an articulated human body model for tracking consisting of ellipsoids and cylinders combined with the reconstruction of 3D voxels of the human body.

Alternatively, Korc and Hlavac [41] represented human silhouettes using six-link

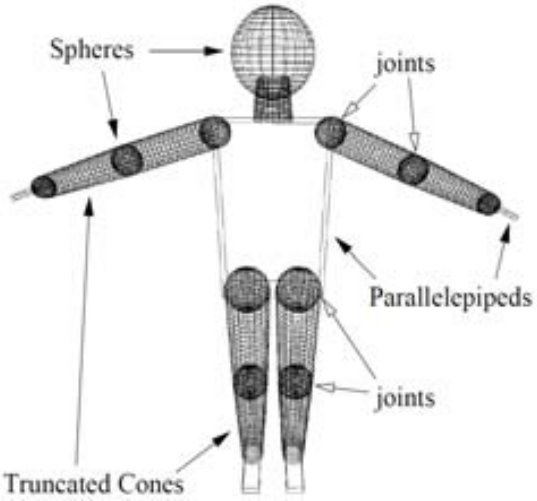
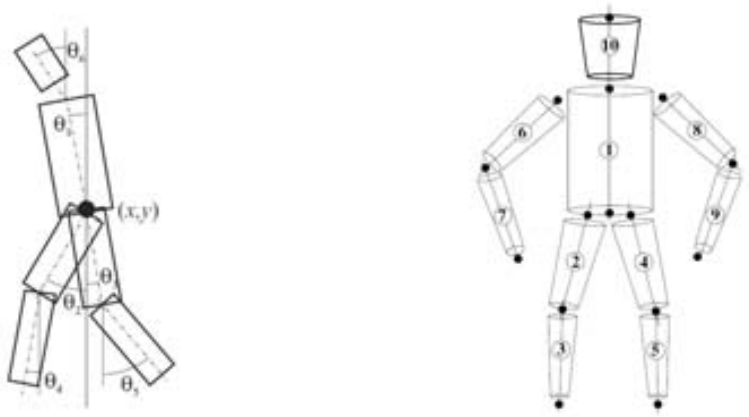


Figure 2.2: A volumetric 3D human model from Delamarre and Faugeras [16, 17].



(a) (b)

Figure 2.3: (a) Six-link biped human body model employed in [41]. (b) Sigal's et al. representation of the human body consisting of a 2D trapezoid for each body part [74].

biped model (see Fig. 2.3.(a)), in which principal body parts such as head, torso and legs are represented by rectangles for detection and tracking of humans from monocular videos with dynamic background. The model is represented by a parameter set consisting of the center of the body, angles between the body parts and a vertical line and relative sizes of body parts.

Sigal et al. [74] model the human body as ten rigid parts consisting of 2D trape-

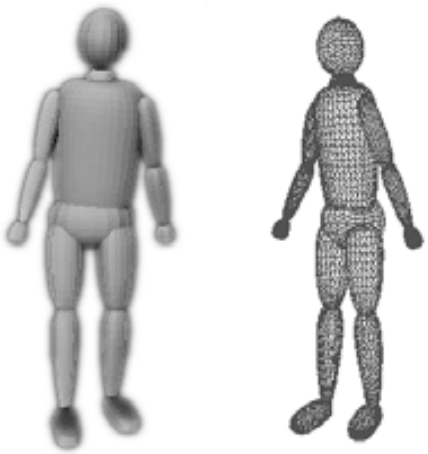


Figure 2.4: Stick figure model fleshed out by superquadric ellipsoids with additional tapering and bending parameters used in [76].

zoids with 5 DOF each (see Fig. 2.3.(b)). Then, probabilistic constraints are defined between the parts resulting in a total of 50 parameters. A more realistic model is used by Sminchisescu et al. [76] as shown in Fig. 2.4. It consists of a stick figure model fleshed out by superquadric ellipsoids with additional tapering and bending parameters. The model has 30 joint parameters plus 8 parameters controlling the internal limb proportions and 9 additional deformation parameters per each body part.

In general, one would expect that the more complex 3D volumetric models, the better results may be expected due to an improvement in the matching step of the model-based tracker. However, although more complex models are specially useful for synthesis purposes, for tracking applications the number of parameters to be estimated grows up since the primitives used need to be characterized. Therefore, a "raw" stick figure is often preferred to characterize human postures instead.

2.2 Particle filters and motion priors

PFs constitute a powerful tool for representing and dealing with complex posterior distributions. The key idea is to represent the posterior distribution function (pdf) over the state-space by a discrete set of weighted samples or particles, and propagate this distribution over time by means of a dynamic and observation model. One of the first approaches to use particle filtering for visual motion estimation was CONDENSATION [34]. However, its initial formulation presented several misbehaviours which caused the filter to unavoidably loose the target over time. Indeed, particle filters suffer from many well known problems related to its discrete nature [40], and to the fact that a good model of the system dynamics is required in order to represent the posterior pdf properly. Additionally, although it has been shown that there exists an upper bound for the number of particles to achieve a certain estimation error [54], there exists an exponential relationship between the number of required

particles and the dimensionality of the state space for being properly populated [50]. This becomes critical for 3D human motion tracking where the models employed are usually high-dimensional, and dynamics are highly non-linear.

In general terms, there are two main methodologies to improve PFs: modifying the algorithm itself to prevent particle wastage, or designing better dynamic and observation models for the tracked objects. The first ones control the particle quality by directly modifying them. For instance, the kernel-based PF [30] faces the sample impoverishment issue by approximating the likelihood and the posterior densities by a Mixture of Gaussians (MoG) at each time step and using them as proposal distributions for sampling new particles. An hybrid search strategy was introduced by Sminchisescu et al., which combines the global particle representation of the posterior with deterministic search for local optimization [75]. Then, in [22] authors make the particle set more efficient, by adapting its size during the estimation process. Hence, they approximate the estimation error due to the sample-based representation by the Kullbach-Leibler distance. Then, the key idea is to bound the error by increasing the number of particles when the uncertainty on the state space is high. Finally, some approaches aim to lower the searching complexity by assuming that the state space can be decomposed. For example, in [25] they perform a hierarchical search of the body configuration, or in [50] they use partitioned sampling to build independent observation densities over each dimension of the state space. Alternatively, Deutscher's et al. annealed PF algorithm [18] replaces the likelihood pdf by a fitness function which measures the quality of a particle relative to an observation. The whole posterior pdf is no longer propagated, thus requiring less costly likelihood evaluations. Additionally, a searching technique is presented which introduces the influence of narrow peaks in the fitness function, gradually. As a result, particles are guided efficiently to a global maximum of the fitness function. See [53] for additional examples.

Alternatively, PFs' efficiency can be improved by designing better observation models (likelihood) and dynamic models (prior) for the tracked objects. Observation models evaluate the fitness of predicted postures to the measurements available. They must deal with severe illumination and viewpoint changes, and most of the times only a few set of body joints may be observable from images. Although observation models have been intensively studied [70, 82, 23, 2, 6, 35], it still constitutes an open problem in 3D full-body tracking from a monocular image sequence. In general, it is very difficult to design robust likelihood functions. As a result, the update of the predictions may not be reliable for a certain period of time due to weak and noisy measurements. Therefore, strong motion priors are also needed to avoid tracking failures.

Indeed, many action recognition and human body tracking works rely on proper models of human motion which usually are learnt from training datasets of real prerecorded motions [57, 71, 28, 80]. However, training sequences are often acquired under very different conditions making it difficult to extract useful information from these training sets. Therefore, methods for synchronizing the motion training sets are often required so that a mapping between postures from different sequences can be established. For example, Ning et al. proposed a method for normalizing the length of cyclic walking sequences using a self-correlation measure [57]. As a result, the training walking cycles are rescaled to last the same period of time and are aligned to

the same phase. However, self-correlation is only suitable for cyclic motion sequences. In [56] a variation of Dynamic Programming (DP) is used to match motion sequences acquired from a motion capture system, but the overall approach is aimed to the optimization of a posterior key-frame search algorithm. The output from this process was used for synthesizing realistic human motion by blending the training set. They divided the body in 4 portions, and similarities are evaluated independently for each part.

Applied to particle filters, human motion priors can be exploited to guide the exploration of the state space and propagate particles efficiently to areas of interest. For instance, Sidenbladh et al. sample new body postures from a database of pre-recorded motions [71]. They used the well-known multivariate principal component analysis (MPCA) method to train a walking model based on the examples. Then, given a motion subsequence they probabilistically searched the best match within the database and predictions were made assuming a Gaussian distribution over the learnt motions. Although they achieved good tracking results, the model could only predict postures which were present in the motion database. Likewise, Ong and Gong [59] employed the hierarchical PCA to learn their motion model that was represented by the matrices of transition probabilities between different subspaces in a global eigenspace and by the matrix of that between global eigenspaces.

Alternatively, Ning. et al [57] tracked a 12 DOF body model of a walking sequence using a PF and a dynamic model of walking including constraints formulated as independent Gaussian distributions per each joint. Chai et al. [14] presented a learning scheme for large motion sets (about 1 hour) to reconstruct 3D motion in real-time from a few 2D control signals. The system learnt a series of local linear models for the on-line mapping with the control signals. The reconstructed motion basically consisted in the retrieved postures from the database which best matched the 2D tracked markers, although it had some ability to interpolate new motions not explicitly present in the training set.

Recently, Urtasun et al. [81] introduced the use of a Gaussian Process Dynamical Model (GPDM) to learn 3D posture and motion priors for 3D human tracking from a small training set. They successfully tested their priors using the 2D position in images of some 3D joints. However, only lateral walking sequences were tested, and instead of a PF, they used an offline inference approach to obtain the MAP estimate over a time window including past and future events. Similarly to this work, Wu et al. [86] learnt a model of feasible hand postures from a real motion database represented in a PCA space, which was used as the importance function in a PF framework for articulated hand tracking. They defined a set of basis hand configurations based on its topology and observed that most natural hand motion can be constrained by the set of linear manifolds spanned by any two basis. However, it is not clear how to define such set of basis for human body postures, and linear manifolds seem to be too restrictive to accommodate natural body motion. Another work using a PCA space to constrain articulated hand motion was carried out by Stenger et al. [77]. They use a tree-based grid filter, partitioning the PCA state space by clustering motion capture data. Then, dynamics are modeled as a first order process with learnt transition probabilities between the states. In addition, a likelihood function based on edge and color information is defined, and initialization of the tracking process

is handled by combining hierarchical detection and Bayesian filtering. Recently, this work has been successfully extended to body tracking in a controlled environment [58] by introducing body shape information for the likelihood computation. However, the method strongly relies in the good behaviour of the likelihood function, since limbs' dynamics are modeled using a simple first order motion model.

Chapter 3

Human Body Modeling

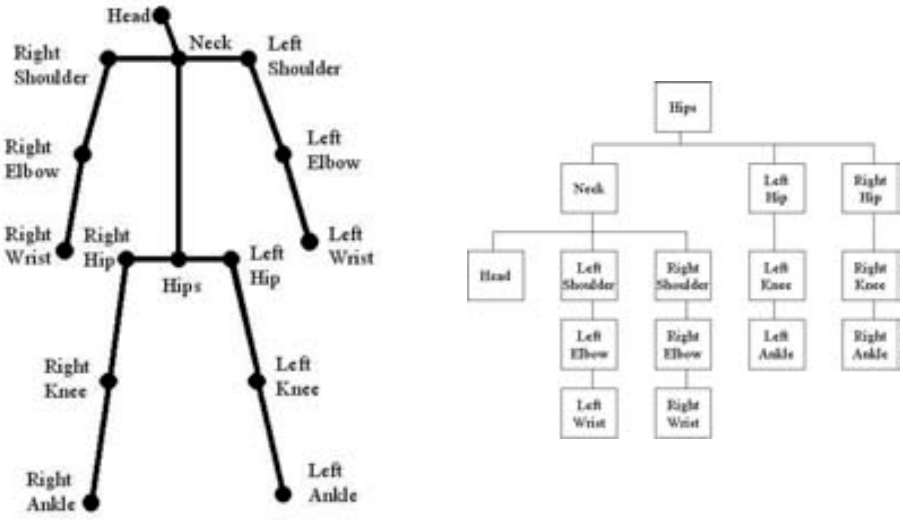
The choice of the human body model to represent 3D human body postures is of critical importance within a model-based tracking approach, since it conditions the choice, design and performance of all underlying methods. Thus, it is desirable to choose models having a good trade off between low complexity and high-generality so they can represent the state of the human body accurately enough while keeping the number of parameters to be tracked low.

Bearing this in mind, the body model employed in this work consists of a stick figure model comprising twelve rigid body parts connected by a total of ten inner joints. Body limbs are structured in a hierarchical manner and the self rotation around its axis is not modeled. Direction cosines are chosen for characterizing 3D directions to avoid discontinuities in the representation and because they have a direct geometric interpretation.

In this chapter, we first motivate the selection of a proper body model with respect to its final application by reviewing common human body models employed in the literature and exploring their advantages and drawbacks within the context of full-body tracking. Then, we discuss different options to represent 3D orientations and their suitability for the tracking problem. Lastly, we fully specify this work's human body model and elaborate on how to convert from Motion Captured data to our representation.

3.1 Motivation

The selection of a proper human model is highly dependant on the final application of it. Typically, one has to carefully choose an appropriate trade-off between complexity, generality and compactness of the model, since there is not a unique solution suitable for all scenarios. In the case of full-body 3D tracking, on the one hand, we need a model which represents well the motion parameters of the full body rather than its accurate surface and texture. On the other hand, facilities for synthesis must be taken into account, and no singularities in the evolution of the parameters are wished due to the model-based tracking approach. Finally, the model has to be easy to manipulate and computationally treatable, since models with high dimensionality often result in



(a) (b)

Figure 3.1: Stick figure Body Model and its Hierarchy. **(a)** Generic human body model composed of twelve limbs and fifteen joints. **(b)** Hierarchy of the joints of the human body model (kinematic chain).

too much parameters to be estimated by the tracking framework, leading to tracking failures and unaffordable computational costs.

Towards this end, stick figure models have been frequently used in the literature [27, 45] because they are simple and well suited for representing body motion. Indeed, such models consists of a predefined set of limbs of a given length connected by a set of joints with a variable number of DOF. The number of joints, limbs and DOF vary from work to work, ranging from about 50 DOF in [71, 72] or to 12 in [57], for instance. In Fig. 3.1 it is shown the stick figure model used in [27]. It is composed of 12 limbs and 15 joints organized in a hierarchical manner.

Stick figure models can be in turn extended with more complex volumetric models, such as cylinder [32, 82, 66], truncated cones [17] or super-quadrics [75, 67]. Additionally, accurate shape and appearance of the human body can be modeled by adding further layers to the core stick figure model. For instance, a complete human body model was presented by Plankers and Fua [63] consisting in an accurate hierarchical human body model, which included four levels: skeleton, ellipsoid meatballs simulating tissues and fats, polygonal surface representing skin, and shaded rendering. As a general rule, the more complex the human body model, the more accurate tracking results may be expected, since give they give better results for synthesis so the model can be better compared to the image data. On the other hand, simpler models require less parameters to be estimated resulting in less computational complexity.

Therefore, for full body motion capture applications, we aim at a human body model with great representational power in terms of motion parameters accuracy,

with synthesis facilities, exempt of singularities, and compact enough to make the tracking problem computationally treatable.

3.2 Representing Orientations in 3D

A very naive approach to model the placement of each body part using stick figures, could be to decide which body parts we want to model, and then simply store its raw 3D positions. With this representation, each human posture could be modeled as an ordered tuple of 3D end-point coordinates of the limbs composing the model, where each limb l would be defined by its end-points (x_i, y_i, z_i) and (x_j, y_j, z_j).

Even though such an approach is very simple to compute and has a direct geometric interpretation, it suffers from a lot of inconveniences. From the one hand, given the fact that the length of the limbs varies from one subject to another, the 3D positions also vary between several performances of the same action. On the other hand, it does not reflect the actual topology of the human body in a treatable manner, since each position does not keep any relationship to each other. Moreover, Cartesian coordinates are not ideal when considering linear approaches for posture variation modeling. Summing up, this approach is very useless since it requires a lot of post-processing for human motion analysis.

This fact is illustrated in Fig. 3.2. The evolution of the 3D positions of a set of four predefined body parts -i.e. both shoulders, left knee and left foot- are shown while performing a bending action. Each column describes a different body part, consisting of three subplots which represent the X, Y and Z Cartesian coordinates of that particular part. Lines in different colors correspond to different actors performing the same bending action. One can observe that all the performances vary a lot from one actor to another, making this model totally useless since establishing a similarity measure between them could be a tough task. Thus, we need to formulate the problem in a more efficient and robust manner.

Alternatively, we can model the 3D configuration of the body by computing the angle between adjacent limbs, thus achieving independence from the size of the actors. As a result, using the stick figure representation, a particular human body posture will be defined by (a) its global **location and orientation** in the space, (b) the **length** of each body limb, and (c) the relative **limbs orientation**, i.e. each limb position is expressed in a relative manner, following the kinematic chain. Final positions of the limbs will depend on their position in the kinematic chain, their length and their relative orientation from others.

Unfortunately, the choice of the angles to model is not direct. There are several problems involved with measuring angles that should be confronted. In the first place, the angle discontinuity problem occurs when a limb returns to the same position after each revolution of orientation. For example, we show in Fig. 3.3.(a) the orientation values corresponding to the left wrist during a bending performance using a spherical coordinates system. During this performance, the left arm hangs performing a circular swing. Despite of the fact that elevation values are stabilized near $-\frac{\pi}{2}$, orientation values *jump* between $-\pi$ and π [1]. This discontinuity in the angle-time curve is not

[1] We consider a bipolar range for angle values, i.e. a range of $[-\pi, \pi]$.

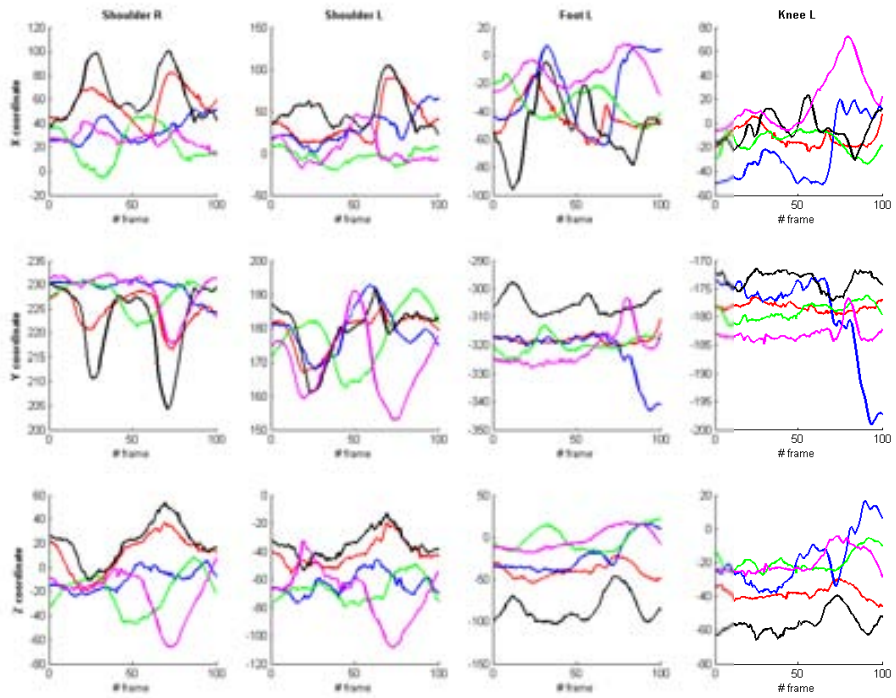
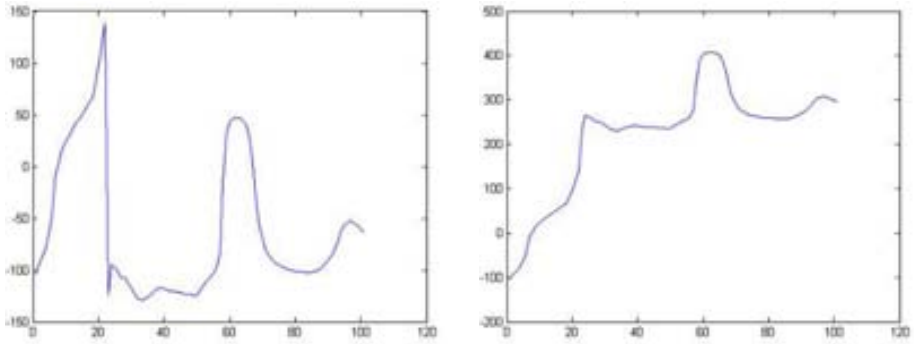


Figure 3.2: Evolution of the $(X\ Y\ Z)$ Cartesian coordinates of 4 body parts (both shoulders, left knee and foot) over 100 frames of the bending action performed by 2 different actors which were recorded using a commercial Motion Capture system. Each column corresponds to a different body part, while each row represents a different Cartesian coordinate.

acceptable because it can cause problems in the computation of a human action model.

On the other hand, the range of the angles should be limited, since angle calculations are a bijective function modulus 2ϕ. Fig. 3.2(b) shows the resulting angle values by avoiding discontinuity jumps. However, note that the resulting continuous angle values are not restricted to lie between any determined range: a specific limb orientation can be mapped to several angle values $\ldots \phi, 2\phi, 4\phi, \ldots$. As the aim is to model angle variations over time, the range of training angle values should also be limited.

In summary, there's not a unique way for representing three-dimensional orientations that fulfills all possible requirements for any models. Therefore, we will describe most several popular procedures for describing orientations of an object in the space, and their drawbacks and advantages will be discussed.



(a) (b)

Figure 3.3: Orientation values per frame during a bending performance. (a) As angle values lie between the range of $[-\pi, \pi]$, the angle discontinuity problem should be confronted. (b) Resulting angle values by avoiding discontinuity jumps.

Rotation Matrices

With this method, we can define the rotation of a local system with regard to a global system by means of a 3×3 matrix. A rotation in a three-dimensional space requires a representation that provides at least 3 DOF. However using 3×3 matrices, result in a total of 9 parameters to be set. Indeed, a 3×3 matrix is suitable to perform more transformations than only 3D rotations. Subsequently, we need to introduce some constraints on its form, thus reducing the number of DOF to 3. In fact, only *orthogonal* matrices are valid rotation matrices. Formally, an orthogonal matrix $\mathbf{A}$ accomplishes that

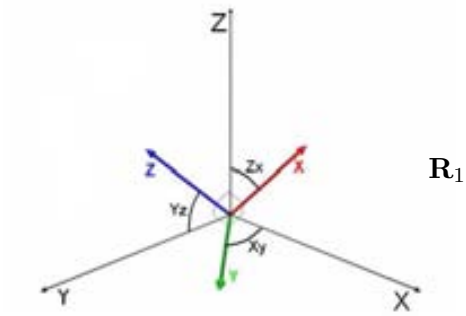
$$\mathbf{A} \cdot \mathbf{A}^T = \mathbf{I}.$$

Rotation of the local system with regard to the global system is represented as follows: the axes (x, y, z) of the local system are represented by their components in the global reference system. Due to the definition of the dot product, dividing each component by the vector length (which is equal to 1) results in the cosine of the angle that the vector makes with each of the coordinate axes (X, Y, Z) of the global system. These angles are the *direction angles* or *direction cosines*. Indeed, the direction cosines written in matrix form as elements of a 3×3 matrix conform the so-called rotation matrix. Thereby, we define a rotation matrix $\mathbf{R}$ as:

$$\mathbf{R} = \begin{bmatrix} \cos_{Xx} & \cos_{Xy} & \cos_{Xz} \\ \cos_{Yx} & \cos_{Yy} & \cos_{Yz} \\ \cos_{Zx} & \cos_{Zy} & \cos_{Zz} \end{bmatrix}.$$

In this notation, $\cos_{ab}$ refers to the cosine formed between the axis a of the global system and the axis b of the local system.

Fig. 3.4 shows the rotation described by the matrix $\mathbf{R_1}$ of a local coordinate



$$\mathbf{R}_1 = \begin{bmatrix} 0.6092 & -0.3137 & 0.7283 \\ 0.6292 & 0.7502 & -0.2032 \\ -0.4826 & 0.5821 & 0.6544 \end{bmatrix}$$

Figure 3.4: Rotation of a local coordinate system (x, y, z) in reference to the global system (X, Y, Z) described by the rotation matrix $\mathbf{R}_1$.

system (x, y, z) in reference to the global system (X, Y, Z). Notice that each row corresponds to the components defined in the global reference system of each vector from the local system. Notice also that the three vectors are orthogonal with norm equal to 1.

A very interesting property from rotation matrices is the fact that by multiplying several rotation matrices, we can concatenate several rotations grouping them into a single rotation matrix. Since matrix multiplication is not commutative, the order of the multiplications will define the order in which we perform the rotations.

On the other hand, rotation matrices require to specify 9 values to represent a single 3D orientation, that requires only 3 DOF. Moreover, it is not easy to establish a geometrical interpretation on the parameters of this representation, and interpolating between the parameters of two rotations doesn't result in uniform soft final transitions. For all these reasons, rotation matrices happen to be a very useful tool for making calculations, but a very tedious way of describing orientations for applications where a compact, easy-to-handle representation is needed.

Euler angles

Euler angles have been widely used in order to describe orientations of the body limbs [82, 17, 57]. It consists on concatenating three successive rotations around preset axes. Notice that finite rotations in 3D space are not commutative, i.e. performing the same rotations in a different order leads to different final orientations. Thereby, there exists a set of 12 different conventions for describing Euler angles, depending on the order of the rotations and the axes used for each rotation. The general process goes as follows:

1. The first rotation is defined relative to an axis of the global reference frame, i.e. one of the Cartesian axis X, Y, Z.

2. Then the second rotation is defined with regard to a local axis from the previous

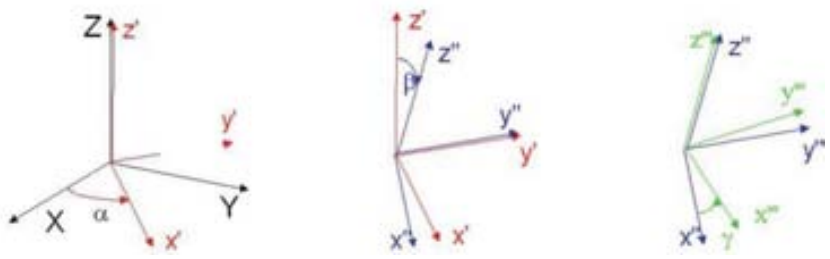


Figure 3.5: Rotation of (α, β, γ) described using the $Zy'z"$ Euler convention.

rotation. We denote this axis as x', y', z'.

3. The final rotation is done again around the local axis transformed from the previous rotation denoted as $x"y"z"$.

Notice that axes are notated with single prime ($'$) or double prime ($"$) according to the number of preceding rotations. Using this notation we can enumerate the 12 conventions about Euler angles depending on the order of the axis used. For example, a sequence of rotations $Xy'z"$ means that the first rotation is done around the Cartesian axis X, then around the local axis y' resulting from the previous rotation, and finally the third rotation is done around the $z"$ axis which was previously rotated around the global X axis and then around the local y'axis.

Fig. 3.5 shows a rotation of (α, β, γ) described using the $Zy'z"$ Euler convention. Starting at the left frame, with Cartesian axes X, Y, and Z, we rotate through a positive (counter clockwise) angle α to arrive in the middle frame having Cartesian axes x', y', z'. Next we rotate by a positive angle β around the y' axis and we arrive in the frame on the right, having Cartesian axes $x", y", z"$. Finally, we rotate by a positive angle γ around the $z"$ axis obtaining the final frame rotation with axes $x"', y"', z"'$.

The main advantages of expressing orientations as Euler angles are their simplicity and easy geometrical interpretation. Moreover Euler rotation can be easily formulated as rotation matrices, thus several Euler rotation can be concatenated by only multiplying their rotation matrices which is a desirable property in a kinematic chain, where local position and orientations of each limb depends on the previous limb in the kinematic chain. However, it suffers from some implicit problems:

- Problems with periodicity: when rotating an object around a particular axis, every time a full turn is completed there exists a leap between 0^{o} and 360^{o}. This makes this representation a bad candidate to be used for calculations that demand a certain continuity on the input data.

- Singularities: in some particular cases, the value of one or more coordinates cannot be defined for certain angular positions. A DOF is lost due to the fact that a rotation axis gets aligned with another one after performing several rotations of 90^{o}. This problem is known as *gimbal lock* [84], and is partly due to the fact that the same orientation can be described in 12 different manners.

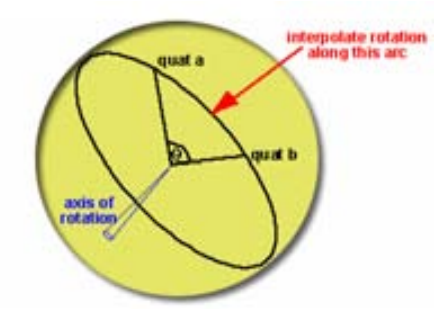


Figure 3.6: SLERP (Spherical Linear intERPolation) interpolation using quaternions.

- No interpolation facilities: it is very difficult to softly interpolate 2 orientations defined with this representation method.

Quaternions

Quaternions appear as an extension of the complex numbers by adding to a real number not 2 but 3 imaginary parts (i, j, k), which satisfy the following relation:

$$i^2 = j^2 = k^2 = ijk = -1.$$

Every quaternion is a number q which is a linear combination of the so-called quaternion units, i.e. $i, j, k, 1$, Thus, a quaternion q has the form $q = a + bi + cj + dk$, where the vector of real numbers (a, b, c, d) define a particular quaternion. Mathematical operations with quaternions are defined by special rules. Quaternions are not restricted to representing orientations, and its use goes much further than this. However, we are interested in the quaternions able to represent orientations, i.e. the ones whose vector (a, b, c, d) has norm equal to 1. Formally, quaternions suitable to represent orientations fulfill the condition:

$$a^2 \cdot b^2 \cdot c^2 \cdot d^2 = 1.$$

An interesting property of quaternions is that the quaternion q_t resulting from two consecutive rotations q_1, q_2 is the product $q_t = q_1 q_2$. Unlike it occurs in real or complex numbers, multiplications between quaternions are not commutative, thus when concatenating rotations using quaternions we must take into account the order of the rotations.

Interpolating rotations using quaternions can be done with very simple operations and results in soft transitions between two (or more) orientations. If one thinks of a quaternion as a point in a sphere, the SLERP (Spherical Linear intERPolation) interpolation allows us to go from point a to point b of the sphere going along the shortest arc between both points [64], see Fig. 3.6.

Using quaternions, SLERP interpolation is defined by the equation

$$SLERP(q_a, q_b, t) = \frac{q_a \sin((1-t)\phi) + q_b \sin(t\phi)}{sin(\phi)},$$

where q_a and q_b are the two quaternions that define two particular orientations (positions on the sphere), t is a value between 0 and 1 that defines the interpolation step, and ϕ is the angle between a and b radius.

Besides the interpolation facilities, using quaternions for representing orientations doesn't suffer from singularities problems (unlike Euler's angles), and is a more compact and faster representation than matrices. However, we require 4 values to describe 3 DOF, and direct geometrical interpretation of the parameters is not straightforward.

Direction cosines

Another interesting method for describing the direction of a vector w.r.t. a reference system is by means of their direction cosines. As pointed out before, the direction cosines of a vector $\mathbf{l}$ are the cosines of the angles that the vector makes with each of the coordinate axes (X, Y, Z) of the reference system, i.e. the direction angles of the vector.

In particular, given the vector $\mathbf{l} = (l_x, l_y, l_z)$ depicted in Fig. 3.7, its direction cosines $(\cos\theta_l^x, \cos\theta_l^y, \cos\theta_l^z)$ are computed as follows:

$$\begin{aligned}
\cos\theta_l^x &= \frac{l_x}{\sqrt{l_x^2 + l_y^2 + l_z^2}}, \\
\cos\theta_l^y &= \frac{l_y}{\sqrt{l_x^2 + l_y^2 + l_z^2}}, \\
\cos\theta_l^z &= \frac{l_z}{\sqrt{l_x^2 + l_y^2 + l_z^2}}.
\end{aligned} \tag{3.1}$$

From these definitions, it follows that

$$\cos^2\theta_l^x + \cos^2\theta_l^y + \cos^2\theta_l^z = 1. \tag{3.2}$$

From the above equations its easy to see that if $\mathbf{l}$ is a unit vector, then the direction cosines of $\mathbf{l}$ are equivalent to the coordinates (l_x, l_y, l_z) of the vector. Therefore, direction cosines are easy to calculate and have a direct and intuitive geometric interpretation.

Also note that while being particularly suitable for representing the direction of a given vector, they are not independent of each other since they are related by Eq.(3.2). Thus, direction cosines only have 2 DOF and can only represent direction of a vector but not its orientation. In other words, direction cosines do not define how much a vector is rotated around its axis, and require 3 parameters to determine only 2 DOF.

On the contrary, unlike polar angles, they are defined anywhere in the unit sphere

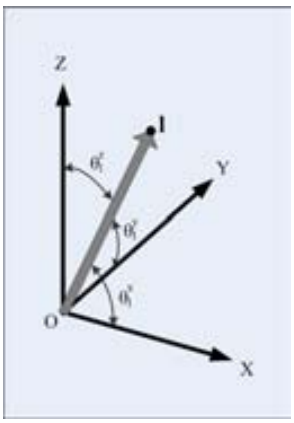


Figure 3.7: Direction angles $(\theta_l^x, \theta_l^y, \theta_l^z)$ for the limb l.

while being continuous and stable, i.e. they are exempt of periodicity problems and singularities present in Euler angles. Therefore, smooth rotations of a vector result in smooth and continuous changes on their direction cosines. Hence, direction cosines are useful for a wide range of applications where a stable, smooth and continuous representation is needed, for instance 3D motion compression, statistics over 3D rotations, 3D motion tracking, etc.

Summarizing, direction cosines are particularly interesting for representing body limbs' direction in a body motion tracking context because they enable a representation exempt of discontinuities with a direct geometric interpretation and easily treatable.

3.3 The Human Body Model

For full body motion tracking we want to characterize where each part of the body is placed and which relationships keep with the other parts, rather than attempting to represent very accurate information about the shape of each limb, the color of the skin, or the clothing, among others. In turn, we need a body model with great representational power in terms of motion parameters accuracy, with synthesis facilities, exempt of singularities, and compact enough to make the tracking problem computationally treatable. In other words, we don't need very realistic and natural looking models as long as we keep the body's geometric structure well enough to support segmentation and matching.

Towards this end, the stick figure model is used to approximate the human body as a rigid articulated object, where each limb is modeled as a segment of fixed length. Adjacent segments are connected between each other by means of a joint or articulation.

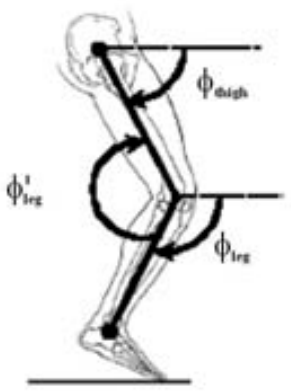


Figure 3.8: Relative Angles: Leg angle ϕ'_{leg} is defined in terms of its angle w.r.t. its parent limb within the kinematic tree chain of the human body model, i.e. relative to the tight.

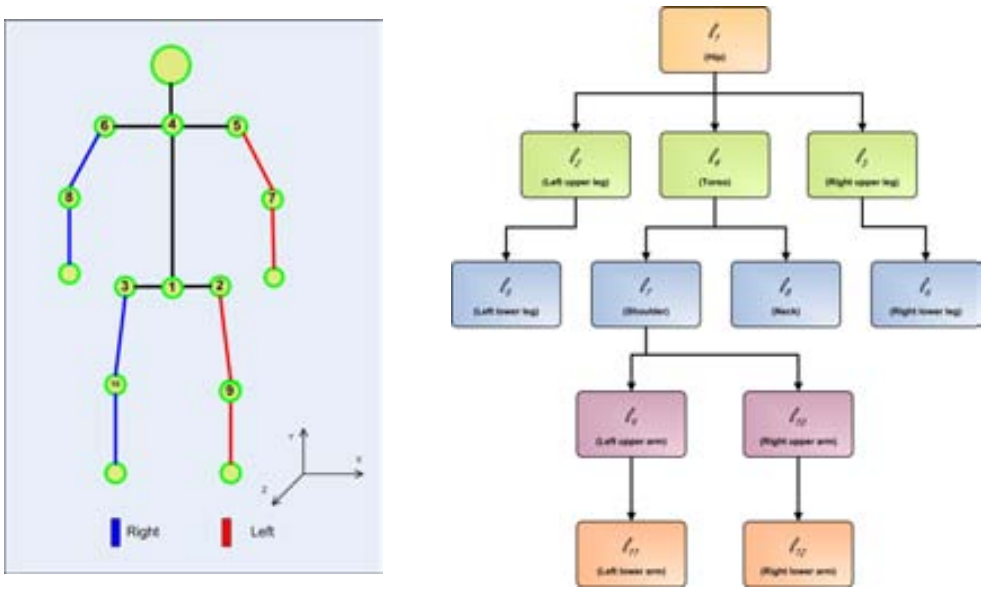
In addition, notice that human actions are constrained motion patterns, i.e. there is a strong relationship between the movement of different limbs while performing a particular action. For instance, during a walking and running action, legs and arms move in a coordinated manner alternating left and right swings of both legs and arms. Consequently, we consider a hierarchy between limbs in order to incorporate these relationships into the model by describing each limb with respect to a parent, i.e. constituting a kinematic tree. Usually, the root of this tree is located at the hip. Consequently, by describing the the human body using the relative angles of the limbs, we actually model the body as a hierarchical and articulated figure. This is illustrated in Fig. 3.8 where the 2D angle of a leg is expressed relative to the upper limb in its hierarchy, i.e. the tight. Hence, ϕ_{tight} and ϕ_{leg} correspond to the absolute angles of both limbs w.r.t. the Cartesian axis, while ϕ'_{leg} is the relative leg angle within the kinematic tree.

Therefore, the body model employed in our work consist of a stick figure model composed of twelve rigid body parts (hip, torso, shoulder, neck, two thighs, two legs, two arms and two forearms) connected by a total of ten inner joints, see Fig. 3.9.(a). Limbs' direction is modeled using 2 DOF, without modeling self rotation of limbs around its axis. The body segments are structured in a hierarchical manner, constituting a kinematic tree as shown in Fig. 3.9.(b). The root, located at the hip, determines the global rotation of the whole body. Notice that body's global position is not considered in the model. Direction cosines are used to represent limbs' direction within the kinematic tree.

As a result, our final representation of a human body posture ψ consists of 36 variable parameters, i.e.

$$\psi = \{\cos\theta_1^x, \cos\theta_1^y, \cos\theta_1^z, ..., \cos\theta_{12}^x, \cos\theta_{12}^y, \cos\theta_{12}^z\}, \tag{3.3}$$

where $\theta_l^x, \theta_l^y, \theta_l^z$ are the relative directional angles for the limb $\mathbf{l}$ according to the



(a) (b)

Figure 3.9: (a) Details of the human body model used. (b) Hierarchy of the limbs of the human body model (kinematic chain).

topology shown in Fig. 3.9. In addition, the global position and orientation of the body as well as the length of each limb are also needed to derive the final 3D position of a joint given a particular body configuration ψ. Nevertheless, each limb length is assumed to be fixed for each subject as it doesn't change significantly along the performance of a particular action.

Consequently, the characteristics of our human body model can be summarized as follows:

- Simplicity and compactness of the representation: the human body is modeled using a stick figure model with a reduced number of parameters.
- High-level interpretation of the parameters, the anatomical characteristics and topology of the human body is reflected by means of the kinematic tree.
- Depth information can be determined. The 3D position of each limb and joint can be determined. In addition, it can be extended by putting it in correspondence with a more realistic and complex models by fleshing it out with volumetric primitives.
- Constraints on human motion can be easily specified, i.e. angles ranges can be constrained for individual joints, thus simplifying the search space for tracking procedures.

- The human action can be represented as a sequence of postures, as motion trajectories of body parts, or as the variation of joint angles over time.
- A comparison measure between different human body poses can be established, due to the hierarchical nature of the stick figure representation.
- The direction cosines representation provides a representation exempt of discontinuities with a direct geometric interpretation. Hence, similar configurations of the body limbs account for close values of this representation.

Summarizing, by using this human body model, we take profit from the advantages of using stick figures due to their simplicity and compactness. Thus, it reflects the anatomical structure of the human body and its topology by means of the kinematic tree, i.e. angles are modeled in a relative manner by defining a hierarchy. In addition, the stick figure can be easily fleshed out with volumetric models in the future without having to modify the stick definition. Finally, singular positions and discontinuities over time w.r.t. to the angles between limbs are prevented by choosing direction cosines for representing limbs' direction.

3.4 Importing Motion Capture Data

Commercial motion capture systems store the captured data in many different formats, i.e. C3D, BVH, FBX, BIP, etc. Such formats may differ between them regarding the internal format used to organize the captured data, the articulated model employed, the body representation (angular vs. positional data), etc. Hence, in order to make use of external motion captured data in this work, we defined a procedure for importing already existing motion captured data into our representation.

Regardless the internal representation of data used, most motion capture systems provide methods for computing the raw positional data of each motion captured marker. Therefore, we adopted this positional representation of motion capture data as a generic intermediate representation constituting the input for importing data from motion capture databases. Hence, we assume, that in this representation, human body postures are represented by the raw XYZ positions of the M markers attached to the subject in an absolute world coordinates system, denoted as $X_t = (x_1, ..., x_M)$. Notice that depending on the capture setup used, the number of markers and their topology within the kinematic tree may vary, so this will need to be considered beforehand when working with different sets of data. Once we have all M markers position, we convert this data to our human body model representation by establishing a mapping between the M markers and our 15 end-points (see Fig. 3.9.(a)).

Subsequently, we will explain the conversion process for one external dataset used in this work, namely the CMU motion capture dataset. This dataset was acquired using a Vicon optical motion capture system[2] which captured a working volume of approximately 3m x 8m. The captured subjects worn a total of 41 markers, whose detailed placement explanation can be found at CMU's motion capture dataset web-

[2]Vicon Motion Systems: `http://www.vicon.com/`

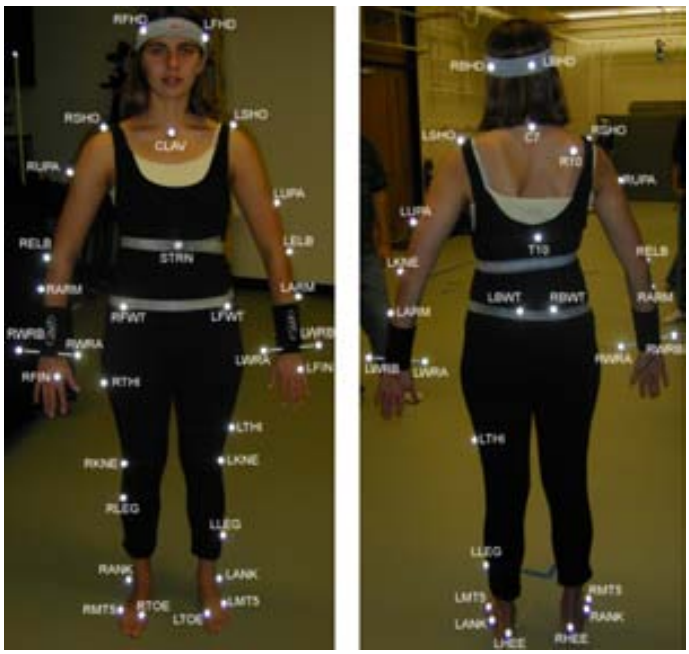


Figure 3.10: Details of the CMU motion capture dataset marker placement.

site[3]. In Fig. 3.10 we show the markers used and their placement on the body, with the names associated to them.

Then, considering the huge detail and number of markers used, we select some principal markers for the mapping in order to make the input motion capture data usable according to our human body representation. Towards this end, we selected the markers named as: LFWT, LBWT, RFWT, RBWT, RFHD, LFHD, LSHO, LELB, LWRB, RSHO, RELB, RWRB, LKNE, LHEE, RKNE and RHEE.

The markerset selected and the mapping used is fully depicted in Fig. 3.11.(b), which relates the absolute position of each joint from our human body model with the markers' used in the CMU database. For instance, in order to compute the position of joint 5 (head) in our representation, we should compute the mean position between the RFHD and LFHD markers from the CMU database, which correspond to the markers placed on each side of the head. Notice that our model considers the left and the right parts of the hip and the torso as a unique limb, and therefore we require a unique segment per each. Hence, we compute the position of joints 1 and 4 (hip and neck joints) as the mean between the previously computed joints 2 and 3, and 6 and 9 respectively.

[3]CMU motion capture dataset marker placement guide:
`http://mocap.cs.cmu.edu/markerPlacementGuide.pdf`

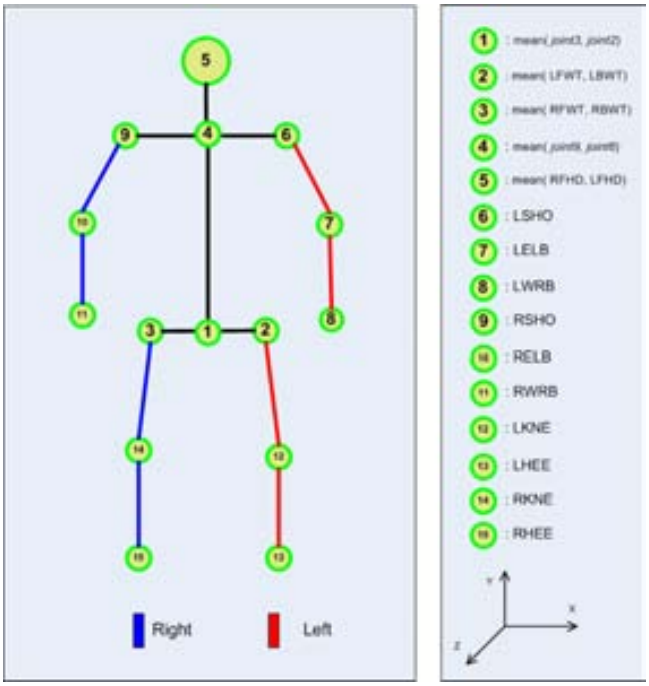


Figure 3.11: Details of the human body model used and the relationship to the markerset employed in the CMU database.

Once the mapping between CMU markers and end-point positions of our body model is done, we use Eq.(3.1) to represent each posture from the CMU dataset in terms of our representation as specified in Eq.(3.3).

Chapter 4

Human Action Modeling

The action model presented in this work aims to describe how humans move, by learning the variability and temporal evolution of full-body motion within a particular action. On the one hand, we characterize which configurations of the human body model's parameters correspond to feasible postures and which postures are typical from each action. On the other hand, we learn how postures change during the performance of the action. Thus, providing a way of making good predictions about human postures within our model-based tracking approach.

Our action model is example-based, i.e. it is learnt from examples of a human motion database comprised by motion sequences of different sources and natures. On the first hand, we built an on purpose made action database by asking several actors to perform a given set of actions. On the second hand, motion sequences from a publicly available dataset (Carnegie Mellon University's (CMU) Graphics Lab Motion capture database) has also been used for the definition of our training set. In both datasets, the 3D postures for each action have been recorded using a commercial Motion Capture system which provides very accurate motion data from a set of reflective markers carefully attached to the human body.

All the motion performances from the training dataset are synchronized using a Dynamic Programming (DP) algorithm and a mean manifold for a set of training performances is computed. As a result, we can analyze intra-performance differences for each time step. In other words, we can quantify the difference between the same part of two different performances of the same action. Then, a mean direction of motion is learnt for subsequences of a determined length, and statistics are extracted from the synchronized dataset that characterize the variation observed in each step between different training performances.

Within this chapter, we first detail the motion datasets used for training. Then, we introduce a representation space for human actions and define the synchronization algorithm. Finally, we explain the learning procedure of our action specific motion model.

4.1 Human Action Training Sets

We define a human motion as a sequence of human postures which are exhibited while performing a particular action. Two human motion datasets were considered for training our action model. Both datasets consist of several sequences of human postures recorded by several actors for a predefined set of actions, and were acquired using a commercial motion capture system based on optical markers. Since these databases vary in terms of the actions covered, the number of subjects recorded, and the parameters available for each, we selected different number of actions and performances to define proper training sets for each.

The first dataset, namely the CVC motion dataset was developed on purpose for this work and covers a comprehensive number of actions, i.e. 9 actions with a balanced number of recorded sequences and actors per each action. The second dataset, namely the CMU motion capture dataset consists of a great number of sequences divided into 6 categories and 23 subcategories. For this work, we focused on the "walking" subcategory from the "Locomotion" category.

The following, explains the data acquisition procedure, the marker placement, and the number of subjects and sequences composing each database.

4.1.1 The CVC training set

In order to fit our needs for studying human motion, we built our own action database by asking several actors to perform a given set of actions. We have concentrated in a predefined set of 9 human actions to be studied, in particular the bending, running, kicking, jumping, squatting, tumbling, sitting, walking and skipping actions. We designed a training set of human actions consisting in the above mentioned 9 actions performed by different actors. The actors performed the same action several times. All the 3D data was acquired using commercial Motion Capture systems following the process described next.

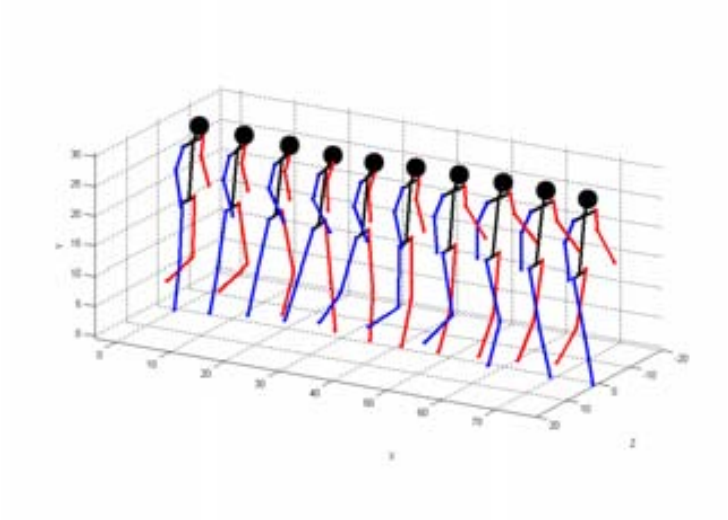
Procedure for data acquisition

We have used an optical Motion Capture system[1] to acquire real training data for our algorithms. Optical systems include minimal reflective markers which are used to recover the relative motion of the agent. This system is based on six synchronized video cameras to record images. The optical system incorporates all the elements and equipment necessary for the automatic control of cameras and lights during the capture process. It also includes a software pack for the reconstruction of movements and the effective treatment of occlusions.

In our experiments, the subject first placed a set of 19 reflective markers on the joints and other characteristic points of the body, see Fig. 4.1.(a). These markers are small round pieces of plastic covered in reflective material. Subsequently, the agent is placed in a controlled environment (i.e., controlled illumination and reflective noise), where the capture will be carried out, see Fig. 4.1.(b).

[1] STT Ingeniería y sistemas: `http://www.stt.es`

(a) (b)



(c)

Figure 4.1: Procedure for data acquisition. Fig. (a) shows the agent with the 19 markers on the joints and other characteristic points of its body. (b) shows the scene where the motion capture system acquired the training samples. (c) corresponds to some frames of the 3D data acquired for a walking cycle.

As a result, the accurate 3D positions of the markers are obtained for each recorded frame, 30 frames per second. In our experiments, not all the 18 markers are considered to model human actions. In fact, we only process those markers which correspond to the joints of the human body model detailed in the last section. In Fig. 4.1.**(c)** the 3D positions acquired from the Motion Capture system have been

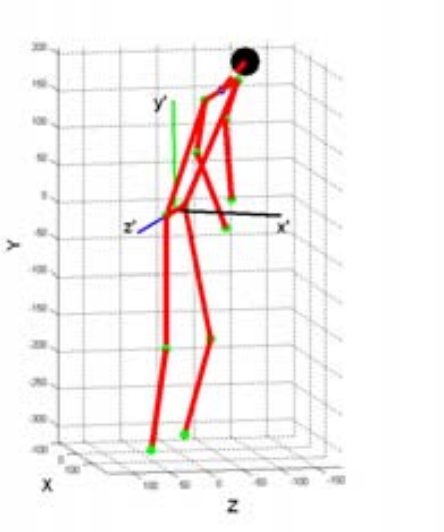


Figure 4.2: Normalization in position and orientation of body postures. The origin of the local reference system (x', y', z') is placed at the center of the hip with its x' axis pointing toward.

mapped into our human body model. We can see the 3D body postures obtained for a walking cycle of an agent.

Due to the high variability in the recording sessions -each recording session had to be calibrated separately, and the actors performed each action in different parts of the room, showing different orientations- , all the performances have been normalized in order to work with them. The normalization process has been done regarding the size of each subject, his height, and the global orientation and position of the human body.

We have calculated a local reference system for all the postures, in order to force to each performance to an identical starting position and orientation of the hip. Subsequently, for each performance of the database, the center of the hip of every first 3D posture has been forced to lie at the Cartesian position $(0, 0, 0)$. All the postures have been re-expressed with regard to the local reference system defined by the (x', y', z') axes. Fig. 4.2 depicts this scenario. The posture shown is expressed in a global reference system spanned by the (X, Y, Z) axes. A local reference system has been calculated which corresponds to the (x', y', z') axes in black, green and blue respectively. The calculation of the local system has been done as follows: first, the z' axis is forced to coincide with the orientation of the hip at the first frame. Then, the y' axis is forced to be perpendicular to the previous one and to point upwards, and finally, the x' axis is constrained to be perpendicular to the y' and z' axis and to look ahead.

The nature of the actions acquired, and the details about the motion database are explained next.

Action Database

We considered a set of 9 different elementary action types which are more typical of human motion:

- **aBend:** a bending action where a subject is standing, then he bends to the floor as if he were picking up something and ends standing like in the beginning of the action. Sequences of both right and left hand were considered when picking the imaginary object from the floor.
- **aJump:** a jumping action where a subject is standing, then he folds his knees, jumps, and ends in the same initial position. The subject uses his arms to help impulsing himself which end up pointing upwards while performing the jump.
- **aKick:** actually this action consists of a standing subject performing the following movements in order: punching with the right arm, punching with the left arm, kicking with the right leg, and finally kicking with the left leg.
- **aRun:** a running action where a subject performs several run cycles. Each cycle is segmented and considered as a performance of the action.
- **aSit:** a sitting action where a standing subject sits to a chair and stands up again.
- **aSkip:** a subject passing his legs over an obstacle lying on the floor to skip it. First the right leg and then the left one.
- **aSquat:** a squatting action, where a standing subject folds his knees until he actually sits on his haunches. Then, he stands up again.
- **aTumble:** a subject standing up, literally sits down on the floor and stands up again.
- **aWalk:** a walking action, where a subject performs several walking cycles. Likewise the running action, each cycle is segmented and considered as a performance of the action.

Once the set of actions was defined, we asked to nine different actors to perform each action. In order to provide a proper learning set which is generic enough, each actor performed each action an average of 5 times. The actor set consisted of 3 females and 6 males. Thus, as a result 45 *performances* of the same action were recorded (in average). Hereafter we refer to a performance as a sequence of 3D human body postures which correspond to a particular action performed by a particular actor in a particular manner.

Table 4.1 shows a summary of the performances contained in this action database. For each elementary action, information about the number of performances recorded, the number of frames, and the number of actors involved is shown. Notice that, due to the cyclic nature of actions such as running or walking, several cycles of the action were captured per each actor in each recording session. Several performances were obtained a posteriori by manually-segmenting each sequence. Differences between the number of samples per action in the database are due to several reasons: in the first place, the number of frames per action depends on the length of the action itself. On

Action	Total #performances	Total #frames	#actors	Average #performances/actor	Average #frames/performance
aBend	51	3921	9	5.7	77
aJump	48	2578	8	6	54
aKick	43	7242	9	4.8	168
aRun	40	4042	5	9	101
aSit	28	3309	6	4.7	118
aSkip	13	1053	4	3.3	81
aSquat	54	4202	9	6	78
aTumble	31	5301	8	3.9	171
aWalk	32	4038	4	8	126

Table 4.1: Detail of the CVC motion database.

the other hand, we had to trash some of the captured data mainly because of noise in the capturing process, and calibration errors. Therefore, the number of performances and actors appears slightly unbalanced between actions.

Finally, in Figures 4.3 and 4.4 we show the remaining actions from this database, i.e. the aBend, aJump, aKick, aRun, aSit, aSkip, aSquat and aTumble actions. For each action, a sequence of postures from a performance is drawn using a stick figure representation.

4.1.2 The CMU motion capture dataset

The CMU motion capture dataset is composed of 2605 trials organized in 6 categories and 23 subcategories[2]. The 6 main motion categories are "Human Interaction", "Interaction with Environment", "Locomotion", "Physical Activities & Sports", "Situations & Scenarios" and "Test Motions". We focused in the Locomotion category which in turn is subdivided into the "walking", "running", "jumping" and "varied" subcategories. From these, the walking, running and jumping subcategories are similar to the 9 actions we selected for building the CVC dataset, but only the walking action comprises a representative number of performances done by many different actors. Therefore, we used this subcategory for training our action model with this dataset.

As a result this training set is composed of 12 subjects showing different performances of the walking action. In turn, each walking performance consists of a variable

[2] at present.

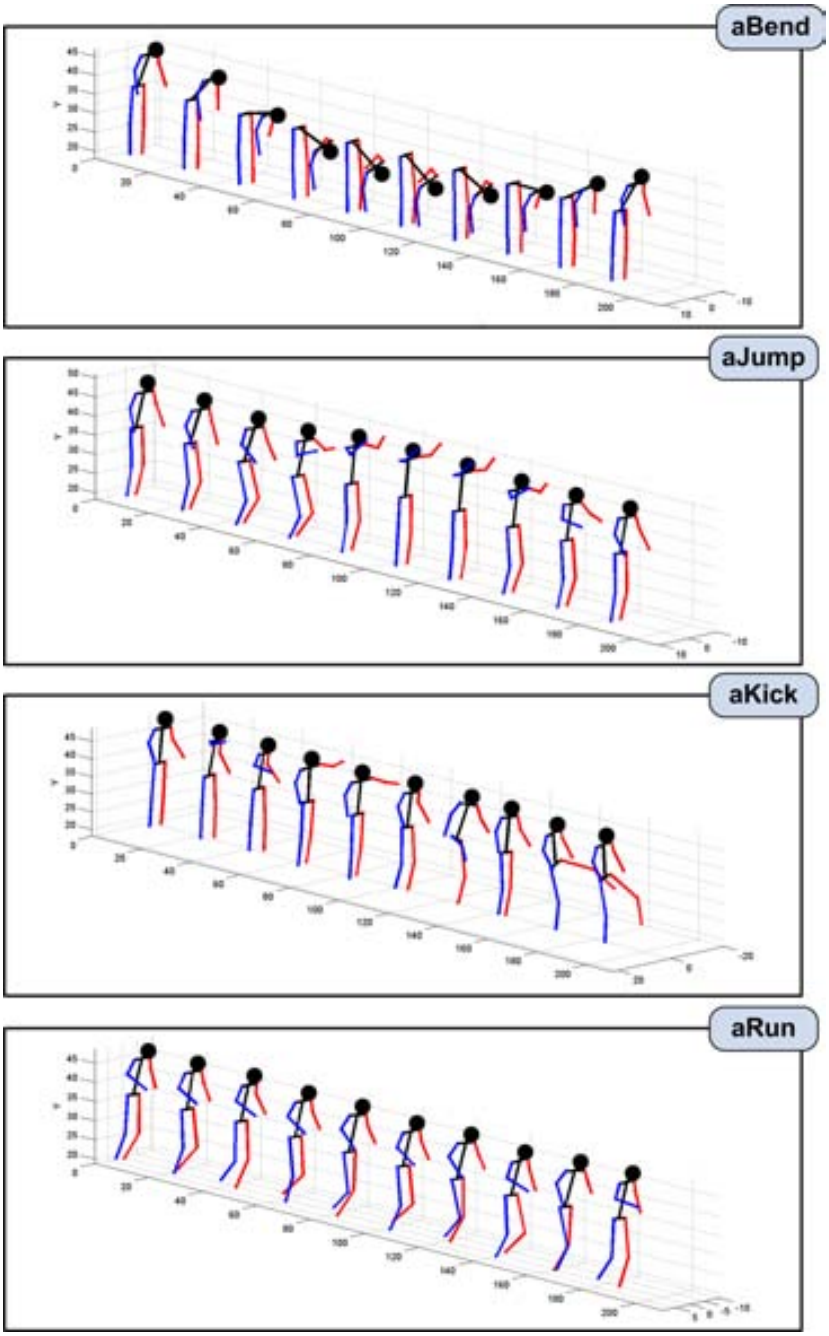


Figure 4.3: Sample frames from the aBend, aJump, aKick and aRun actions from the CVC dataset.

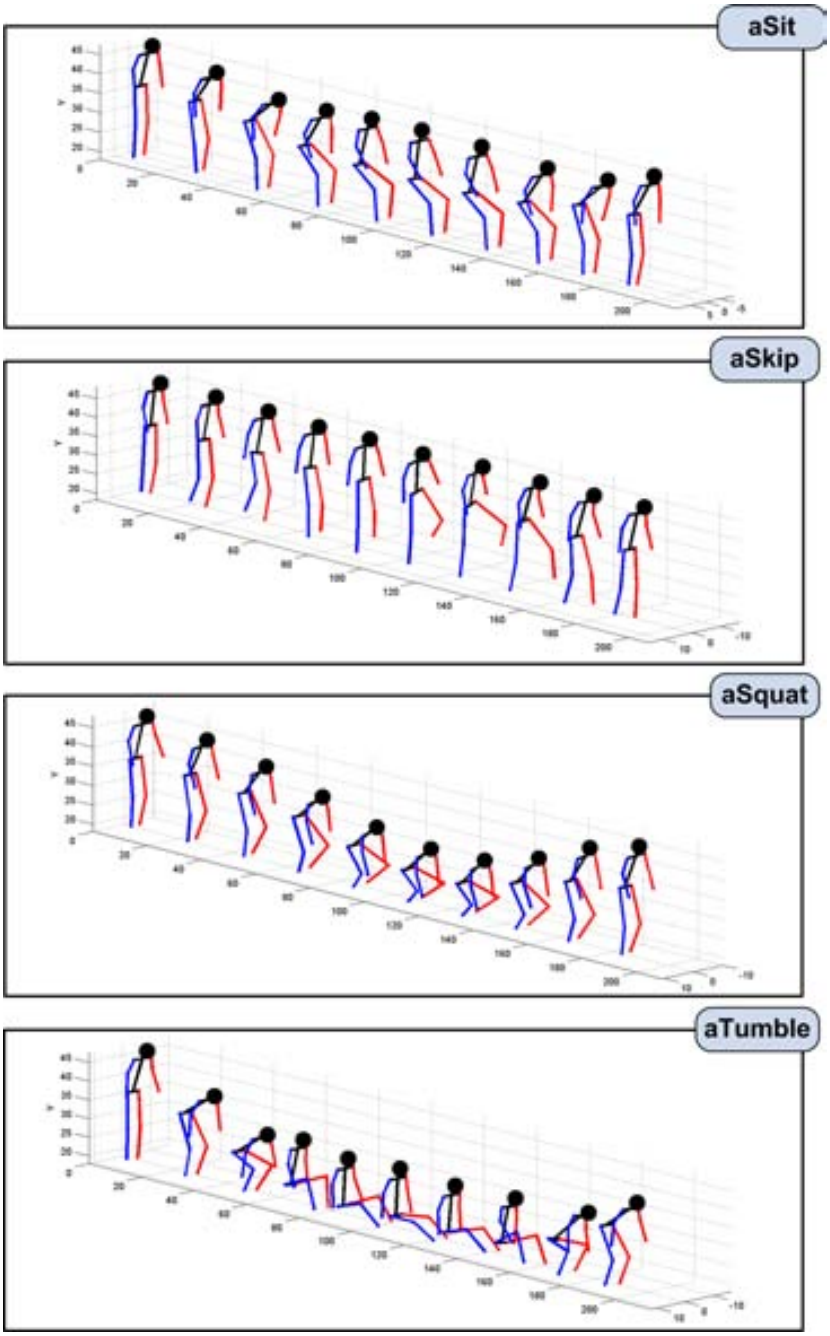


Figure 4.4: Sample frames from the aSit, aSkip, aSquat and aTumble actions from the CVC dataset.

Subject id.	Index of selected performances	# recorded performances	Total # of walking cycles	Total # body postures
2	{1, 2}	2	3	372
5	{1}	1	3	448
7	{1, 2, 3, 6, 7, 8, 9, 10 ,11}	9	15	2027
8	{1, 2, 3, 6, 9, 10}	6	9	1058
12	{3}	1	3	482
16	{15, 16, 21, 22, 31, 32, 47}	7	15	1977
35	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 28, 29, 30, 31, 32, 33, 34}	23	42	5782
38	{1, 2}	2	4	540
39	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14}	13	26	3260
43	{1}	1	2	263
49	{1}	1	3	491
55	{4}	1	1	191
TOTAL		**67**	**126**	**16891**

Table 4.2: Detail of the CMU training set composition.

number of cycles ranging from 1 to 5. Subsequently, each recorded performance is split into its composing walking cycles. We used the angle between the left and right legs as the criterion for splitting walking cycles. A full cycle is defined as all the body postures in between two consecutive maximums of the angle between both legs when the left leg remains in the back. Incomplete cycles and erroneous sequences were discarded from the training set. As a result, we finally end up with a set of 16891 body postures corresponding to 126 walking cycles performed by 12 different actors showing different speeds and different body configurations while performing the same action. Table 4.2 details the composition of our training set. The number of each subject and recorded performance corresponds to the same indexes used in the CMU database.

For more details on the marker placement for this dataset and the mapping to our human body model, please, revisit section 3.4 and in particular Figures 3.10 and 3.11.

4.2 Human Action Spaces or *aSpaces*

It is desirable that our action model fulfills the requirements of compactness, accuracy and specificity. In other words, it should capture most of the information of the training data set in a small number of parameters. Also, it should model well enough all the set of body configurations that a human body can exhibit while performing a particular action, but not those postures which are not likely to be adopted during such an action.

In addition, due to the direction cosines representation we are using 3 parameters to determine only 2 DOF for each limb. Such representation generates a considerable redundancy of the vector space components. Furthermore, the human body motion is intrinsically constrained, and these natural constraints lead to highly correlated data in the original space. Therefore, we aim to find a more compact representation of the original data to avoid redundancy. To do this, we consider the training set of all the postures belonging to an action, and perform Principal Component Analysis (PCA).

By applying PCA, a new orthogonal basis for the training data is computed, which are the so-called eigenvectors of the data covariance matrix. Hence, all the training postures are expressed as a linear combination of the new basis. Furthermore, the vectors of the new basis coincide with maximum variance directions of the training data. As a result, projections into that new space lead to an uncorrelated version of the original data.

Usually, few eigenvectors describe the most of the variance of the training data, so a lower dimensional version of the data is obtained which preserves the most amount of information from it. Moreover, distances within the eigenspace provide a natural way of measuring similarities between human postures. Furthermore, only plausible samples are learned and new ones can be generated by modifying the existing ones within the limits of the principal modes of variation.

Hence, the PCA-like space, hereafter the *aSpace*, is built from the recorded examples of human motion in order to capture the intrinsic characteristics of it. Subsequently, we detail the process followed for building this *aSpace* which has been done for each individual action of the training set.

First, let us define the training set for an action $\mathbf{A}_k$ as a collection of performances for that particular action. A performance Ψ_i of an action consists of a time-ordered sequence of body postures such as

$$\Psi_i = \{\psi_i^1, ..., \psi_i^{F_i}\}, \tag{4.1}$$

where i stands for the index of the performance Ψ_i of the action action $\mathbf{A}_k$ and F_i is the total number of human postures of that performance.

As a result, we define the complete training set of human postures for an action $\mathbf{A}_k$ as:

$$\mathbf{A}_k = \{\Psi_1, ..., \Psi_P\}, \tag{4.2}$$

where P refers to the overall number of training performances for this action. Each posture ψ_i^j of a performance Ψ_i is of of dimensionality 36×1 corresponding to the parameters of the human body model from Eq. (3.3). Therefore, the total number of

training postures ψ_i^j, for an action k is determined by

$$N_{A_k} = \sum_{i=1}^{P} F_i. \tag{4.3}$$

The next step is concerned in building the action space by performing PCA on the training data set. For each action $\mathbf{A}_k$, we compute its covariance matrix as:

$$\mathbf{\Sigma}_{A_k} = \frac{1}{N_{A_k}} \sum_{i=1}^{P} \sum_{j=1}^{F_i} (\psi_i^j - \bar{\psi})(\psi_i^j - \bar{\psi})^T, \tag{4.4}$$

where N_{A_k} refers to the overall number of postures for that action, and $\bar{\psi}$ is the mean human body posture for that action. Then, the eigenvectors $\mathbf{u}_n$ and the eigenvalues λ_n of $\mathbf{\Sigma}_{A_k}$ are calculated by solving the eigenvector decomposition equation:

$$\lambda_n \mathbf{u}_n = \mathbf{\Sigma}_{A_k} \mathbf{u}_n. \tag{4.5}$$

Each obtained eigenvector $\mathbf{u}_n$ corresponds to a mode of variation of the data, while its corresponding eigenvalue λ_n accounts for the variance explained by that eigenvector. The full set of n eigenvectors constitute a new orthogonal basis spanning an space where the data can be projected. However, by selecting only the first b eigenvectors, a $\{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_b\}$, as the new basis, the original data can be re-expressed in a lower dimensional space. Thus, b determines the dimensions of the action space or *aSpace*.

The value of b is commonly set by eigenvalue thresholding. The overall variance λ^T of the training samples that we keep is computed as the sum of the eigenvalues, i.e.:

$$\lambda^T = \sum_{i=1}^{n} \lambda_i. \tag{4.6}$$

For example, if we want to guarantee that the new representation of the data keeps the 95% of the variance of the original data, we must impose that:

$$\frac{\sum_{i=1}^{b} \lambda_i}{\lambda^T} \geq 0.95. \tag{4.7}$$

Consequently there exists a direct trade-off between the accuracy of the new representation of the data, and the number of parameters needed to model it. That is, the lower number of eigenvectors we choose, the lower dimensionality of the action model is obtained at expenses of errors in reconstructing the original data.

Finally, all the training postures are projected to the PCA space, thus obtaining a lower-dimensional representation of human postures for that action, i.e.

$$\tilde{\psi}_i^j = [\mathbf{u}_1, ..., \mathbf{u}_b]^T (\psi_i^j - \overline{\psi}), \tag{4.8}$$

where ψ_i^j refers to the original posture, $\tilde{\psi}_i^j$ denotes the lower-dimensional version of the human posture represented in the PCA space, $[\mathbf{u}_1, ..., \mathbf{u}_b]$ is the PCA space

transformation matrix that correspond to the first b selected eigenvectors, and $\overline{\psi}$ is the mean of all the training postures. Subsequently, we denote the lower dimensional version of a particular performance as $\tilde{\mathbf{\Psi}}_i$ such as

$$\tilde{\mathbf{\Psi}}_i = \left\{ \tilde{\psi}_i^1, ..., \tilde{\psi}_i^{F_i} \right\}. \tag{4.9}$$

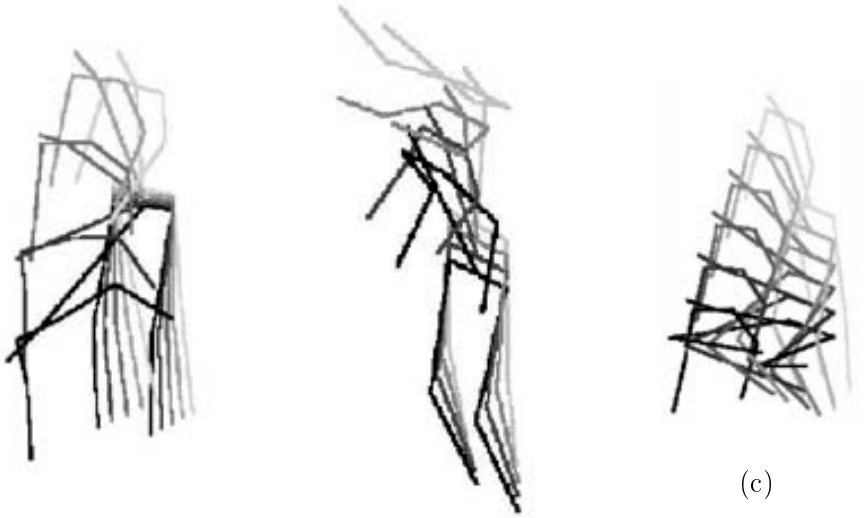
We name the resulting PCA-like space as the *aSpace* [26] for a particular action. Each dimension of the *aSpace* describes a natural mode of variation of human motion while performing an action, resulting in a more suitable and compact representation than the original 36-dimensional vector ψ from Eq.(3.3). Notice that, by choosing different values for b we result in models of more or less complexity in terms of their dimensionality. Hence, while the *gross-motion*[3] is explained by the first eigenvectors, subtle motions require more dimensions to be considered in the *aSpace* representation. Therefore, choosing an appropriate value of b, and due to the use of real-life training data, the model for human motion is restricted to plausible configurations, thus avoiding non typical human postures and providing realistic deformations. Another very interesting property of the *aSpace* is that closer points between different manifolds correspond to similar human postures.

Figure 4.5 illustrates this concept by showing the human postures resulting of varying the principal component found for 3 particular actions, namely the bending, jumping and tumbling actions.

The walking action *aSpace*, hereafter *the aWalk* space, has been used to illustrate the overall approach. In Fig. 4.6 we depict the modes of variation found for the human posture along the three first components obtained from PCA. The first (a), second (b), and third (c) eigenvectors of the mean posture are modified from -3 to 3 times the standard deviation found in training postures. As we may observe, the main motion present in the walking action is related to arms and legs, while the motion of the torso is barely perceptible. Hence, the first dimension accounts for the coupled motion between arms and legs, and most of the variance from the training data (69.7%) is explained by this component. Notice that the right arm moves accordingly to the left leg, and in an opposite manner to the pair composed of the left arm and the right leg. In addition, the second and third components explain the 8.5% and 8.2% of the variance present in the training set, and encode a more subtle motion of legs and arms.

In addition, in Fig. 4.7 we show the first three dimensions of a performance of the bending action projected into the *aSpace*. The performance has been sampled at 5 different time steps (frames 1, 20, 50, 75 and 100) which are depicted by big red dots. Their corresponding body postures are shown next to each dot. Similarly to other action spaces, we can observe the fact that each dimension of the *aSpace* corresponds to natural modes of variation of human gait. Moreover, the first dimensions are the

[3] mainly, the motion of the torso, legs and arms.



(a) (b) (c)

Figure 4.5: Variation of the human posture within the *aSpace* space explained by the first principal component found for the bending (a), jumping (b), and tumbling (c) actions.

(a) (b) (c)

Figure 4.6: Variation of the human posture within the *aWalk* space explained by the first (a), second (b), and third (c) principal components.

most important ones, i.e. the dimensions which capture higher variance in the human motion. Thus, it is natural that one main mode of variation corresponds to the fact of bending the torso instead of small movements of hands. This fact is certified by

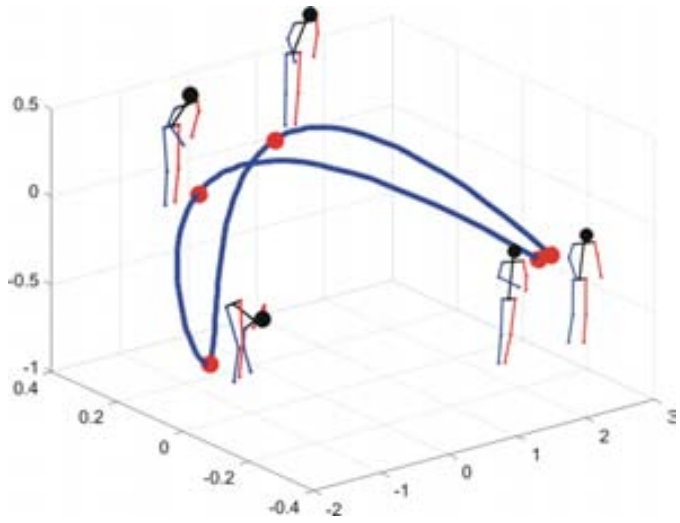


Figure 4.7: Bending action performance projected in the *aSpace*. The three first dimensions are shown which correspond to the main three modes of variation of human motion. The postures from frames 1, 20, 50, 75 and 100 are depicted in the curve by big red dots. Their corresponding sampled body postures are also shown.

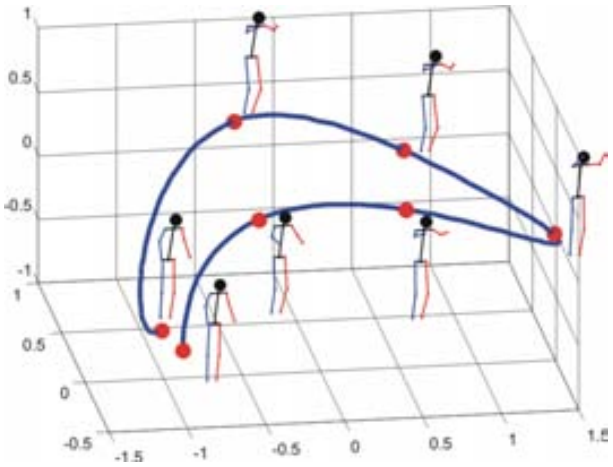


Figure 4.8: Jumping action performance projected in the *aSpace*. The three first dimensions are shown which correspond to the main three modes of variation of human motion. The postures from frames 1, 32,45,60, 80, 90 and 100 are depicted in the curve by big red dots. Their corresponding sampled body postures are also shown.

the postures shown in Fig. 4.7 and their corresponding position in the *aSpace*. One can observe on the one hand, that both initial and final positions lie almost in the

same place of the space: indeed, they are very similar postures. However, on the other hand, when the subject is totally bent, the postures lie in the opposite part of the figure, i.e. the first dimension captures information about the "bendness" of the torso.

Finally, we provide another example for the jumping action. The same situation can be observed in Fig. 4.8 where the projections of a jumping performance have been plotted. In this case, the manifold has been sampled at 7 different time steps (frames 1, 32,45,60, 80, 90 and 100). One can observe that as the subject is completing the jumping action, the position of the posture in the *aSpace* moves from the left to the right part of the figure. In this case, the corresponding dimension expresses the motion of the shoulder complex.

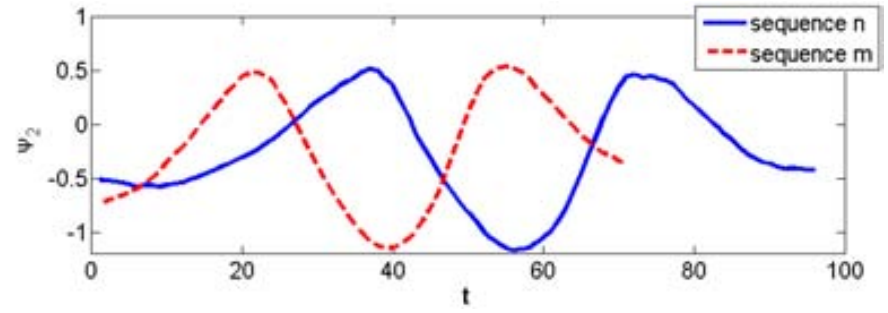
Summarizing, the *aSpace* has been proved suitable to represent human postures, thus obtaining a lower dimensional representation for postures than the original human body model with some intrinsic interesting properties. First, similar human postures lie in close points within the space. Then, each component of the space corresponds to a natural mode of variation of human motion. Finally, the complexity of the model can be selected by choosing the number b of dimensions of the *aSpace* to keep.

4.3 Synchronization of the Training Set

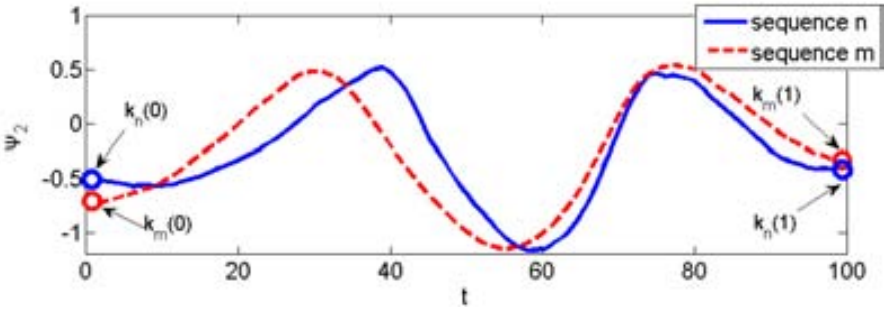
As stated before, the training sequences have been acquired under different conditions and by different actors, showing different durations, velocities and accelerations during the performance of a particular action. As a result, it is difficult to put in correspondence postures from different sequences of the same action in order to perform useful statistical analysis to the raw training data. Therefore, a method for synchronizing the whole training set is used so that we can establish a mapping between postures from different sequences.

Inspired by techniques used in the stereo-matching and image processing literature [13, 68], we developed a novel dense matching algorithm based on Dynamic Programming (DP), which allows us to find an optimal solution for synchronizing the pre-recorded motion sequences of the same class in the presence of different speeds and accelerations. Towards this end, we first compute the similarity between each pair of training sequences with a given metric. Then, in order to extract from the input data set the best time scale pattern for synchronization, an intra-class minimum global distance criterion is used. Finally, all performances are synchronized to the computed time pattern. The detailed explanation of the process is as follows.

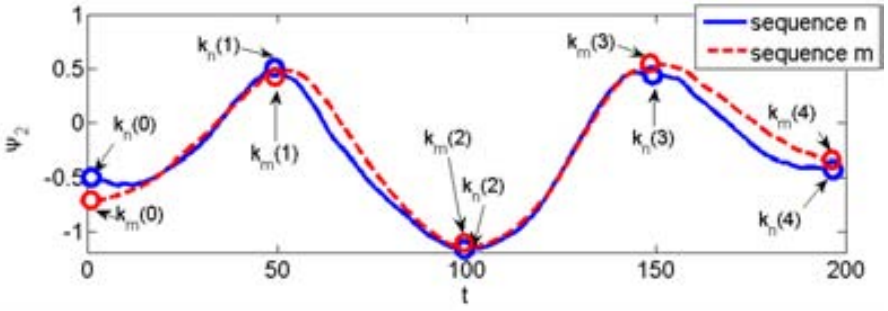
The projection of the training sequences into the PCA space constitutes the input for our sequence synchronization algorithm. Before starting synchronizing the dataset, all the performances are resampled, using cubic spline interpolation, so that all the performances have exactly the same number of frames F. The longest performance from the training set is chosen to be the one which determines the number of frames F of the rest of the set.



(a)



(b)



(c)

Figure 4.9: (a) Non synchronized one-dimensional sequences. (b) Linearly synchronized sequences. (c) Synchronized sequences using a set of key-frames.

Hereafter, we consider a multidimensional signal $\mathbf{x}_i(t)$ as an interpolated expansion of each training performance $\tilde{\mathbf{\Psi}}_i = \left\{\tilde{\psi}_i^1, ..., \tilde{\psi}_i^F\right\}$ such as

$$\mathbf{x}_i(t) = \tilde{\psi}_i^f \quad if \quad t = (f-1)\delta f; \qquad\qquad f = 1, ..., F; \tag{4.10}$$

where the time domain of each action performance $\mathbf{x}_i(t)$ is $[0, T)$. Notice that all the input sequences $\mathbf{x}_i(t)$ have the same period T.

Before describing the full approach, let us introduce the problem by discussing on different strategies one could follow to synchronize 2 given signals, $\mathbf{x}_n$ and $\mathbf{x}_m$ with different periods.

Let us assume that the two considered signals correspond to the identical action, but one runs faster than another (e.g. Fig. 4.9.(a)).

Then, under the assumption that the rates ratio of the compared actions is a constant, the two signals might be easily linearly synchronized in the following way

$$\mathbf{x}_n(t) \approx \mathbf{x}_{n,m}(t) = \mathbf{x}_m(\rho t); \quad \rho = \frac{T_m}{T_n}; \tag{4.11}$$

where $\mathbf{x}_n$ and $\mathbf{x}_m$ are the two compared multidimensional signals, T_n and T_m are the periods of the action performances n and m, $\mathbf{x}_{m,n}$ is the linearly normalized version of $\mathbf{x}_m$, hence $T_n = T_{m,n}$. Unfortunately, in real-world scenarios we rarely if ever have a constant rate ratio ρ. An example, which is illustrated in Fig. 4.9.(b), shows that a simple normalization using Eq.(4.10) does not give us the needed signal fitting, and a nonlinear data synchronization method is needed. Further in this section, we shall assume that the linear synchronization is done beforehand and all the periods T_n have the same value T.

Instead, the nonlinear data synchronization could be done by

$$\mathbf{x}_n(t) \approx \mathbf{x}_{n,m}(t) = \mathbf{x}_m(\tau);\ \tau(t) = \int_0^t \rho(t)dt; \tag{4.12}$$

where $\mathbf{x}_{n,m}(t)$ is the best synchronized version of the sequence $\mathbf{x}_m(t)$ to the sequence $\mathbf{x}_n(t)$. In the literature the function $\tau(t)$ is usually referred to as the distance-time function or rate-to-rate synchronization function.

The rate-to-rate synchronization function $\tau(t)$ satisfies several useful constraints, i.e.

$$\tau(0)=0;\ \ \tau(T)=T;\ \ \tau(t_k) \geq \tau(t_l)\ \ \text{if } t_k > t_l. \tag{4.13}$$

One common approach for building the function $\tau(t)$ is based on a key-frame model. This model assumes that the compared signals $\mathbf{x}_n$ and $\mathbf{x}_m$ have similar sets of singular points, that are $\{t_n(0), \ldots t_n(p), ..t_n(P-1)\}$ and $t_m(0), \ldots t_m(p), ..t_m(P-1)\}$ with the matching condition $t_n(p) = t_m(p)$. The aim is to detect and match these singular points, thus the signals $\mathbf{x}_n$ and $\mathbf{x}_m$ are synchronized.

However, the singularity detection is an intricate problem itself, and to avoid the singularity detection stage we propose to use a dense matching algorithm. In this case a time interval $t_n(p+1) - t_n(p)$ is constant, and in general $t_n(p) \neq t_m(p)$.

The function $\tau(t)$ can be represented as $\tau(t) = t(1 + \Delta_{n,m}(t))$. In this case, the sought function $\Delta_{n,m}(t)$ might synchronize two signals $\mathbf{x}_n$ and $\mathbf{x}_m$ by

$$\mathbf{x}_n(t) \approx \mathbf{x}_m(t + \Delta_{n,m}(t)t); \tag{4.14}$$

Let us introduce a formal measure of synchronization of two signals by

$$D_{n,m} = \int_0^T \|\mathbf{x}_n(t) - \mathbf{x}_m(t + \Delta_{n,m}(t)t)\| \, dt + \mu \int_0^T \left\| \frac{d\Delta_{n,m}(t)}{dt} \right\| dt. \tag{4.15}$$

where $\|\bullet\|$ denotes one of possible vector distances, $D_{n,m}$ is referred to as the synchronization distance that consists of two parts, where the first integral represents the functional distance between the two signals, and the second integral is a regularization term, which expresses desirable smoothness constraints of the solution. The proposed distance function is simple and makes intuitive sense. It is natural to assume that the compared signals are synchronized better when the synchronization distance between them is minimal. Thus, the sought function $\Delta_{n,m}(t)$ should minimize the synchronization distance between matched signals.

In the case of a discrete time representation, Eq.(4.15) can be rewritten as

$$D_{n,m} = \sum_{i=0}^{<P} \left| \mathbf{x}_n(i\delta t) - \mathbf{x}_m\left(i\delta t + \Delta_{n,m}(i)\delta t\right) \right|^2 + \mu \sum_{i=0}^{<P-1} \left| \Delta_{n,m}(i+1)\delta t - \Delta_{n,m}(i) \right|, \tag{4.16}$$

where δt is a time sampling interval. Eq.(4.13) implies

$$|\Delta_{n,m}(p+1) - \Delta_{n,m}(p)| \leq 1, \tag{4.17}$$

where index $p = \{0, \ldots, P-1\}$ satisfies $\delta t \, P = T$.

The problem of synchronizing two multidimensional signals $\mathbf{x}_n(t)$ and $\mathbf{x}_m(t)$ is similar to the matching problem of two epipolar lines in a stereo image. For stereo matching a Disparity Space Image (DSI) representation is usually employed [13, 68]. The DSI approach assumes that a 2D DSI matrix has dimensions time p and and disparity d, ranging from $0 \leq p < P$, and $-D \leq d \leq D$. Let $E(d, p)$ denote the DSI cost value assigned to each DSI matrix element (d, p) calculated by

$$E_{n,m}(p, d) = \left| \mathbf{x}_n(p\delta t) - \mathbf{x}_m(p\delta t + d\delta t) \right|^2. \tag{4.18}$$

Consequently, we formulate the synchronization task as an optimization problem as follows: find the time-disparity function $\Delta_{n,m}(p)$, which minimizes the synchronization distance between the compared signals $\mathbf{x}_n$ and $\mathbf{x}_m$, i.e.

$$\Delta_{n,m}(p) = \arg\min_d \sum_{i=0}^{<P} E_{n,m}(i, d(i)) + \mu \sum_{i=0}^{<P-1} \left| d(i+1) - d(i) \right|. \tag{4.19}$$

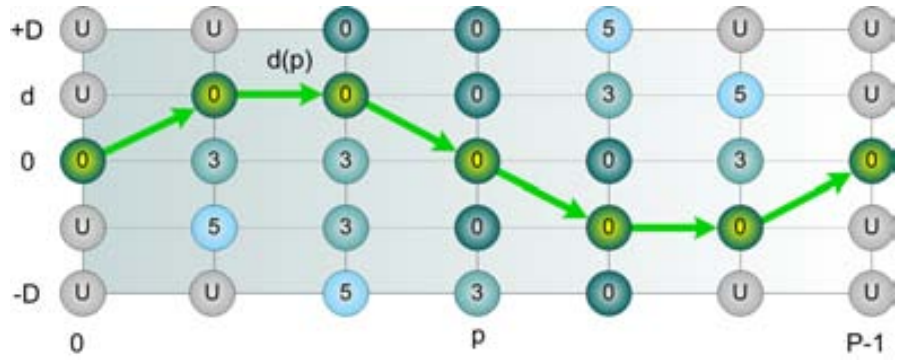
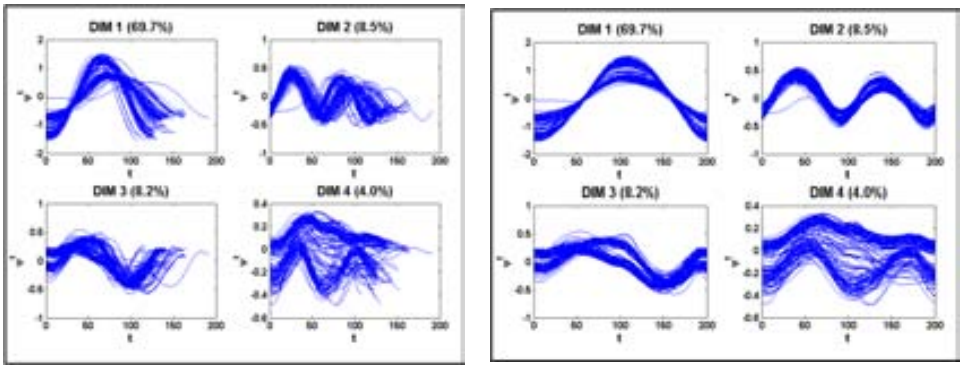


Figure 4.10: The optimal path trough the DSI trellis.



(a) (b)

Figure 4.11: The first $b = 4$ dimensions within the *aWalk* PCA space before (a) and after (b) synchronization of the training set for the walking action.

The discrete function $\Delta_{n,m}(p)$ coincides with the optimal path through the DSI trellis as it is shown in Fig. 4.10. In other words, we must find the path whose sum of cost values plus its weighted length is minimal among all other possible paths. This is solved efficiently by using the Dynamic Programming. The method consists of an step-by-step control and optimization given by the following recurrence relation:

$$\begin{aligned} S(p,d) &= E(p,d) + \min_{k\in 0,\pm 1} \{S(p-1,d+k) + \mu 1d + k1\}, \\ S(0,d) &= E(0,d), \end{aligned} \tag{4.20}$$

where the scope of the minimization parameter $k \in \{0, \pm 1\}$ is chosen in accordance with Eq.(4.17). By using that recurrence relation, the minimal value of the objective function in Eq.(4.19) can be found at the last step of optimization. Next, the algorithm works in reverse order and recovers a sequence of optimal steps (stored in a lookup

table $K(p,d)$ for the values of the index k in the recurrence relation given by Eq.(4.20)) and eventually the optimal path, given by

$$\begin{aligned} d(p-1) &= d(p) + K(p, d(p)), \\ d(P-1) &= 0, \\ \Delta(p) &= d(p). \end{aligned} \tag{4.21}$$

Finally, having found $\Delta_{n,m}(p)$, the synchronized version of $\mathbf{x}_m(t)$ to a base rate sequence $\mathbf{x}_n(t)$ might be calculated by

$$\mathbf{x}_{n,m}(p\delta t) = \mathbf{x}_m(p\delta t + \Delta_{n,m}(p)\delta t). \tag{4.22}$$

Summarizing, the dense matching algorithm that synchronizes two arbitrary human motion sequences $\mathbf{x}_n(t)$ and $\mathbf{x}_m(t)$ is as follows:

1. Prepare a 2D DSI matrix, and set initial cost values E_o using Eq. (4.18)

2. Find the optimal path trough the DSI using recurrence Eqs. (4.20), (4.21).

3. Synchronize $\mathbf{x}_m(t)$ to the rate of $\mathbf{x}_n(t)$ using Eq.(4.22).

Our algorithm assumes that a particular sequence is chosen to be a time scale pattern for all other sequences. In order to make an optimal choice of the sequence that will be used as the pattern for synchronizing the rest, a statistically proven rule according to some appropriate criterion is desirable. Towards this end, we use the synchronization distance between a pair of sequences (n,m) given by Eq.(4.16) to determine which sequence from the training set will be used as the time pattern.

Hence, we compute the global distance of the full synchronization of all the sequences m relative to the pattern sequence n as

$$D_n = \sum_{m \in A_k} D_{n,m}. \tag{4.23}$$

Therefore, we choose the synchronizing pattern sequence n with minimal global distance D_n: in a statistical sense, such signal can be considered as a median value over all the performances that belong to the set of A_k or can be referred to as *median* sequence.

Finally, after running the algorithm against the whole training set for each action, all the performances $\tilde{\Psi}_i$ are synchronized and will be denoted hereafter as

$$\hat{\Psi}_i = \{\hat{\psi}_i^1, ..., \hat{\psi}_i^F\}. \tag{4.24}$$

Figure 4.11.(a) shows the first 4 dimensions of the input walking sequences represented in the PCA space without performing any synchronization. Figure 4.11.(b) shows the same situation after applying the synchronization algorithm proposed here. Notice that a common motion pattern arises after the synchronization step.

4.4 Learning an action specific model of human motion

Once all the motion sequences share the same time pattern, we learn an action specific motion model which will be used to improve the performance of a particle filter tracker. Towards this end, we want to learn where the postures lie in the *aSpace*, how do they change over time as the action goes by, and what characteristics do the different performances have in common which can be used as *a priori* knowledge within the tracking framework. In other words, we aim to characterize the *shape* and the temporal evolution of the synchronized version of the training set for the action in the *aSpace*. The learning process is detailed below.

First, we extract from the action training set $A_k = \{\hat{\Psi}_1, ..., \hat{\Psi}_P\}$ a mean representation of the action by computing the so-called mean performance $\bar{\Psi} = \{\bar{\psi}_1, ..., \bar{\psi}_F\}$, where each mean posture $\bar{\psi}_t$ is defined as

$$\bar{\psi}_t = \sum_{i=1}^{P} \frac{\hat{\psi}_i^t}{P}, \;\; t = 1, ...F, \tag{4.25}$$

where $\hat{\psi}_i^t$ corresponds to the t-th posture from the i-th training performance, and F denotes the total number of postures of each synchronized performance.

Additionally, to handle the cyclic nature of actions such as walking or running, were applicable, we concatenate the last postures from each cycle with the initial postures of the most close performance according to a Euclidean distance criterion within the PCA space. Then, the very first and last postures from the mean performance are resampled using cubic spline interpolation to soft the transition between cycles.

Subsequently, we observed that there are parts of the action which are performed in a very similar manner by all the subjects, while other parts do not. Furthermore, there is also a dependency between the time step of the action, and the variation from frame to frame among performances. Therefore, it makes sense to learn different parameters of the motion model depending on the time step of the segmented action. This phenomena can be seen in Fig. 4.11.(b) for the case of the walking action.

Towards this end, we quantify how much the training performances Ψ_i vary from the computed mean performance $\bar{\Psi}$ of Eq.(4.25). Therefore, for each time step t, we compute the standard deviation σ_t of all the postures $\hat{\psi}_i^t$ that share the same time stamp t, i.e.

$$\sigma_t = \sqrt{\frac{1}{P} \sum_{i=1}^{P} (\hat{\psi}_i^t - \bar{\psi}_t)}. \tag{4.26}$$

Fig. 4.12 shows the learnt mean performance $\bar{\Psi}$ (red solid line) and ± 3 times the computed standard deviation σ_t (dashed black line) for the walking action. Only the first $b = 6$ principal components from the *aWalk* space are represented in the figure, which explain the 93.3% of the total variation of training data.

Third, we are also interested in characterizing the temporal evolution of the action. Therefore, we compute the mean direction of the motion $\overline{\mathbf{v}}_t$ for each subsequence of

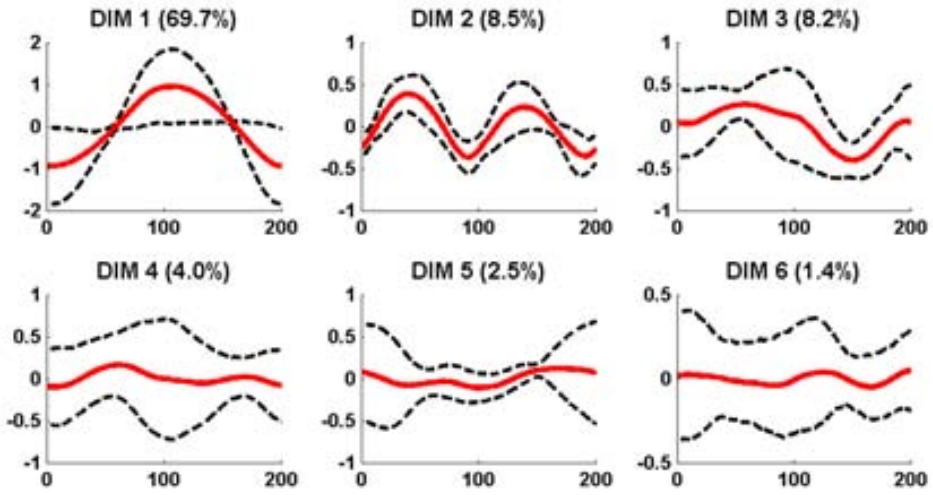


Figure 4.12: Learnt mean performance $\bar{\Psi}$ and standard deviation σ_t for the walking action.

d postures from the mean performance $\bar{\Psi}$, i.e.

$$\overline{\mathbf{v}}_t = \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|}; \qquad \mathbf{v}_t = \frac{1}{d} \cdot \sum_{j=t-d+1}^{t} \frac{(\bar{\psi}_j - \bar{\psi}_{j-1})}{\left\|(\bar{\psi}_j - \bar{\psi}_{j-1})\right\|}, \tag{4.27}$$

where $\overline{\mathbf{v}}_t$ is a unitary vector representing the observed direction of motion averaged from the last d postures at a particular time step t. In our experiments for the walking action, we used $d = 10$ as the length of the subsequences considered out of a mean walking cycle length of $F = 198$ postures.

In Fig. 4.13, the first 3 dimensions of the mean performance are plotted together with the direction vectors computed in Eq.(4.27). Each black arrow corresponds to the unitary vector $\overline{\mathbf{v}}_t$ computed at time t, scaled for visualization purposes. Hence, each vector encodes the mean observed motion's direction from time $(t-d)$ to time t, where d stands for the length of the motion window considered. Additionally, selected postures from the mean performance have been sampled at times $t = 1$, 30, 55, 72, 100, 150 and 168 and have been overlaid in the graphic.

Finally, we learn the expected error from the dynamic model at a given position of the mean performance. Given that new particles will be propagated within the PF following a first order motion model with Gaussian noise, we characterize the expected error committed by the dynamic model as follows. First, for each training sequence, we apply our dynamic model to every posture, and then we compute the error observed w.r.t. the truly performed ones. Then, we store the covariance of the error committed at each time step. Hence, different parameters for the diffusion model are learnt depending on the current time step within the walking cycle.

The step by step process is as follows. First, for each posture $\hat{\psi}_i^t$ of each training

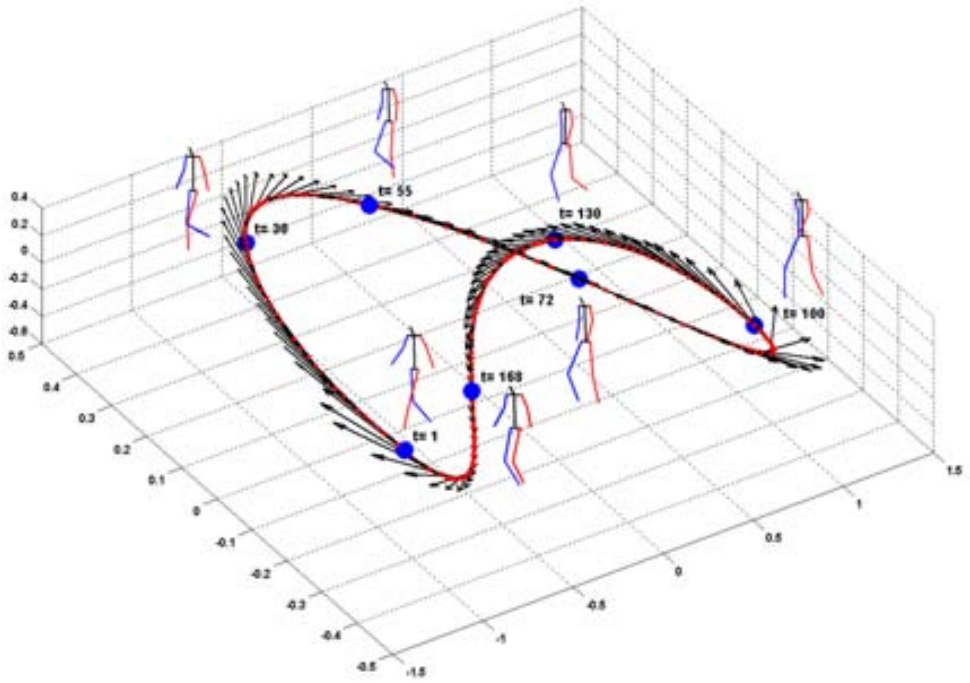


Figure 4.13: Sampled postures at different time steps, and learnt direction vectors $\overline{\mathbf{v}}_t$ from the mean performance for the walking action. Note that the vectors have unit length and have been rescaled for visualization purposes.

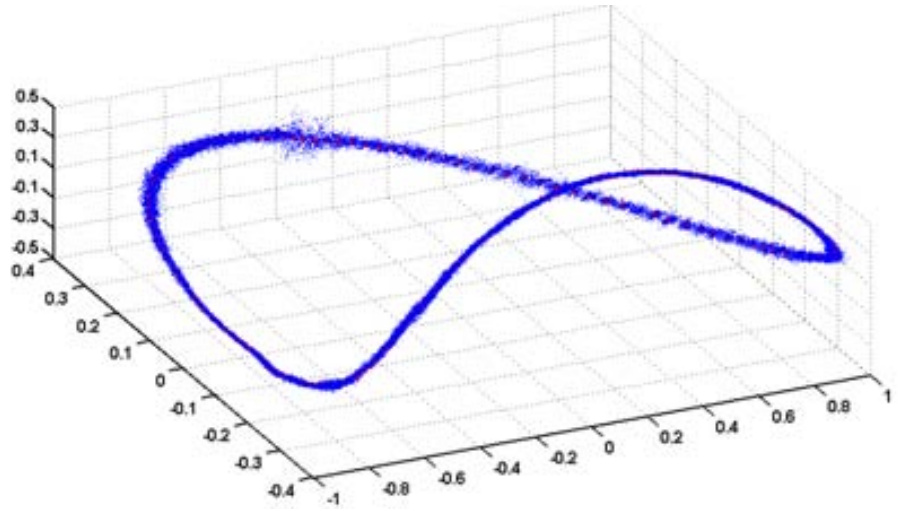


Figure 4.14: Prediction expected error at each step of the mean performance.

performance $\hat{\Psi}_i$, we predict a posture $\breve{\psi}_i^{t+1}$ for time $t+1$ using the velocity observed in the mean performance at the corresponding time instant, i.e.

$$\breve{\psi}_i^{t+1} = \hat{\psi}_i^t + (\bar{\psi}_{t+1} - \bar{\psi}_t), \tag{4.28}$$

where $(\bar{\psi}_{t+1} - \bar{\psi}_t)$ is the observed velocity from the mean performance at time t.

Then, we compute the error between the predicted posture $\breve{\psi}_i^{t+1}$ and the real posture from the training set at time $(t+1)$ as $e_i^t = |\hat{\psi}_i^{t+1} - \breve{\psi}_i^{t+1}|$. Doing this for all the postures from all the training performances we end up with P error measures per each time step t, i.e. $\mathbf{e}_t = (e_1^t, ..., e_P^t)$. Then, we characterize the error committed by the constant velocity model by learning the error covariance matrix for each time step, i.e.:

$$\Sigma_t = \mathbb{E}\left[(\mathbf{e}_t - \mathbb{E}(\mathbf{e}_t))\,(\mathbf{e}_t - \mathbb{E}(\mathbf{e}_t))^{\mathrm{T}}\right], \tag{4.29}$$

where Σ_t is the covariance matrix computed at time t, $\mathbf{e}_t$ is the error committed by the assumed first order motion model for all the training sequences, and $\mathbb{E}(\bullet)$ is the expectation of a distribution. The computed covariance matrices Σ_t are used for characterizing a Gaussian diffusion model in the stochastic search process in our tracking approach. Fig. 4.14 shows 100 samples from each Gaussian distribution for the diffusion model centered at their corresponding posture from the mean performance. Hence, we characterize different distributions for each posture from the mean performance. This results in an adaptive-noise term within the particle filter which improves the efficiency of this stochastic search process as opposed to fixed-diffusion models.

Finally, the action-specific motion model Γ_{A_k} is defined as

$$\Gamma_{A_k} = \{\bar{\Psi}, \sigma_t, \overline{\mathbf{v}}_t, \Sigma_t\}, \;\; t = 1..F, \tag{4.30}$$

where $\bar{\Psi}$ is the mean performance for the action A_k, and $\sigma_t, \overline{\mathbf{v}}_t, \Sigma_t$ correspond to the computed standard deviation, mean direction of motion and covariance matrix of the error at time step t, respectively.

In Fig. 4.15, we plot the computed mean performance (solid red line) and standard deviation (dashed black line) for the jumping, kicking, sitting, squatting and tumbling actions. Only the first three dimensions of the *aSpace* are shown indicating the amount of variation from the original data explained by each.

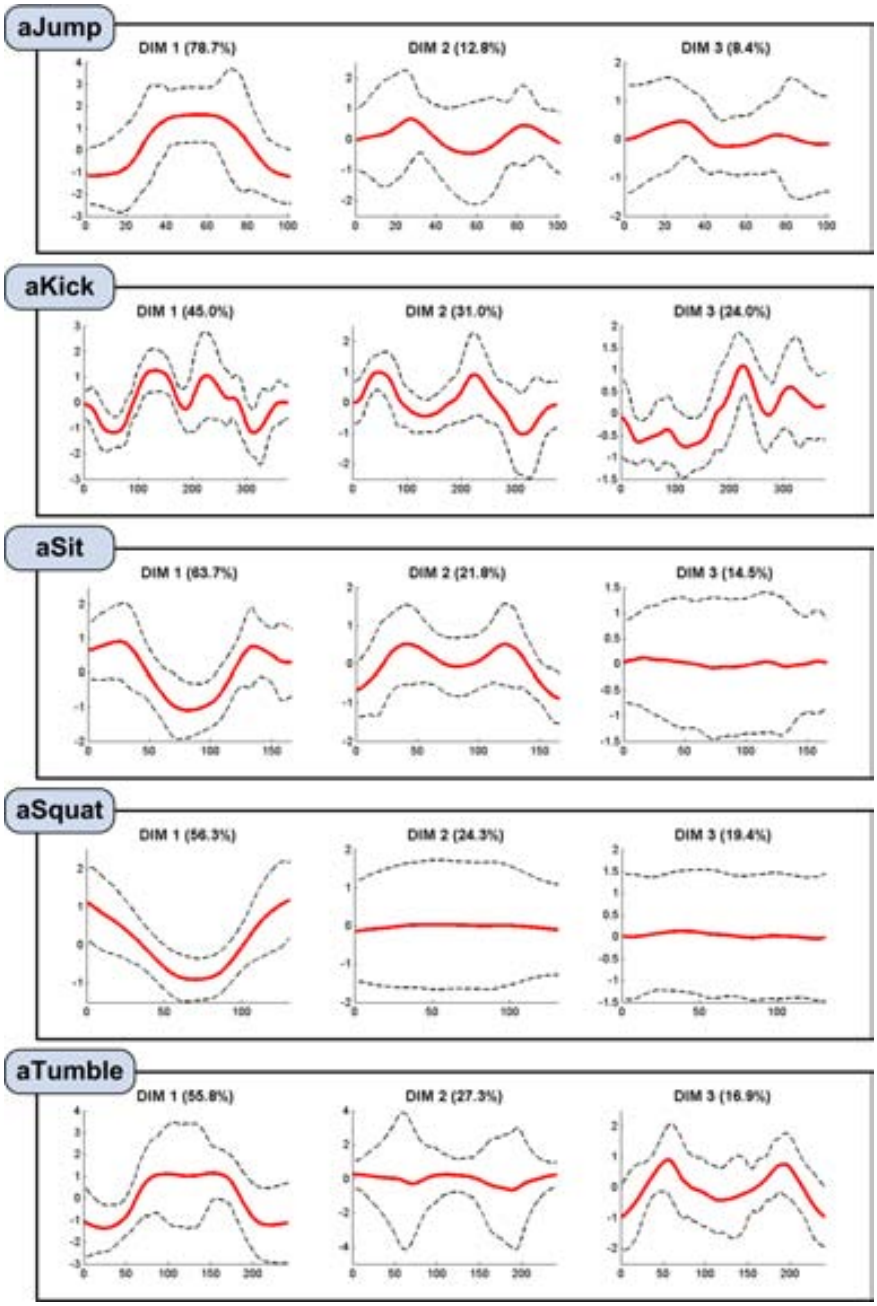


Figure 4.15: Learnt mean performance $\bar{\Psi}$ and standard deviation σ_t for the jumping, kicking, sitting, squatting and tumbling actions.

Chapter 5

The Tracking Framework

In this chapter, we describe the probabilistic framework used to face the body tracking problem. As stated before, the tracking is formulated as a Bayesian inference task, in which the *posterior* distribution over the model parameters is estimated at each time step, given the evidences available up to that moment. Therefore, in the first place, we will introduce the Bayesian filtering approach in general terms. Subsequently, we will give the basis of Particle Filtering as a technique to approximate the *pdf* over the human body model parameters at each time step, and its application to human body tracking. In the third place, we will detail the use of the learnt action models within this framework to improve tracking performance.

5.1 Introduction to Bayesian Filtering

In visual tracking, the process of sequentially estimating the parameters of a model of a target over time from visual data is known as a model-based tracking task.

Typically, model-based visual tracking processes follow the cycle represented as a flow chart on Fig. 5.1 [61]. The state of the object for the next time step is projected forward according to a dynamical model and all the information extracted up to the current step. Then, the model is projected into the image plane in order to match the prediction to the image and establish a measure of fitness between the predicted state and the image data. This steps may be repeated until the model state successfully matches the image data for each frame, and finally the predictions are refined and a new state is calculated.

In this work, the problem of estimating the full-body's 3D model parameters over time is faced as a model-based tracking approach formulated as a Bayesian filtering problem.

The Bayesian formulation of the problem consists of a probabilistic inference task whose aim is to estimate the *posterior* pdf of the model parameters at each time step (the state[1] of the tracked object) from a sequence of measurements available (image

[1]The state of an object defines a particular configuration for a given representation for such an object.

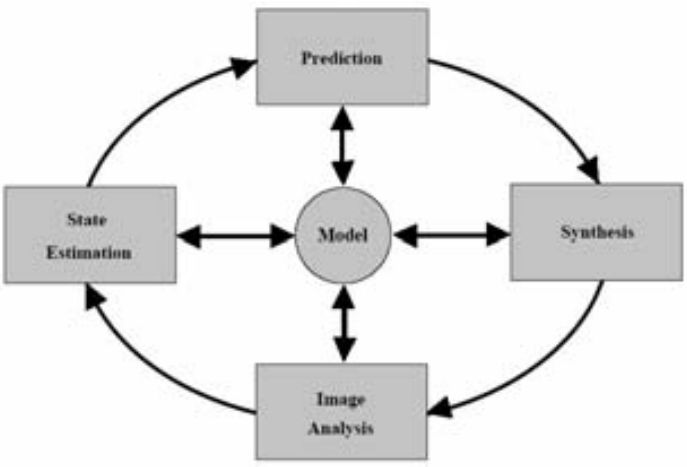


Figure 5.1: Model based tracking cycle [61].

data) up to that moment. Hence, the *posterior* pdf represents the sum of knowledge regarding the state of the tracked object from frame to frame.

The key idea of Bayes filters is to sequentially estimate the beliefs over the state space conditioned on the measurements (i.e. the images) obtained at each time step. The state at time t is represented by random variables ϕ_t, and the uncertainty about the real state of the object is represented by a probability distribution over the model parameters ϕ_t given the evidences available up to that moment (i.e. a sequence of images $\mathbf{I}_t$), hereafter the *posterior* pdf $p(\phi_t|\mathbf{I}_t)$. In other words, the *posterior* pdf provides an answer to the question: "What is the probability that the model is parametrized as ϕ_t given the history of images is $\mathbf{I}_t$, for all possible parametrizations of ϕ_t ?".

The Bayesian formulation for the tracking problem (Eq.(5.1)) is derived as follows:

The posterior $p(\phi_t|\mathbf{I}_t)$ at time t can be expressed as a marginalization of the joint posterior probability $p(\vec{\phi_t}|\mathbf{I}_t)$ over all parameters $\vec{\phi_t} = [\phi_0, ..., \phi_t]$, given all the images $\mathbf{I}_t$ up to that moment:

$$p(\phi_t|\mathbf{I}_t) = \int p(\vec{\phi_t}|\mathbf{I}_t)\, d\tilde{\phi}_{t-1}.$$

By introducing a first-order Markov assumption, i.e. the state ϕ_t at time t is assumed to depend only on the state ϕ_{t-1} at time $t-1$, the previous expression can be reduced to:

$$p(\phi_t|\mathbf{I}_t) = \int p(\phi_t, \phi_{t-1}|I_t, \mathbf{I}_{t-1})\, d\phi_{t-1}.$$

Now, using Bayes' rule, and assuming independence from frame to frame, we reformulate the equation as:

$$p(\phi_t|\mathbf{I}_t) = \int \frac{p(I_t|\phi_t, \phi_{t-1}))\, p(\mathbf{I}_{t-1}|\phi_t, \phi_{t-1})\, p(\phi_t, \phi_{t-1})}{p(I_t)\, p(\mathbf{I}_{t-1})}\, d\phi_{t-1}.$$

Taking into account that I_t conditioned on ϕ_t is independent of ϕ_{t-1}, and $\mathbf{I}_{t-1}$ is independent of ϕ_t, the above expression can be further simplified to:

$$p(\phi_t|\mathbf{I}_t) = \int \frac{p(I_t|\phi_t)}{p(I_t)}\, p(\phi_t|\phi_{t-1})\, \frac{p(\mathbf{I}_{t-1}|\phi_{t-1})\, p(\phi_{t-1})}{p(\mathbf{I}_{t-1})}\, d\phi_{t-1}.$$

Notice that $k = 1/p(I_t)$ is constant with respect to the model parameters ϕ. By applying Bayes' rule again to the last term of the integrand, we obtain

$$\begin{aligned} p(\phi_t|\mathbf{I}_t) &= \int k\, p(I_t|\phi_t)\, p(\phi_t|\phi_{t-1})\, p(\phi_{t-1}|\mathbf{I}_{t-1})\, d\phi_{t-1} \\ &= k\, p(I_t|\phi_t) \int p(\phi_t|\phi_{t-1})\, p(\phi_{t-1}|\mathbf{I}_{t-1})\, d\phi_{t-1}. \end{aligned}$$

Thus, using the Bayes' rule, we formulate the computation of our model parameters over time as [7]:

$$p(\phi_t|\mathbf{I}_t) = k\, p(I_t|\phi_t) \int p(\phi_t|\phi_{t-1})\, p(\phi_{t-1}|\mathbf{I}_{t-1})\, dt\,, \tag{5.1}$$

where ϕ_t represents a particular pose of the human body at time t, $\mathbf{I}_t$ is the image sequence up to time t, k is a normalizing factor, $p(I_t|\phi_t)$ is the *likelihood* of observing the image I_t given the parametrization ϕ_t of our model at time t, and finally $p(\phi_t|\phi_{t-1})$ is the *temporal prior*, or dynamic model in this work.

Eq. (5.1) can be split in two differentiated parts which divide the estimation problem in two steps, i.e. the term outside the integral corresponds to the *update* step, and the whole integral can be referred to as the *prediction* step. The distribution $p(I_t|\phi_t)$ is the *likelihood* of observing the image I_t given the model parameters ϕ_t. The prediction part of the equation, i.e. the integral, can be in turn understood as the product of two terms: $p(\phi_{t-1}|\mathbf{I}_{t-1})$ which is the posterior pdf at the previous time step, and the *temporal prior* $p(\phi_t|\phi_{t-1})$ or human dynamical model in this work which propagates the posterior distribution from time $t-1$ to time t. In other words, the dynamical model defines how human postures evolve over time.

Thus, the aim is to recursively estimate the parameters of the human body model ϕ_t at time t given the sequence of images $\mathbf{I}_t$ up to that moment by computing the posterior pdf $p(\phi_t|\mathbf{I}_t)$ over the model parameters at each time step. Unfortunately, Eq.(5.1) relies on an integral which cannot be analytically calculated unless strong assumptions about Gaussianity and linearity on the involved distributions are made. When both the temporal prior and the likelihood pdfs follow a Gaussian distribution, the Kalman filter (KF) [38] provides the optimal solution yielding an also Gaussian distribution of the posterior. The Kalman filter can also be seen as an optimal recursive estimator for linear systems, considering the likelihood and the posterior linear functions which define the fitness of the model to the data and the transition from state to state. The Extended Kalman Filter (EKF) [7, p. 106] improves the performance of the simple KF on non-linear systems by linearizing the non-linear models before applying the KF algorithm, thus obtaining Gaussian posteriors too. A more refined approach is the Unscented Kalman Filter (UKF) [36] which uses a set of dis-

cretely sampled points to parametrize the mean and covariance of the posterior pdf. However, there is still an assumption about the Gaussianity of the posterior distribution, and this may be a too strong assumption for applications such as ours where the involved models are highly non-linear leading to a multi-modal and non-Gaussian posterior distribution.

Alternatively, such distributions can be propagated over time by using particle filtering techniques [5], which constitute the most general class of filters which are based on Monte Carlo integration methods. The current posterior distribution is approximated by a weighted set of random samples or particles. The new posterior pdf is computed based on these particles and their weights, and no assumption about Gaussianity on any of the involved distributions is needed.

5.2 Particle Filtering in human motion tracking

As stated before, the recursive Bayesian filter provides the theoretical optimal solution to the tracking problem represented by Eq.(5.1). However, as illustrated in section 1.3, human body tracking presents several characteristics which makes the tracking task difficult. Mainly, due to occlusions and self-occlusions, 2D-3D projection ambiguities, changes in appearance and shape and non-linearity of human motion, the involved distributions results highly non-Gaussian.

Therefore, given that Eq.(5.1) cannot be analytically solved for non-Gaussian distributions, we can approximate the true posterior distribution $p(\phi_t|\mathbf{I}_t)$ instead by means of a *particle filter* [34, 5]. Particle filters belong to a set of simulation-based methods named Sequential Monte Carlo methods [19] which provide a convenient and useful approach for computing the posterior distributions.

In a particle filtering framework, the posterior distribution at time t is represented by a weighted set of samples or particles. Each particle corresponds to a particular human posture, and has its own probability of being propagated over time depending on its weight. If a particle is selected to be propagated at time t, a transition model or *dynamic model* is used to predict the new location in the parameter space at time $t+1$, i.e. the new configuration at the following time step.

Using a particle filter to solve the estimation problem is motivated mainly because:

- it can approximate non-Gaussian *posterior* pdfs: human limbs motion suffer from large non-uniform accelerations while performing an action, resulting in non-linear human dynamics. Moreover, self-occlusions, 2D-3D ambiguities and singularities make the density function to be estimated highly non-Gaussian and multi-modal.

- it provides us a principled way to incorporate *a priori* knowledge about human motion dynamics by means of a dynamic model: human dynamics happen to be highly correlated, so by introducing this knowledge into the tracking process we can restrict the search space only to the most plausible configurations, making the tracking more robust and efficient.

- it establishes a method for naturally keeping multiple hypotheses about the performed motion: due to the inherent ambiguity between 2D projections of

human postures, there are a lot of actions which share similar postures, and a lot of postures which share similar 2D projections. Thus, it is necessary to enable a way for considering multiple possible configurations which can be estimated from the image data.

In this work, each particle ϕ_t^s represents a particular 3D body configuration. Then, a normalized weight $\overline{\pi}_t^s$ is assigned to each particle depending on how likely is the body posture that it represents to be found on the image I_t.

The sequential estimation method works as follows: the posterior pdf at time $t-1$ is represented by a weighted set of N samples, i.e. $\left\{\phi_{t-1}^s, \overline{\pi}_{t-1}^s; s = 1:N\right\}$. Each weight $\overline{\pi}_{t-1}^s$ corresponds to the normalized likelihood value for each sample ϕ_t^s. Then, the posterior pdf for the next time step is obtained according to the following procedure:

1. N new particles are sampled from the posterior pdf at time $t-1$ using Monte Carlo sampling, i.e. the normalized weight $\overline{\pi}_{t-1}^s$ of each particle is equal to the probability of selecting that particular particle s as the new sample. It can be proven that when $N \to \infty$ the sampling is equivalent to the true posterior distribution $p(\phi_{t-1}|\mathbf{I}_{t-1})$.

2. Then, each new sampled particle is propagated in time by applying the dynamic model to it. In other words, the *temporal prior* $p(\phi_t|\phi_{t-1})$ is sampled, thus obtaining a set of N new particles for the next time step, i.e. $\{\phi_t^s\}$. These N new samples now represent the *prior* distribution over ϕ_t, i.e. $p(\phi_t|\mathbf{I}_{t-1})$. This step usually comprises some sort of *diffusion* process where some noise is added to the prediction in order to represent the growth of uncertainty from frame to frame about the model parameters, so the space of solutions is sufficiently explored and non-possible configurations about the human body pose are omitted.

3. For each particle ϕ_t^s its weight π_t^s is computed by evaluating the likelihood function $p(I_t|\phi_t)$ given that particular sample and image. The normalized likelihood $\overline{\pi}_t^s$ is computed as $\overline{\pi}_t^s = p(I_t|\phi_t^s)/\sum_{s'=1}^{N} p(I_t|\phi_t^{s'})$. Finally, the current posterior pdf is approximated by the sampled particles plus its corresponding normalized weights, i.e. $\{\phi_t^s, \overline{\pi}_t^s; s = 1:N\}$. In other words, the likelihood function evaluates the fitness of the predicted particle to the measurements available which determines the particle's weight, and thus, its probability of being propagated to the next time step.

Figs. 5.2 and 5.3 illustrate the procedure described above in a more graphical manner. Fig. 5.2 shows the key idea behind the approximation of the posterior (solid line) by a weighted set of samples (full circles below the graphic). The solid line on the upper part, represents the true posterior distribution $p(\phi_{t-1}|\mathbf{I}_{t-1})$ over the parameters ϕ_t of a 1D model. On the bottom of the figure, we find the scheme which describes how the true pdf is approximated by particles or samples: each sample is represented by a circle whose size and position correspond to its normalized weight and its position in the parameter space respectively.

In addition, Fig. 5.3 illustrates the propagation procedure of the posterior from frame to frame. The process has been divided in the three main steps previously

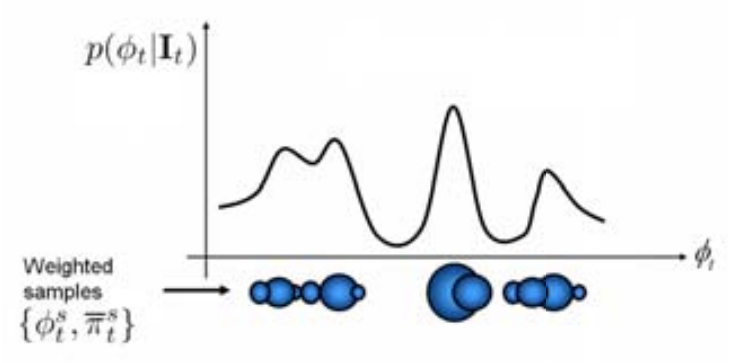


Figure 5.2: Representation of the posterior by a weighted set of samples $\tau_t^s\ \phi_t^s; s = 1 : N$. Circles represent the position in the parameter space of each particle τ_t^s (center) and its weight (diameter).

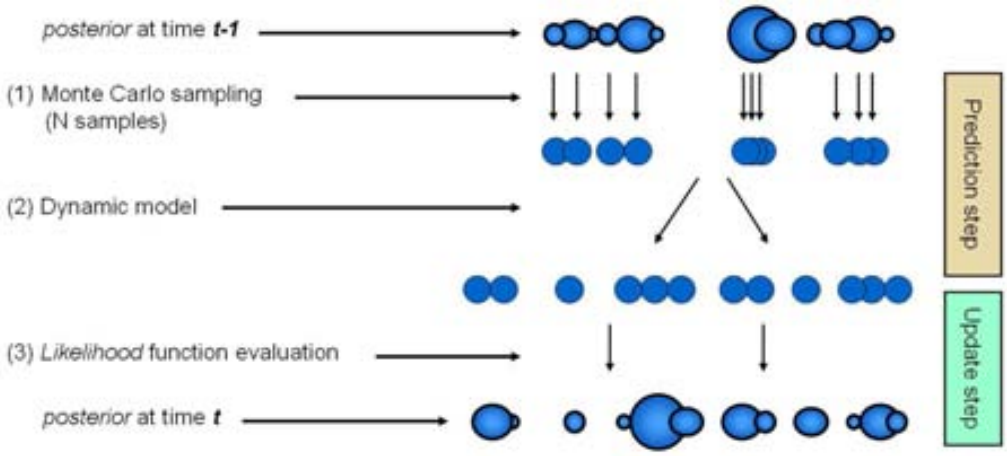


Figure 5.3: Propagation procedure scheme of the posterior pdf from frame to frame.

described. The first two steps, i.e. sampling from the previous posterior and applying the dynamic model, belong to the *prediction* stage, while the evaluation of the likelihood function and the assignment of normalized weights to the particles constitute the *update* step. Big dots on the top of the figure represent the posterior at time $t\ \ 1$ in the fashion described by Fig. 5.2. Scanning top-down the rest of the figure, we can appreciate the drawing of samples with replacement from the previous posterior (1st step), thus obtaining a new set of N unweighted samples. After, the dynamic model is applied to each new sample (2nd step). Thereby the particles are propagated forward to the next time step. Finally, the likelihood function is evaluated for each particle (3rd step), and the new posterior representation at time t is obtained.

5.2.1 Final state estimation

We have seen that particle filters are useful for approximating and propagating the posterior distribution about the target model parameters, i.e. ϕ_t, at each time step. However, for many applications, we may need to take a decision and determine a particular state of our tracked object at a given time step. There are several criteria one could follow. For example, one option is to select the particle that maximizes the likelihood at each time step:

$$\hat{\phi}_t = \{\phi_t^{s_0} \,|\, s_0 = \arg\max_s \overline{\pi}_t^s\}, \tag{5.2}$$

where $\hat{\phi}_t$ represents the estimated state of our parameters at time t. However, this approach has several drawbacks. On the one hand, if not enough particles are used, the estimations for successive time steps tend to "jump around" since the approximation of the posterior pdf is too erroneous and full of spurious modes. On the other hand, an image-based likelihood may present multiple-modes, high peaks, etc. due to changes in the appearance model, shape and appearance ambiguities, etc., making the maximization likelihood criterion not stable and reliable over time. Consequently, an approach that takes all the particles into account is preferred. Thereby, we use the expected value of the distribution as the estimated state for each time step, i.e.:

$$\hat{\phi}_t = \mathcal{E}[\phi_t] = \sum_{s=1}^{N} \overline{\pi}_t^s \phi_t^s. \tag{5.3}$$

Notice, that Eq. (5.3) provides a good summary for the distribution if the posterior is uni-modal. Otherwise, even though the estimated state may be totally erroneous, the distribution is properly propagated over time.

5.2.2 Choosing the number of particles

The use of particles for representing the posterior brings a very powerful tool for representing complex posterior distributions, since they don't involve any assumptions about Gaussianity, uni-modality, or linearity. However, the number of parameters to be computed at each time step is huge in comparison to other methods such as the basic KF where the posterior is represented by only its mean and covariance. For these reasons, a drawback of particle filters is the high computational cost which directly depends on the number of particles used to represent the posterior. Hence, the computational complexity for computing each time step is $\mathcal{O}(N)$, being N the number of particles used.

According to McCormick and Isard [50], the number of needed particles depends on both the dimensionality of the search space, and the shape of the distribution. In our case, we aim to estimate the configuration of a high articulated structure, which spans a search space with a rather large volume. Moreover, human dynamics results in complicated shapes of the posterior distribution. Therefore, a large value for N is needed, which outcomes high computational costs for properly estimating the model's parameters. On the one hand, too low values for N will lead to either not enough

accuracy while tracking the target, or even loosing it eventually. On the other hand, too high values for N nevertheless result in unnecessarily wasted computational cost, which is a critical issue for real-time tracking applications.

Therefore, setting the number of needed particles is a though task, although in [50] a relation for N regarding the dimensionality of the space and the *sharpness* of the posterior distribution is derived:

$$N \geq \frac{\mathcal{D}_{min}}{\beta^b},$$

where the exponent b corresponds to the dimensionality of the parameter space, $\mathcal{D}_{min}$ stands for the minimum acceptable number of particles to survive the sampling, and β is the *survival rate*, being $\beta \ll 1$. Basically, the survival rate β is related to the shape of the posterior and prior distributions, and measures how well the posterior is predicted at each time step. Notice that as β decreases, N grows up exponentially. Generally, distributions with sharp peaks and noise lead to lower values of β, thus demanding of a higher number of particles to be properly represented and propagated.

It is remarkable that there exists a trade-off condition between the number of particles used and the accuracy of the estimations made, or in other words, more accuracy accounts for more computational cost. Nevertheless, it is worth saying that implementations of particle filters are fully parallelizable due to their particle nature. The calculations for the propagation, diffusion and evaluation of each particle could be performed in parallel since there are no dependencies between them.

As a result, within this work, the number of particles is set empirically by running several tracking tests varying the number of particles used and analyzing its performance.

5.3 Using the Action Models to improve tracking performance

In the previous section, we've introduced a general PF framework applied to human body tracking. However, although it has been widely used within this application context [18, 50, 71], it suffers from a significant number of problems as stated in section 2.2. This work is aimed to leverage these problems by introducing two main contributions within the prediction step of the PF: an efficient dynamic model for predicting human postures, and the constraint of the state space to the most plausible solutions. Thus, particle wastage is avoided and robustness is added to the overall tracker compared to a standard PF with a generic motion prior.

As a result, the prediction stage works as follows: each selected particle ϕ_t^n is propagated over time according to these 3 steps:

1. Identify which part of the learnt mean performance $\bar{\Psi}$ from Eq. (4.25) is more similar to ϕ_t^n. Thus, we probabilistically match the particle ϕ_t^n and a subsequence of the last estimated motion history against all the subsequences from $\bar{\Psi}$ having the same length.

2. Propagate the particle over time by means of a 1st order motion model and a Gaussian diffusion term whose parameters are retrieved from the action model at the time instants that matched the mean performance.

3. Constrain the possible solutions to feasible postures according to the learnt action model at the matched time instant. If the predicted posture is considered to be invalid, the prediction is wasted, and a new particle is stochastically selected from the particle set representing the posterior pdf. Then, start over with step 1 until a prediction is accepted.

In the following subsections a detailed explanation of each step is given. In addition, the full posterior's propagation process can be found in Algorithm 5.1.

5.3.1 Probabilistic Match

Our probabilistic matching approach aims to identify which part of the mean performance is more similar to the current particle. On the one hand, we define the subsequence of estimated motion to be matched at time t, by concatenating the currently selected particle with the last $(d-1)$ estimated postures of the motion history, i.e. $\Phi_t^n = (\hat{\phi}_{t-d+1}, ..., \hat{\phi}_{t-1}, \phi_t^n)$.

On the other hand, we define a motion subsequence of length d from the mean performance at time instant i as $\bar{\Psi}_i = (\bar{\psi}_{i-d+1}, ..., \bar{\psi}_{i-1}, \bar{\psi}_i)$.

Then, abusing the notation, we define a similarity measure between 2 motion subsequences of length d within the *aSpace*, namely $\bar{\Psi} = \{\bar{\psi}_1, ..., \bar{\psi}_d\}$ and $\Phi = \{\phi_1, ..., \phi_d\}$, as

$$S(\bar{\Psi}, \Phi) = \exp\left(-D_M(\bar{\Psi}, \Phi)\right) \left[\frac{(\overline{\mathbf{v}}_{\bar{\Psi}} \cdot \overline{\mathbf{v}}_{\Phi}) + 1}{2}\right]^{\alpha}, \tag{5.4}$$

where $\cdot$ stands for the dot product between the average motion direction vectors $\overline{\mathbf{v}}_{\bar{\Psi}}$ and $\overline{\mathbf{v}}_{\Phi}$ from Eq.(4.27), and D_M is the sum of the Mahalanobis distances within the *aSpace* space between each subsequences' postures $\bar{\psi}_j$ and ϕ_j, $j = 1..d$.

This similarity measure is composed of two terms. The exponential term accounts for the spatial proximity between postures within the *aSpace*, while the dot product term expresses similarity w.r.t. directions of motion across time, regardless the body postures exhibited. The exponent α is introduced to balance the importance of each term in the final similarity computation: high values for α leads to high similarity between sequences sharing the same motion direction, while low values will take more into account the position of their postures within the *aSpace*. Therefore, this similarity metric is a trade off between sequences that exhibit similar motion directions and sequences with similar postures within the *aSpace* according to a Mahalanobis distance criterion. The key idea is that close sequences which follow the same direction get high scores, while sequences that do not match in motion direction or position are given low similarity scores. Notice also, that $S(\bar{\Psi}, \Phi) \in [0, 1]$.

Finally, we probabilistically match Φ_t^n to a subsequence from the mean performance $\bar{\Psi}$ by computing the similarity s_i^t between Φ_t^n and all the possible subsequences $\bar{\Psi}_i$ from the mean performance as $s_i^t = S(\bar{\Psi}_i, \Phi_t^n)$, $i = 1..F$, and then randomly se-

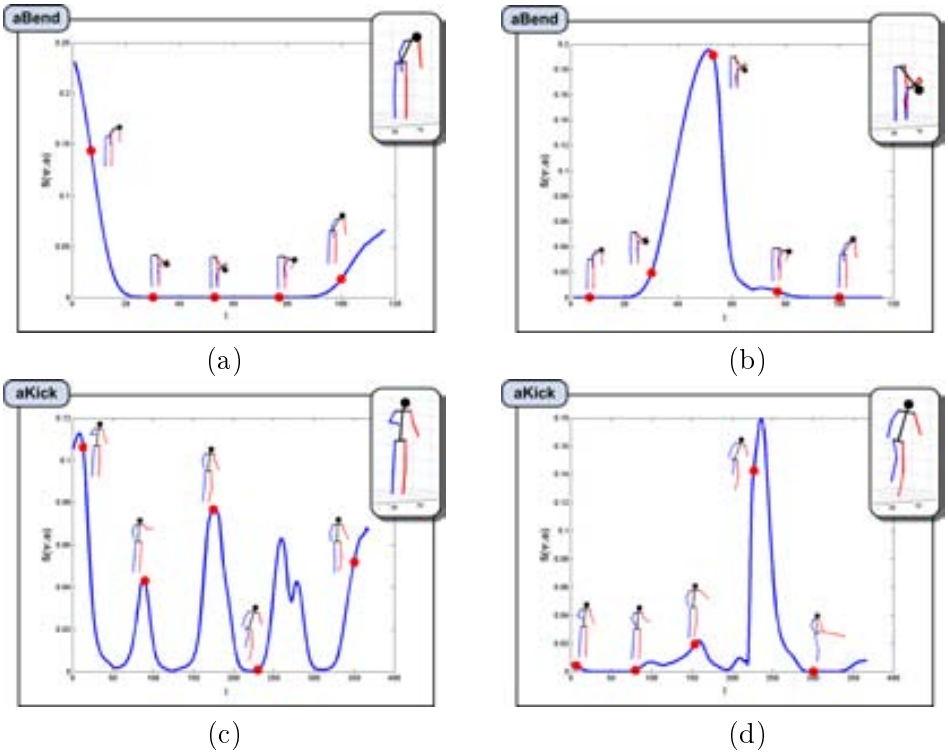


Figure 5.4: Similarity between motion subsequences of the aBend and aKick actions. In (a) and (b) a motion subsequence from the aBend action is compared to all the subsequences from the mean performance. Similarly, in (c) and (d) the same comparison is done for the aKick action.

lecting a matching sequence $\bar{\Psi}_i$ with probability

$$p(\bar{\Psi}_i|\Phi_t^n) = \frac{s_i^t}{\sum_{i=1}^{F} s_i^t}, \quad i = 1..F, \tag{5.5}$$

where F is the total number of postures from the mean performance.

Alternatively, one could opt for a deterministic matching approach by selecting the subsequence $\bar{\Psi}_i$ which maximizes the similarity s_i^t. However, in case $p(\bar{\Psi}_i|\Phi_t^n)$ is multimodal, only the mode with highest similarity would be selected, thus loosing the remaining ones. Furthermore, the extension to multiple action models is straightforward given the probabilistic definition of the matching process. Hence, the multimodality introduced in $p(\bar{\Psi}_i|\Phi_t^n)$ by similar postures shared between different action models is kept by the approach. In such case, the cost of matching for a time step of the PF tracker is $N \cdot \sum_{i=1}^{K} F_i$, where N is the number of particles used, and F_i is the number of postures of the mean performance for the action i out of a total of K actions.

Figure 5.4 illustrates the behaviour of the given similarity measure. Hence, $S(\bar{\Psi}, \Phi)$ has been computed for 2 different subsequences of the aBend and aKick actions against all possible subsequences from their mean performance. For visualization purposes, the final posture of the matched subsequence is overlaid on the upper right corner of each figure. Also, some selected postures from the mean performance are plotted. For instance, in Fig. 5.4.(a) a motion subsequence from the beginning of the bending action is compared to all the subsequences from its mean performance. As expected, from the one hand, higher scores are obtained on the very first postures from the mean performance which correspond to an almost standing posture going towards the floor. On the other hand, totally bent postures in the middle of the performance get a score of 0 because they are completely different from the compared subsequence. Then, the similarity grows up again by the end of the action given that these postures correspond to an almost standing posture. Interestingly, although they are indeed very similar postures, their score is much lower than in the beginning of the action due to the difference w.r.t. their direction vectors. In Fig. 5.4.(b) a posture from the middle of the action is compared instead. Analogously, the similarity of subsequences from the aKick action is shown in Fig. 5.4.(c) and (d).

5.3.2 Dynamic Model definition

Within the prediction step of the PF, we project forward the particle set representing the posterior at time $(t-1)$ by drawing new samples ϕ_t^n from the dynamic model $p(\phi_t|\phi_{t-1})$ of Eq.(5.1). Following the approach described by Sidenbladh in [71], we extend the state space to store the history $\Phi_{t-1} = (\phi_{t-1}, ..., \phi_{t-d})$ of the last d estimated postures, and sample from the conditional distribution $p(\phi_t|\Phi_{t-1})$, instead of considering only the last posture ϕ_{t-1}. Finally, given Φ_{t-1}^n, new samples ϕ_t^n are computed as

$$\phi_t^n = \phi_{t-1}^n + (\bar{\psi}_{i+1} - \bar{\psi}_i) + \eta(\Sigma_{i+1}), \tag{5.6}$$

where i is the index of the motion subsequence from the mean performance which probabilistically matched Φ_{t-1}^n according to Eq.(5.5). Hence, $(\bar{\psi}_{i+1} - \bar{\psi}_i)$ corresponds to the velocity present in the mean performance $\bar{\Psi}$ (Eq.(4.25)) at the matched subsequence, and $\eta(\Sigma_{i+1})$ is a zero-mean Gaussian noise function with covariance Σ_{i+1} learnt computing Eq.(4.29). Therefore, by sampling from the prior $p(\phi_t|\Phi_{t-1})$ a particle is propagated by a first order motion model that uses the learnt velocity and error's covariance from the matched subsequence $\bar{\Psi}_i$ of the mean performance. Thus, a priori knowledge on human motion is used to guide the exploration of the state space.

It is worth mentioning that as a result, on the one hand we achieve a more efficient use of the particle set compared to more generic dynamic models as long as both the training and testing sequences belong to the same class of motion. On the other hand, a poor estimate could be obtained from the mean performance in case the motion to be tracked is too different from the learnt model, or its framerate differs too much from the framerate of the training sequences. While the former would require training the system with examples of this kind of motion, the latter is accommodated by the present approach as long as the framerate difference is not too large. This is due to the probabilistic matching approach, and the nature of the particle filtering

framework. Hence, although in the presence of substantial framerate differences between the testing sequence and the mean performance, the final matching probability is still maximum between the most similar subsequences, since its similarity scores are normalized in Eq. (5.5). Then, the Gaussian diffusion term from Eq.(5.6) and the stochastic nature of the particle filtering framework contribute in accommodating the prediction error, up to a certain limit.

5.3.3 Constrained solution space

After new postures are sampled, we apply a filtering step which discards predicted particles which do not correspond to feasible human postures according to our action model. Hence, given the matched subsequence $\bar{\Psi}_i = (\bar{\psi}_{i-d+1}, ..., \bar{\psi}_{i-1}, \bar{\psi}_i)$ we determine that a particular posture ϕ_t lies within the variation bounds accepted for the action if

$$|\phi^n_{t,j} - \bar{\psi}_{i+1,j}| < k_1 \cdot \sigma_{i+1,j}, \quad \forall j = 1..b_{filt}, \tag{5.7}$$

where $\phi^n_t = (\phi^n_{t,1}, ..., \phi^n_{t,b})^T$ is the b-dimensional predicted particle representing a particular body posture in the *aSpace*. In addition, $\bar{\psi}_{i+1} = (\bar{\psi}_{i+1,1}, ..., \bar{\psi}_{i+1,b})^T$ is the next posture to $\bar{\Psi}_i$, i.e. the immediately following posture from the mean performance which probabilistically matched the current particle according to Eq.(5.5). Then, $\sigma_{i+1} = (\sigma_{i+1,1}, ..., \sigma_{i+1,b})$ stands for the learnt standard deviation for the i-th posture of the matched subsequence. Subsequently, k_1 is a scale factor for the variance allowed. Hence, too small values for k_1 lead to accepting only postures almost equal to the ones stored in the mean performance $\bar{\Psi}$. Typically we set k_1 to 3, thus including the 99.73% of variation of the training set.

Finally, b_{filt} determines the number of dimensions from the *aSpace* considered for filtering. Notice that $b_{filt} \leq b$, where b is the total number of dimensions in the *aSpace* (Eq.(4.8)). While the accuracy of the representation is a matter of as more dimensions the better, by not using all the b dimensions for filtering, we allow to track subtle motions which were not present in the training set, while filtering out non-likely postures according to the most important modes of variation found for the action. Hence, b_{filt} controls the trade off between generality and specificity of the filtering method. For instance, in our experiments for the walking action, we achieved better results setting $b_{filt} = 3$ and $b = 5$.

Therefore, the approach is as follows: given a selected particle ϕ^n_{t-1} to be propagated, we use the dynamic model defined in Eq. (5.6) to obtain a new sample. Then, if the new sample is not accepted according to Eq.(5.7), this particle is dropped and a new one is stochastically selected. Then, the propagation process is restarted until a prediction is accepted.

By removing the particles which are not accepted we modify the particle set representing the posterior distribution. Thus, after this process, they are not representing that distribution anymore, but a pruned version of the posterior pdf, since indeed, the posture filtering step can be seen as a pruning of the state space where particles live. An alternative to dropping rejected particles, which does not modify the convergence results of the PF, is to sample additional particles by importance sampling until enough accepted particles are achieved. However, such a method requires

computing the likelihood of each extra sampled particle, which is usually the most computationally expensive part of a PF framework. Consequently, being aware that we are sampling from a pruned version of the posterior pdf, in this work we implement the particle removal method and show that also good tracking results are obtained by dropping beforehand those predictions which are not likely to appear during the performance of a particular action.

5.4 Updating the predictions

Once the particle set $\{\phi_{t-1}^n\}$ has been propagated from $t-1$ to t, the predictions are updated assigning a weight to each particle corresponding to the fitness of the predicted posture to the evidences available at time t. This is done by evaluating the likelihood function $p(I_t|\phi_t)$ from Eq.(5.1) for each particle from the set $\{\phi_t^n\}$. Then, the computed weights are normalized, obtaining the *posterior* representation $\{\phi_t^n, \bar{\pi}_t{}^n\}$ at time t.

As discussed in chapter 2, there exists a great number of approaches in the literature that aim to find suitable likelihood functions that model well enough the probability that an image I_t explains the target state ϕ_t. However, many of them require to compute image-based measures for each particle to determine the probability of the image given the particle, which usually constitutes the most time consuming step of particle filter trackers. In addition, a lot of research effort has also been done on characteristic 2D point detectors [4, 8, 49, 79] and the extraction of statistically relevant features from 2D or 3D patches in images [44, 21, 48, 43, 37], suitable for activity recognition. Body limb detection methods also retains popularity in the vision community and indeed there exist many works which extract the whole 2D position of body joints from images with relative success [65, 47, 51].

Consequently, in this work we assume that there exists a method for detecting a rough approximate of the 2D position of some body joints given an input image. We also assume that this detection stage produces noisy results, since typically, only a reduced set of joints is observable at each time step. In particular, the head, hands and feet are easily visible, so finding an approximate of their 2D position in some frames might be a feasible task. Summarizing, we assume that this detection method is executed once per each frame, and that its output is noisy and incomplete. The output from this detection stage at time t is composed of a set of 2D image coordinates, and is denoted as $\mathbf{X}_t^{DS} = (x_1^{DS}, ..., x_D^{DS})$ where D stands for the number of detected joints.

Given the previous assumption, we compute each particle likelihood as follows.

First, we reconstruct the human body posture encoded by each particle. Hence, given a predicted particle ϕ_t^n in the *aSpace*, we project it back to the original 36-dimensional representation and divide each limb's direction cosines vector by its norm, so that the restriction $(\cos\theta_l^x)^2 + (\cos\theta_l^y)^2 + (\cos\theta_l^z)^2 = 1$ is satisfied. Then, we compute the 3D absolute positions of each joint j from the human body model and project it to the image plane according to the camera's projection matrix. As a result, a set of 2D image coordinates are obtained, which correspond to the projection of the

human body joints. The resulting vector is denoted as

$$\mathbf{X}_t = (x_1, ..., x_J), \tag{5.8}$$

where J is the number of virtual markers composing the human body model.

Finally, a mapping between detected and estimated joints is computed. For our tests, we used a simple Nearest Neighbor (NN) criteria for defining this mapping. Let's define the mapping function as

$$m_{j,d} = \begin{cases} 1 \ if & joint\ x_j\ matches\ with\ x_d^{DS} \\ 0 \ otherwise & \end{cases}. \tag{5.9}$$

Subsequently, we define a likelihood function based on a distance between the detected 2D positions from the detection stage, i.e. $\mathbf{X}_t^{DS}$, and the reconstructed and projected 3D joints positions encoded by the particle, i.e. $\mathbf{X}_t$. Formally, the likelihood of observing the evidences I_t given the predicted particle's ϕ_t^n corresponding posture is defined as

$$p(I_t|\phi_t) \propto e^{-\gamma \cdot \sum_{j=1}^{J} \sum_{d=1}^{D} m_{j,d} \cdot dist(x_j, x_d^{DS})}, \tag{5.10}$$

where *dist* stands for the Euclidean distance between the joint position x_j from the predicted posture and the detected joint position x_d^{DS} from image data, weighted by the mapping function $m_{j,d}$. Additionally, γ is a scale factor which determines the "peakiness" of the likelihood function, with a direct impact on the particle survival rate. Hence, the higher γ is, the higher is the difference between the probability of the most likely and the most non-likely postures. In our experiments, $\gamma = 80$ showed to be a good trade off for keeping a balanced particle set to represent the posterior pdf for the tested sequences.

In addition, it should be noted that this likelihood definition is mainly aimed for monocular image based tracking where 2D measurements from the image sequence are available. However, it could be easily employed in a multicamera setup, by using the 3D joint positions estimated from the individual 2D detections by triangulation. Hence, in such scenario, each particle likelihood could be computed analogously by evaluating $p(I_t|\phi_t)$ but using 3D joint positions when defining $\mathbf{X}_t$ and $\mathbf{X}_t^{DS}$.

Finally, this work assumes that the tracker has been already initialized, i.e. that we know the first d 3D body postures from the performed motion. Notice that this framework can be combined with direct 3D body posture inference methods that use 2D body shape information extracted from monocular video images [65]. Such methods perform well with postures having low 2D/3D ambiguity which are suitable to be used for initializing the tracker and recovering from critical failures.

The pseudo-code for propagating the posterior estimation over time is shown in Algorithm 5.1.

Algorithm 5.1 Pseudo-code of the posterior's propagation algorithm over time.

- **for** t=d, .., T

  - **for** n=1, .., N

    1. Select a particle to be propagated according to its weight
       * Draw $n' \sim \{1, ..., N\}$ such that $p(n' = i) = \bar{\pi}_{t-1}^{i}$, $i = (1, 2, ..., N)$
    2. Propagate the selected particle $\phi_{t-1}^{n'}$

       (a) Build $\Phi_{t-1}^{n'} = (\hat{\phi}_{t-d}, ..., \hat{\phi}_{t-2}, \phi_{t-1}^{n'})$

       (b) Probabilistically match $\Phi_{t-1}^{n'}$ with a motion subsequence $\bar{\Psi}_i$ of the same length from the mean performance

          i. Compute $s_i^{n'} = S(\bar{\Psi}_i, \Phi_{t-1}^{n'})$, $i = (1, 2, ..., T)$ using Eq.(5.4)

          ii. Stochastically select $\bar{\Psi}_i$ with matching probability $p(\bar{\Psi}_i|\Phi_{t-1}^{n'})$ given by Eq.(5.5)

       (c) Predict a new particle ϕ_t^n using the dynamic model, Eq.(5.6)
       * $\phi_t^n = \phi_{t-1}^{n'} + (\bar{\psi}_i - \bar{\psi}_{i-1}) + \eta(\Sigma_i)$

       (d) Constrain the solution space by filtering out non-feasible predictions according to Eq.(5.7)

          i. Compute deviation between the predicted particle ϕ_t^n and the last posture $\bar{\psi}_i$ from the matched sequence $\bar{\Psi}_i$ of the mean performance
          * $\Delta_t^n = |\phi_t^n - \bar{\psi}_i| = (\Delta_{t,1}^n, \Delta_{t,2}^n, ..., \Delta_{t,b_{filt}}^n)$

          ii. Check if the particle is accepted by the action model

          if $(\Delta_{t,j}^n < k_1 \cdot \sigma_{i,j})$ $\forall j = 1..b_{filt}$ (posture lies within the boundaries)
          * Accept the prediction ϕ_t^n for this particle and proceed on propagating the next particle.

          else
          * Drop the predicted particle ϕ_t^n and proceed to step (1).
          * Repeat the whole process until a prediction for particle with index n is accepted.

          endif

    3. Update step: Compute the likelihood of the prediction using Eq.(5.10)
       * $\pi_t^n \propto p(I_t|\phi_t)$

  - **endfor**
  - Compute normalized weights $\{\bar{\pi}_t\}$ such that $\sum_{n=1}^{N} \bar{\pi}_t^n = 1$

- **endfor**

Chapter 6

Experimental results

In this chapter, experimental results are presented to show the performance and behaviour of the different contributions of this work.

First, we present some results regarding the synchronization method for the motion training set. Then, we analyze the performance of the probabilistic matching methodology. In particular, we present some tests carried out using the similarity measure between motion subsequences in a gait recognition scenario.

Finally, we discuss on the methodology for estimating the parameters of the human body tracker supported by empirical tests, an show results of the overall tracking framework for several test motion sequences.

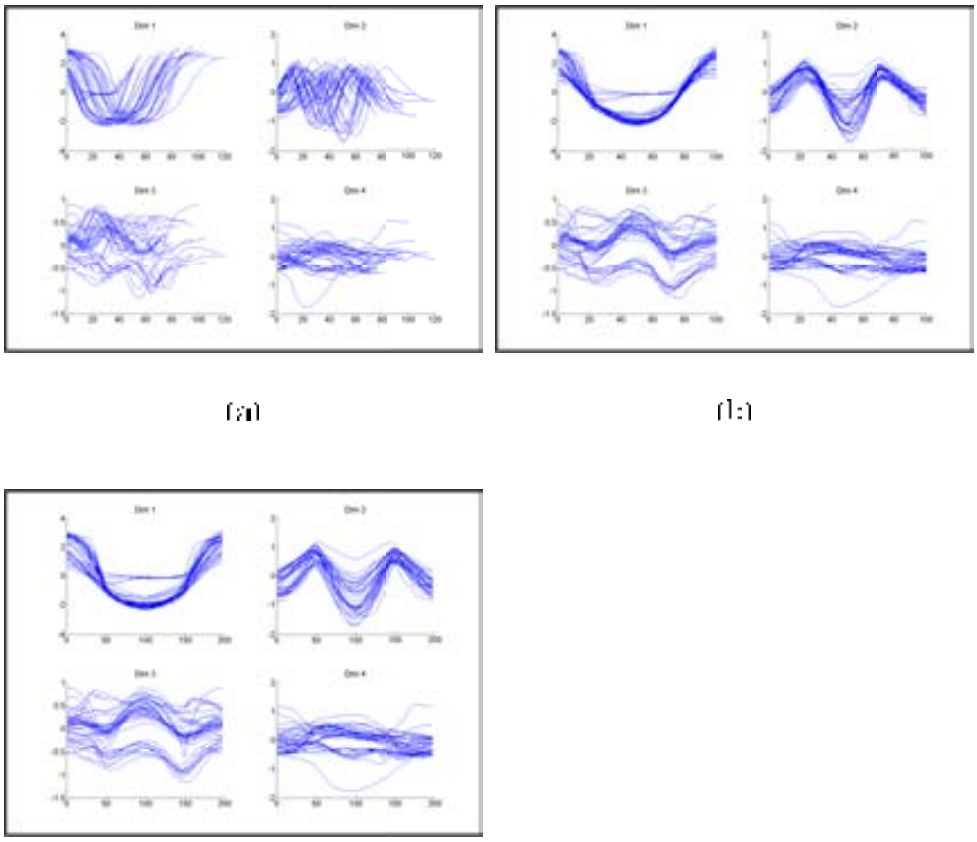
6.1 Synchronization of the training set

In the first place, we consider the training set for the bending action to illustrate the performance of the motion sequences synchronization method explained is section 4.3. The training set for this action consists of 51 performances carried out by 9 different actors as seen in Table 4.1.

We chose the first 16 eigenvectors that captured 95% of the original data to build the aSpace representation. The first 4 dimensions within the aSpace of the training sequences are illustrated in Fig. 6.1.(a). All the performances have different durations with 100 frames on average. The observed initial data shows different durations, speeds and accelerations between the sequences. Such a mistiming makes it very difficult to learn any common pattern from the data.

The proposed synchronization algorithm was coded in C++ and run with a 3 GHz Pentium D processor. The time needed for synchronizing two arbitrary sequences taken from our database is $1.5 \cdot 10^{-2}$ seconds and 0.6 seconds to synchronize the whole training set. The output from the synchronization stage is illustrated in Fig. 6.1.(b).

To prove the correctness of our approach, we manually synchronized the same training set by selecting a set of 5 key-frames in each sequence by hand following a maximum curvature subjective criterion. Then, the training set was resampled so each sequence had the same number of frames between each key-frame according to



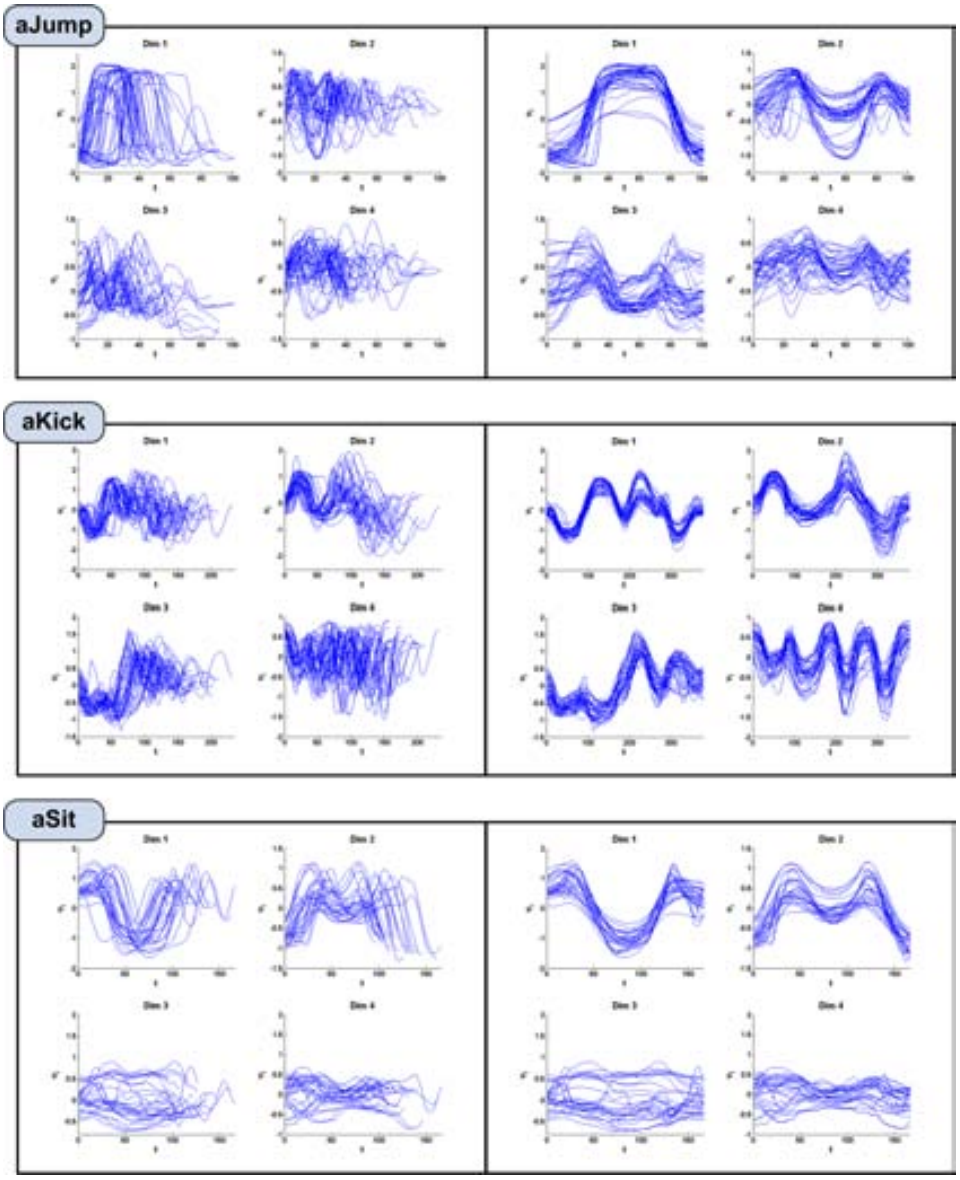
(a) (b)

(c)

Figure G.1: (a) Non-synchronized training set. (b) Automatically-synchronized training set with the proposed approach. (c) Manually-synchronized training set with key-frames.

Eq.(4.12). In Fig. C.1(c), the first 4 dimensions within the *aSpace* of the resulting manually synchronized sequences are shown. We might observe that the results are very similar to the ones obtained with the proposed automatic synchronization method.

Finally, on the second place, additional experimental results for other sets of actions are depicted in Figures C.2 and C.3. In particular, the actions shown are: the jumping, kicking, sitting, squatting and walking actions. For each action, we plot both (a) the non synchronized (left) and (b) the automatically synchronized training performances (right). Again, only the first 4 dimensions within the *aSpace* are shown.

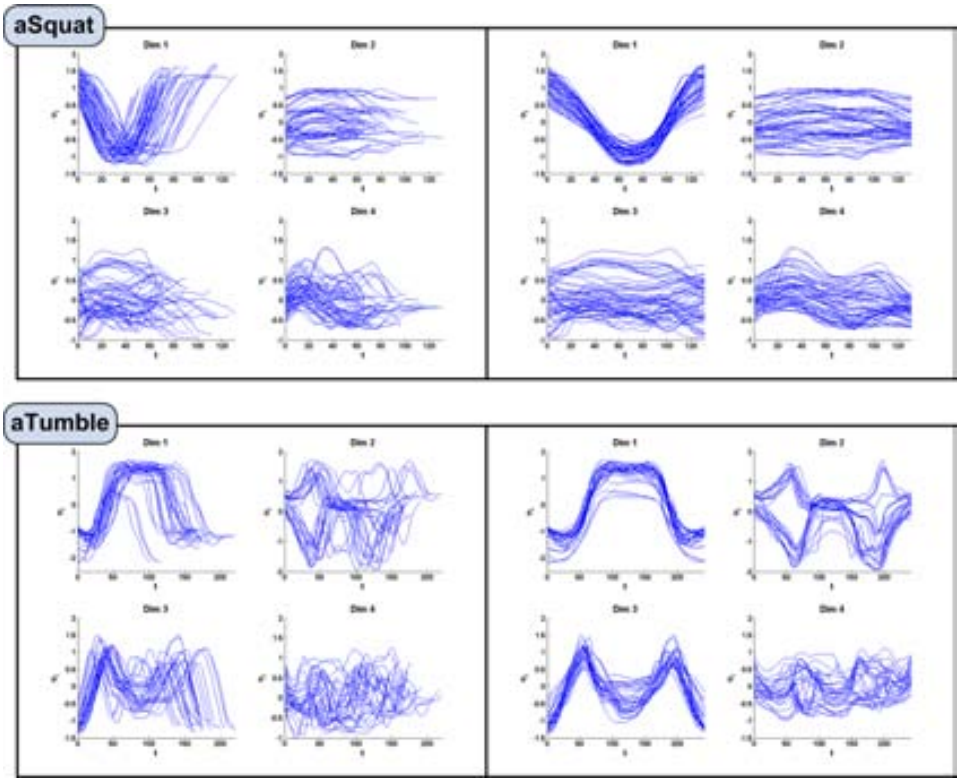


(a) (b)

Figure 6.2: Synchronization results for the jumping, kicking and sitting actions. (a) Non-synchronized training set. (b) Automatically-synchronized training set with the proposed approach.

6.2 Probabilistic matching of motion sequences

A second set of experiments were done for testing the behaviour of the probabilistic matching methodology introduced in section 5.3.1. Towards this end, we designed a



(a) (b)

Figure 6.3: Synchronization results for the squatting and tumbling actions. (a) Non-synchronized training set. (b) Automatically-synchronized training set with the proposed approach.

test scenario for gait recognition, i.e. recognize the id of a subject given a sequence of his body motion. To carry out this experiment, were considered the walking performances from the CMU dataset (see Table 4.2) given the great number of samples available and the diversity between subjects, i.e. 126 walking cycles performed by 12 different actors. The experiment setup, methodology and results is as follows.

The aim of this test is to identify which subject is performing an action by analyzing the observed motion from a particular test subject. Hence, instead of training a global action model considering all the motion performances available for a particular action, we trained an specific model per each subject Si, where i identifies the subject according to Table 4.2. As a result, we learned 11 different action models, denoted as Γ^{S2}, Γ^{S5}, Γ^{S7}, Γ^{S8}, Γ^{S12}, Γ^{S16}, Γ^{S35}, Γ^{S38}, Γ^{S39}, Γ^{S43}, and Γ^{S49}. All subject-dependent action models share the same PCA space representation Ω^{A_k} so all the postures are represented in a common space. Notice that subject $S55$ was not considered in this experiment since we had only 1 walking cycle available from this

	S2	**S5**	**S7**	**S8**	**S12**	**S16**	**S35**	**S38**	**S39**	**S43**	**S49**
S2	**66.7**	0	0	0	0	0	0	0	33.3	0	0
S5	0	**100**	0	0	0	0	0	0	0	0	0
S7	0	0	**80**	13.3	0	0	0	6.7	0	0	0
S8	0	0	0	**77.8**	0	0	0	0	22.2	0	0
S12	0	0	0	0	**100**	0	0	0	0	0	0
S16	0	0	0	0	0	**40**	33.4	13.3	0	13.3	0
S35	0	0	0	0	0	7.14	**92.86**	0	0	0	0
S38	0	0	0	0	0	25	0	**75**	0	0	0
S39	15.4	0	0	19.2	0	0	0	0	**65.4**	0	0
S43	0	0	0	0	0	0	0	0	0	**100**	0
S49	0	0	0	0	0	0	0	0	0	0	**100**

Table 6.1: Confusion Matrix in percentages for full cycle recognition

subject.

The approach is as follows: given an input motion sequence of length d, we compute the similarity S to all the subsequences of the same length from the 11 learned mean performances. Then, the subsequence which best matched a subject's mean performance according to our measure determines the identity of the subject.

Hence, the similarity measure used for gait identification between 2 subsequences of length d, namely $\Psi^a = \{\psi_a^1, ..., \psi_a^d\}$ and $\Psi^b = \{\psi_b^1, ..., \psi_b^d\}$ is given by Eq.(5.4). As a remainder:

$$S(\Psi^a, \Psi^b) = \exp\left(-D_M(\Psi^a, \Psi^b)\right) \left[\frac{(\overline{\mathbf{v}}_a \cdot \overline{\mathbf{v}}_b) + 1}{2}\right]^{\alpha},$$

where $\cdot$ stands for the dot product between vectors $\overline{\mathbf{v}}_a$ and $\overline{\mathbf{v}}_b$ corresponding to the average direction of motion computed following Eq.(4.27). D_M is the sum of the Mahalanobis distance within the PCA space Ω^{A_K} between each posture ψ_a^j and ψ_b^j from the subsequences, $j = 1..d$.

In our first experiment, we took a full walking cycle of each individual for testing the identification approach. We chose $b = 10$ dimensions for the PCA space representation of human postures. Subsequently, the similarity of the full test cycle to each specific action model's mean performance was computed according to Eq.(5.4). The tested walking cycle was removed from the training set in the learning stage. Then, this experiment was repeated for each cycle of the database, resulting in a total of 126 identification tests. The confusion matrix explaining the recognition performance can be seen in Table 6.1.

As we may observe, several miss classifications occur due to different reasons. On the one hand, results obtained for subjects $S2$, $S38$, $S43$ and $S49$ are not statistically confident since less than 5 cycles are provided in the training database. On the other hand, looking at the miss classification obtained between subjects $S16$ and $S35$ we discovered that indeed they correspond to the same actor who performed the recording. Despite of the fact that in the specification of the CMU database, these subjects are defined as different, the authors of this work recognized that the same person performed the recordings for both subjects datasets by subsequently checking

	S2	**S5**	**S7**	**S8**	**S12**
S2	**91.28**	0	0.05	0.47	0
S5	0	**97.21**	0	0	1.92
S7	0.35	1.80	**89.88**	0.12	0
S8	0.47	0	0.29	**91.86**	0
S12	0	0	0	0	**99.83**
S16	0	0	0	0	3.20
S35	0	1.34	0.06	0	4.19
S38	0	1.28	0	0	1.86
S39	6.51	0	0.06	3.49	0
S43	0	0	0	0	0
S49	0	0	0	0	0.17

	S16	**S35**	**S38**	**S39**	**S43**	**S49**
S2	0	0	0	7.04	1.16	0
S5	0	0	0.35	0	0	0.52
S7	0	0.06	2.50	0.12	1.92	3.25
S8	0	0	0.12	7.26	0	0
S12	0	0	0	0	0	0.17
S16	**64.17**	19.37	6.34	0	2.04	4.88
S35	19.09	**69.28**	3.84	0	1.10	1.10
S38	6.17	2.91	**76.85**	0	0	10.93
S39	0	0	0	**88.66**	1.28	0
S43	0	0	0	0	**100**	0
S49	0	0	0	0	0	**99.83**

Table 6.2: Confusion Matrix in percentages for subsequences of d=10 postures

the video recordings from those sessions.

A second experiment was run taking $d = 10$ as the length of the subsequences considered for performing gait identification. All the testing walking cycles have a total length of $F = 200$ postures, and therefore, only the 5% of a full walking cycle was used for gait recognition.

For each subject, a random test walking cycle was selected from the database. Thus, each tested cycle is composed of a total of $(F - d + 1)$ overlapping motion subsequences. Hence, we ran the gait identification experiment for each possible motion subsequence of each tested subject and computed its confusion matrix. The same experiment was repeated a total of 10 times.

The average of the obtained confusion matrices can be seen in Table 6.2. One can observe that the performance obtained is comparable with the full cycle experiment, but using only 1/20 of a walking cycle. Although some miss classifications occur between subjects that did not appear in the previous experiment, in some cases the performance is even better. This can be explained because of the better statistical robustness of this experiment, since we performed an identification test for each of the $(F - d + 1) = (200 - 10 + 1) = 191$ subsequences belonging to a full tested cycle. This results in a total of $191 * nSubjects * timesRepeated = 191 * 11 * 10 =$

21010 identification tests as opposed to the 126 identification tests from the previous experiment.
The results are very encouraging, since they show that we are able to recognize which subject is performing an action by observing only a very reduced motion portion from it.

6.3 Human body tracking results

In this section, we detail the experiments carried out to test the overall tracking approach. First, we comment on the training and testing sets used for this particular set of experiments, and define a suitable error measure for evaluating and comparing the results obtained. Then, we discuss on choosing an appropriate number of dimensions for the *aSpace* representation. Finally, we evaluate the performance of the tracking approach regarding the tracking efficiency improvement in terms of the number of needed particles, the computational cost, and the robustness against ambiguous and incomplete measurements from the detection stage, which are used to update the predicted postures. Sequences from three motion databases are used allover the experiments, namely the CVC and CMU databases and the HumanEva-I dataset[1].

In addition, qualitative tracking results are also presented for one walking sequence from the CAVIAR dataset[1] and video footage from two bending performances.

6.3.1 Training and testing sets

We have focused on the walking action for testing the overall tracking framework and illustrating the methodology of the approach within this section. This is motivated by the fact that walking constitutes one of the activities humans do more often and, in consequence, a very important action for HSE applications. In addition, we have a good amount of data available attaining this action from different databases. Nevertheless, the approach is easily extensible to any other actions by choosing a representative training set.

For the walking experiments carried out, the training set is composed by the selected walking performances from the CMU dataset as detailed in Table 4.2. Regarding the testing set, several walking cycles are selected from different motion databases.

In particular, the first testing sequence (testing sequence #1) consists in 4 continuous walking cycles from the same database used for training. Consequently, we randomly selected 4 continuous cycles from the CMU database, removed them from the training set, and used them for testing.

Alternatively, the second test sequence (testing sequence #2) consists of 2 and a half walking cycles from the HumanEva-I dataset [73]. This dataset comprises 4 subjects performing 6 different types of actions recorded in 7 calibrated video sequences from different viewpoints. Additionally, 2 trials were recorded per each action, and at least for one trial the video sequences are synchronized with the corresponding 3D pose parameters of the human body obtained by means of an optical motion capture

[1] available at `http://vision.cs.brown.edu/humaneva/`

system. We selected 2.5 cycles from the first trial of the walking action from subject 1 for testing our tracking approach. Notice that this database was acquired under different conditions and used different marker placements than the CMU database used for training. Hence, we aim to show that the learning process does not overfit to the training data by running similar tests on motion sequences from a totally different source than the one used for training.

Lastly, we used testing video sequences whose ground truth 3D pose parameters were not available. Hence, they are aimed to show qualitative results of the tracking approach. On the first hand, we used a testing sequence from the CAVIAR dataset [1] which corresponds to a walking video from a subject in a shopping mall. This dataset comprises several manually annotated video sequences of real subjects in an entrance lobby and a shopping center. On the other hand, some results for the bending action are also shown. The training set used corresponds to the CVC dataset and the testing set consists of 2 sequences of 2 different subjects performing a bending action whose 2D joints positions have been manually annotated.

6.3.2 Error measure

Given that in some of our experiments we are using motion captured performances to define the testing set, we can compare and quantify the error between the full 3D estimated body postures and the full 3D body postures from the ground truth data. To do so, it is needed to define some error measure between the tracker output and the ground truth data available.

Let's represent the pose of the body using $J = 15$ virtual markers, corresponding to the joint centers and limb ends of the human body model. Hence, a particular body configuration can be written as $\mathbf{X} = (x_1, ..., x_J)$, where $x_j \in \mathbb{R}^3$ is the 3D position of the marker j as in Eq.(5.8). We denote as $\mathbf{X}^e$ the 3D body posture estimated by the tracker, whereas the known body posture from ground truth is denoted by $\mathbf{X}^{GT}$. Thus, given a particular body posture in the *aSpace* representation, we first rewrite its body configuration as $\mathbf{X} = (x_1, ..., x_J)$ according to the procedure detailed in section 5.4. Then, the error between an estimated body posture $\mathbf{X}^e$ and the truly performed one $\mathbf{X}^{GT}$ from ground truth data is computed as the average squared distance between individual 3D joints, i.e.

$$D(\mathbf{X}^e, \mathbf{X}^{GT}) = \frac{1}{J} \sum_{j=1}^{J} ||x_j^e - x_j^{GT}||. \tag{6.1}$$

6.3.3 Determining the number of dimensions of the aSpace

In the following, the methodology to determine the appropriate number of b dimensions considered for building the *aSpace* representation is explained. Two main criteria are taken into account.

On the one hand, we compute the implicit *aSpace* representation error by projecting back the training sequences to the original representation space. Then, the error between the original postures and the reconstructed ones is computed according to Eq.(6.1). In Fig. 6.4.(a) we show a boxplot of the mean reconstruction error in mm.

computed for all the training sequences varying the b parameter. It can be seen, that the error is high for the first 4 dimensions, and it gets stabilized below 8 mm after $b = 5$ dimensions.

On the other hand, we empirically validate the most suitable value for b by running several complete tracking tests with fixed parameters but varying b. Towards this end, we selected 4 different walking cycles from the CMU database, and ran the particle filter tracker fixing all the parameters involved but b. In particular, we used $N = 500$ particles, $d = 10$, $\gamma = 80$, and $b_{filt} = 3$. The likelihood of each predicted particle was computed according to the mean 2D distance of all the projected joints between ground truth and estimated postures, both from a lateral viewpoint. Hence, we can evaluate the accuracy without the influence of artifacts caused by image-based likelihood measures. The tracker was initialized with ground truth data.

Then, several runs were carried out varying the b parameter from 3 to 12. Finally, for each value of b tested, we computed the mean estimated 3D joints error from the 4 testing cycles. The results are shown in Fig. 6.4.(b). We observed that while considering more dimensions for the *aWalk* representation actually lowers reconstruction error, the final estimation error gets higher as one keeps adding more dimensions after $b = 5$. This is due to the fact that the number of needed particles grows exponentially, up to a certain bound, w.r.t. the number of dimensions of the state space [50, 54].

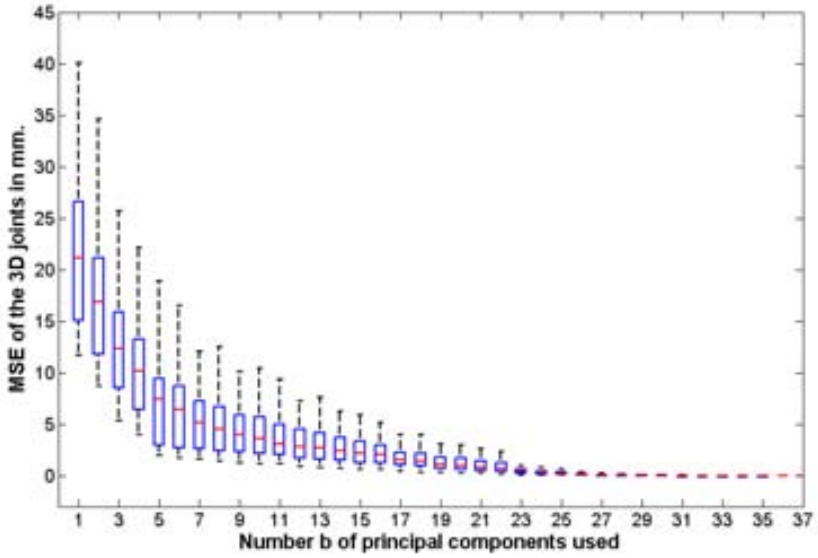
Therefore, by choosing $b = 5$ dimensions we explain the 94.55% of the variance present in the original training data with a mean reconstruction error of 7.68 mm. In conclusion, it results in a good trade off between the dimensionality reduction performed by PCA and accuracy of the estimation for the walking action with a manageable number of particles.

6.3.4 Tracking performance results

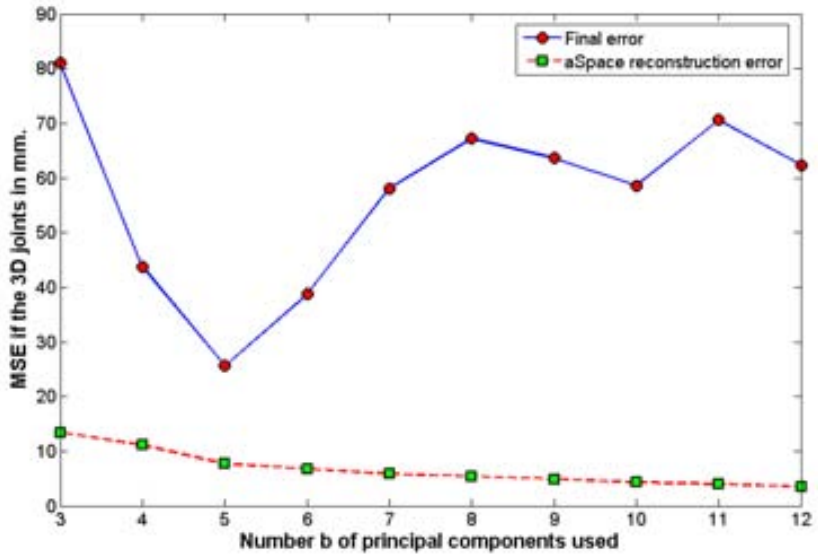
In order to test the performance of the proposed approach, we carried out several tests comparing the results obtained between three different tracking methods.

First, we used a standard Particle Filter tracker with a first order motion model. Second, our tracker using the prior model but without including the posture filtering step defined in section 5.3.3. Finally, our full tracking approach using the prior model and the posture filtering step. Hereafter the three tracking methods are referred to as: generic PF tracker, our tracker with the posture filtering step (PFT), and our tracker without the posture filtering step (NPFT), respectively. By posture filtering step, we refer to the rejection of predictions corresponding to non-likely postures according to Eq.(5.7). Notice that omitting this step is equivalent to setting the parameter $b_{filt} = 0$, since we are accepting all the predictions, and thus, the posture filtering step does not have any effect.

The first tests are intended to show the efficiency improvement of our tracker in terms of the computational time and the number of particles needed to achieve a certain error, for motions belonging to the type of action learnt. Thus, we compare the time consumption and the mean estimation error obtained by a standard PF with a very general motion prior against our motion model guided tracker with and without the posture filtering step, while varying the number of particles used.



(a)



(b)

Figure 6.4: (a) Boxplot of the reconstruction error of the training postures from the *aWalk* space varying b. (b) Total mean error obtained in a typical run of the tracker vs. reconstruction error, varying b.

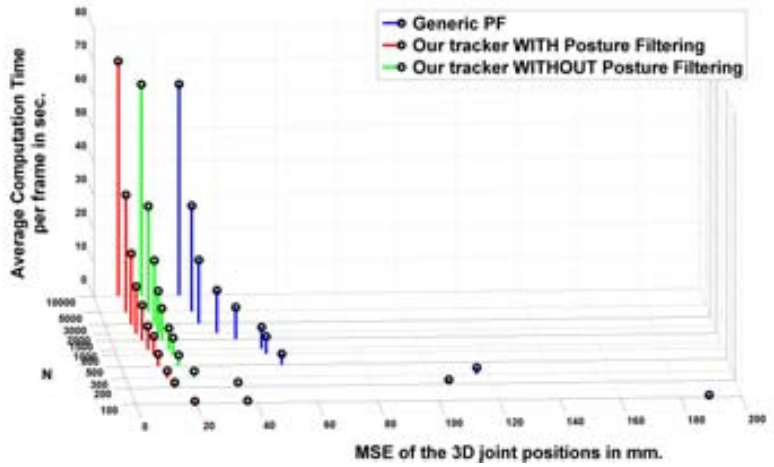
The PF's generic motion prior consists of a constant velocity model for the parameters, were each parameter is independent. The diffusion term consists of a fixed White Additive Gaussian Noise term (WAGN) with covariance computed in the same fashion than Eq.(4.29) for the dynamic's model expected error but considering all the postures from the training set. We used only 1 cycle from the CMU database for testing motivated by the fact that the generic PF tracker lost track very quickly, and after that point, the error obtained does not scale well to the number of particles used. All the parameters were fixed except the number of particles. Namely, we fixed $b = 5$, $d = 10$, $\gamma = 80$, and $b_{filt} = 3$ or $b_{filt} = 0$ for runs with or without the posture filtering step, respectively. The same likelihood computation method as the previous test was used.

Figure 6.5 relates the average computation time per frame, the mean estimation error, and the number of particles used in each filter run. All three trackers are implemented in MATLAB and running on a PC with an Intel(R) Pentium(R) 4 CPU @ 3.2 Ghz. with 2 Gbytes of RAM, and the code has not been optimized for high performance. On the one hand, the high error obtained by the generic PF for less than 500 particles ($> 53mm$), is explained by the fact that it totally lost track of the target after a few frames. Misstracks generally occurred after a large acceleration in the parameter space, since the generic motion prior assumes a constant velocity for each parameter. Hence, the ability of the generic PF to handle accelerations depends on the diffusion model and the number of particles considered. Thus, the larger the acceleration is, the larger the diffusion applied should be, demanding more particles to properly populate the state space.

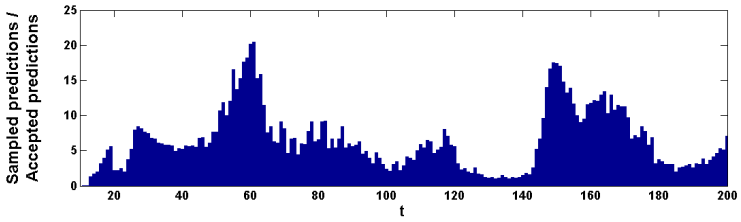
On the other hand, from 500 particles or more, the generic PF tracker could complete a full cycle without loosing track. However, the final error of the generic PF is much higher (21.32 mm. for the $N = 10000$ particles test) than any of the errors obtained by our tracker with posture filtering. Hence, even adding more particles to the generic PF the estimation error stabilizes and never achieves better results than the 100 particles test for the tracker with posture filtering (MSE of 19.53 mm.). This is due to the role of the posture filtering step within the tracking process. Hence, we observed that indeed, it discards modes in the likelihood function which would give high weights to particles corresponding to badly estimated human postures at a given time instant. Thus, without filtering these non-likely postures, these particles are considered for computing the final estimated state and consequently, the overall tracking performance decays.

Regarding time consumption, the generic PF and the tracker without the posture filtering step have very similar processing times. This is explained by the fact that the computation of the likelihood is usually the most time consuming part of a particle filtering framework. In comparison, the overhead introduced by the dynamic's model probabilistic matching is almost negligible in the experiments carried out.

Then, the tracker with posture filtering shows a slight increase w.r.t. the processing time at a given number of particles, but outperforms the other two regarding the final estimation error obtained. For instance, for the $N = 10000$ particles test, the average processing time per frame was 62.97, 63.02 and 70.08 seconds, with an MSE



(a)



(b)

Figure 6.5: (a) MSE of the estimated 3D joints' position and average processing time per frame, obtained varying the number N of particles, of a generic PF tracker (blue), our tracker with (red) and without (green) filtering non-feasible human postures according to our motion model. (b) Mean prediction acceptance ratio per frame of the posture filtering step.

of 28.32, 15.85 and 8.16 mm. for the generic PF, the tracker without posture filtering, and the tracker with posture filtering, respectively. Comparatively, for the $N = 100$ particles test, the results obtained were 0.63, 0.63 and 0.70 seconds with an MSE of 189.27, 36.93 and 19.53 mm.

On average, the overhead introduced by the posture filtering step takes 10.1% of the total processing time on the experiments carried out. This overhead is directly proportional to the prediction acceptance ratio, defined as the ratio between the number of predictions sampled from the dynamic model and the number of postures accepted. Fig. 6.5.(b) shows the mean prediction acceptance ratio per frame obtained. Hence, while the mean ratio is 6.26, there are two peaks of around 20 sampled predictions per accepted posture in the beginning and the end of the tested sequence. Interestingly, we observed that they occur in areas where the mean performance has high curva-

ture, since particles show some inertia which makes its adaptation to abrupt changes in the state of the tracked object more difficult. This inertia can be explained by the assumption of a first order motion model and the probabilistic sequence matching technique of the prediction step. However, specially in this situation, the posture filtering step shows bigger improvements in the particle set efficiency, since most of the predictions that wrongly follow the particle's inertia are rejected.

The second set of tests are intended to validate the ability of our approach to keep track of the target's state when reduced and ambiguous measurements from the scene are used to update the predictions. As discussed in section 5.4, we assume that there exists a detection stage which is applied once to each input image I_t whose output is a set of 2D image coordinates corresponding to the detected joint positions at time t.

Therefore, we use 2D ground truth data to test the robustness of our tracker against incomplete evidences in two different manners. From the one hand, we use ground truth information from all the joints available, and from the other hand, we consider only three joints, namely the head, one foot and one hand. This is motivated by the fact that there exist many approaches to detect some body limb positions in the images, and in addition, background subtraction algorithms can be used in combination in a static camera setup such as ours. Furthermore, this reduced set of joints is typically observable in images most of the times under normal circumstances. As a result, we assume that finding an approximate 2D position of the head, and at least one hand and one foot in input images is a feasible task for the detection stage under normal conditions.

The viewpoint used is also varied between a lateral and a totally frontal one. Consequently, we designed different tests with an increasing level of difficulty regarding these parameters for the likelihood computation. In every test the proposed tracker is also compared to a PF tracker with our motion prior, but without including the posture filtering step.

Finally, we fixed some parameters and varied the number of joints and the viewpoint considered the likelihood computation. Specifically, we used $N = 500$ particles, $\alpha = 30$ from Eq.(5.4), $\gamma = 80$, $b = 5$, $d = 10$, and $k = 3$ and $b_{filt} = 3$ was used for the posture filtering step from Eq.(5.7).

First, we projected the 3D postures to a 2D plane using a perspective projective model. We ran the tracker against testing sequences #1 and #2 for both viewpoints, and computed the likelihood of the predicted postures based on the 2D positions of all the body joints from the estimated postures vs. their ground truth. Additionally, every test was repeated 30 times in order to provide statistical significance to the results obtained. Fig. 6.6 shows the mean estimation errors per joint in mm. for both testing sequences and the standard deviation observed. Blue solid and red dashed lines encode whether we used the filtering step (PFT filter) to constrain the state space or not (NPFT filter). First of all, we may observe that for the frontal viewpoints a greater overall error is obtained compared to the lateral ones. This is explained by the fact that on a lateral viewpoint most of the motion is observable since the subject is walking parallel to the projection plane, regardless of self-occlusions. Contrarily, on a totally frontal view the main motion of arms and legs is lost, because there is no

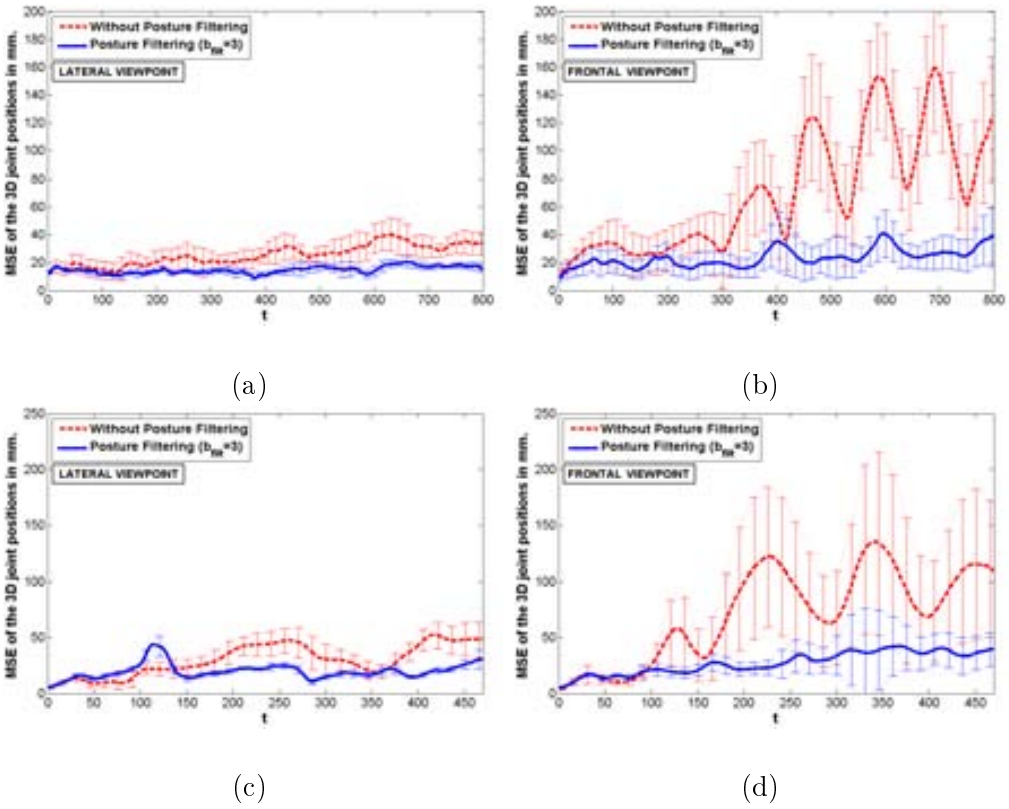


Figure 6.6: MSE of the estimated 3D joints' position for the PFT (solid blue line) and NPFT (dashed red line) filters, using 2D measurements from all the joints. (a) and (b): results for testing sequence #1. (c) and (d): idem for testing sequence #2.

depth information on the 2D projected walking postures. However, for the lateral view test in sequence #2 the tracking accuracy is slightly better in some frames without applying the posture filtering step. This is explained by the fact that sequence #2 was extracted from a different database than the one used for training. Therefore, the non-constrained tracker can track more general motions. However, the mean overall error for both sequences is still lower using the posture filtering step.

Then, for the lateral viewpoint, both filters (PFT and NPFT) can track the sequences with a stable behaviour. However, on the frontal viewpoint, the NPFT filter clearly fails after a few frames ($t \simeq 300$ and $t \simeq 100$ in sequences #1 and #2, respectively), while the PFT filter shows a low and stable error over time.

Regarding the standard deviation, for both viewpoints and sequences, the NPFT filter shows much bigger variability between different runs than the PFT filter. Also, frontal viewpoint tests have bigger standard deviations than lateral ones. This is due to the fact that frontal 2D postures are more ambiguous and difficult to track.

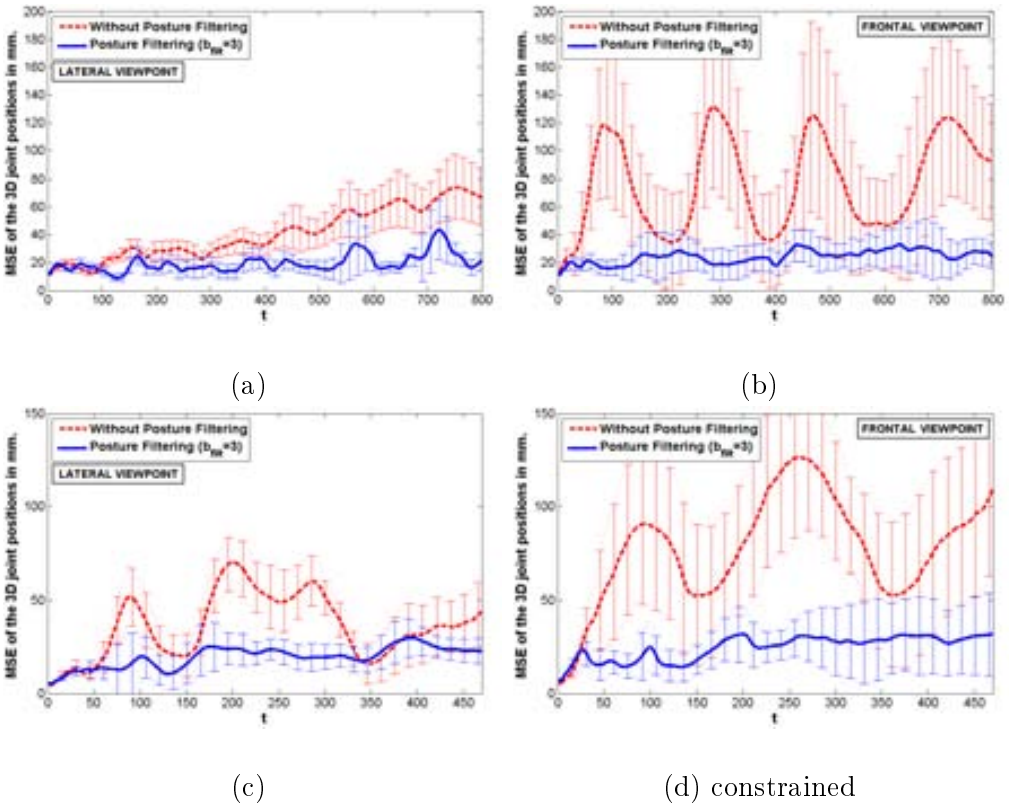


Figure 6.7: MSE of the estimated 3D joints' position for the PFT (solid blue line) and NPFT (dashed red line) filters, using 2D measurements from only 3 joints. (a) and (b): results for testing sequence #1. (c) and (d): idem for testing sequence #2.

Finally, we repeated the experiment but reducing the number of 2D joints considered. For the likelihood computation, we used the known 2D position of only 3 joints, namely the center of the head, the left hand and the left foot. In Fig. 6.7 we show the results obtained. As in the previous experiment, the frontal viewpoints show a greater overall error and standard deviation compared to the lateral ones for the same reasons mentioned above. From the one hand, the performance of the non-constrained tracker is very poor even for the lateral viewpoint, since it rapidly looses track of the target as supported by Figs. 6.7.(b), (c) and (d). However, although it has higher standard deviation than in the previous experiment, the PFT filter does perform well and is able to keep track of the target with relatively low and constrained error throughout all the walking cycles from both sequences. Therefore, although resulting in a less generic tracker, the use of the posture filtering step is responsible for stabilizing the tracking error even using a very reduced set of joints to update the predictions, as supported by the tests performed.

		Mean error		Max error		Min error	
		NPFT	PFT	NPFT	PFT	NPFT	PFT
All joints	**S1**	25.53	14.80	32.83	19.29	16.08	13.13
2D Lateral	**S2**	29.91	19.95	54.31	21.10	16.89	15.72
All joints	**S1**	69.57	23.48	118.22	48.98	32.15	12.60
2D Frontal	**S2**	71.34	27.57	149.36	51.38	29.37	17.37
3 joints	**S1**	40.87	19.70	55.30	39.55	17.03	10.14
2D Lateral	**S2**	36.42	19.68	64.59	36.60	19.22	10.75
3 joints	**S1**	77.08	24.42	169.43	41.21	35.17	12.87
2D Frontal	**S2**	78.00	24.82	165.19	51.51	38.02	14.65

		STD		Estimate STD	
		NPFT	PFT	NPFT	PFT
All joints	**S1**	4.29	1.08	3.48	0.81
2D Lateral	**S2**	8.00	1.81	3.80	1.42
All joints	**S1**	22.53	10.07	29.15	7.24
2D Frontal	**S2**	39.63	12.02	27.57	8.36
3 joints	**S1**	9.40	6.65	20.15	2.64
2D Lateral	**S2**	10.05	7.49	30.43	3.73
3 joints	**S1**	40.82	10.43	57.44	7.54
2D Frontal	**S2**	33.89	12.71	61.83	6.29

Table 6.3: Summary of the tracking error obtained from each experiment in mm.

Table 6.3 summarizes the experiments carried out regarding all the tests performed on the two sequences, namely the CMU and HumanEva-I chosen sequences, with and without filtering. The error measures have been obtained on a basis of 30 filter runs per sequence, due to the non-deterministic nature of particle filters. NPFT and PFT refers to the results for the Non-Posture-Filtered tracker ($b_{filt} = 0$), and the full tracking approach with Posture Filtering ($b_{filt} = 3$), respectively. Each row corresponds to a particular experiment. We present the mean, maximum and minimum of the mean estimation errors, in millimeters, obtained from each run for each testing sequence (S1 or S2), for each experiment. Additionally, the standard deviation observed in the mean error is also shown, as well as the standard deviation of the estimated postures across different runs to test the quality of the estimates. As previously discussed, the tracking error is considerably higher using frontal views as opposed to lateral views for the likelihood computation. Also, the tracker which does not include the posture filtering step presents an overall higher mean error, specially in the 2D frontal viewpoint tests, since it lost track of the target in most of the experiments carried out. Additionally, for the tracker with posture filtering, the standard deviation of the mean error is noticeably lower among the different tests, since the estimation error remains constrained and stable throughout all the frames of the test-

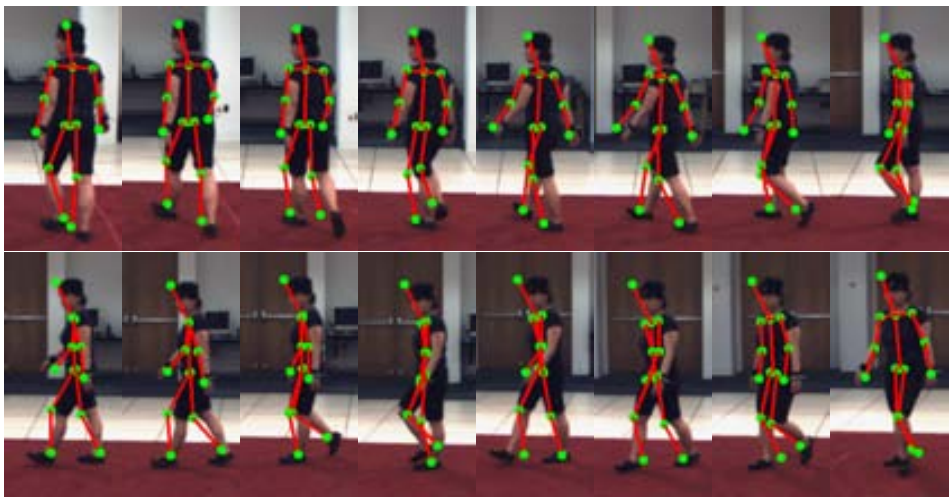


Figure 6.8: Tracking results: 16 frames of the HumanEva dataset test sequence with their corresponding estimated body postures overlapped, seen from camera #1.

ing sequences. Regarding the standard deviation of the estimates, we computed the distance of Eq.(6.1) between each 2 estimated postures of each pair of runs. The value shown corresponds to the mean of the standard deviation of these distances along a testing sequence, in order to summarize the estimate variance observed in each test. The results point out that the estimated postures for the 2D lateral experiment considering all the joints are very stable for both the NPFT and PFT trackers. However, on the other experiments, the NPFT tracker shows large variance compared to the PFT one. This is explained by the fact that the PFT tracker constrains the state space and the estimated postures do not differ too much between runs as long as a severe misstrack does not occur.

Then, Figs. 6.8 and 6.9 show the stick figures of the estimated 3D postures for the testing sequence #2. They have been overlapped with their corresponding image frames from the HumanEva-I dataset, from two different cameras. The 6 DOF corresponding to 3D position and orientation of the body were taken from ground truth data as well as the camera calibration parameters. The relative human body configuration corresponds to the estimated body postures by our tracker from the last experiment, i.e. using the known 2D positions of only 3 joints observed from a frontal viewpoint. For further details on the camera placement and acquisition conditions of the HumanEva dataset, we may refer the reader to [73].

Finally, we ran additional experiments in order to test qualitatively the output of our tracking approach under more realistic conditions. Towards this end, we used video footage from walking and bending sequences whose 3D ground truth data is not available. The first sequence was taken from the CAVIAR dataset [1], and is

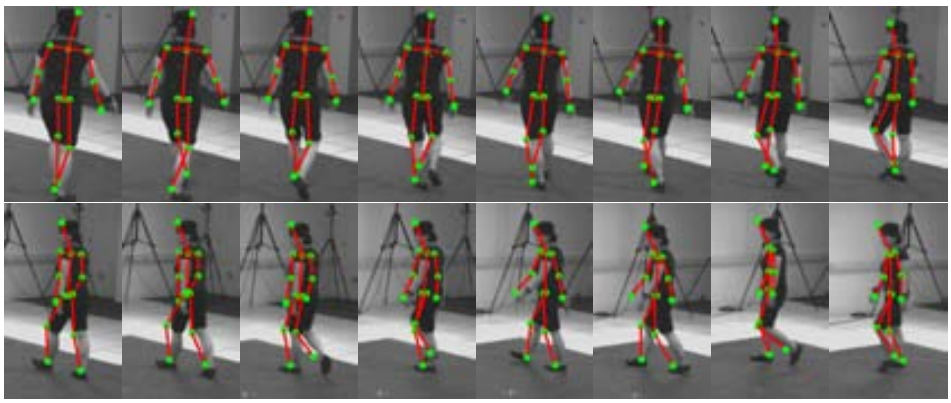


Figure 6.9: Tracking results: 16 frames of the HumanEva dataset test sequence with their corresponding estimated body postures overlapped, seen from camera #6.

Figure 6.10: Tracking results: 16 frames of the sequence TwoEnterShop1front from the CAVIAR dataset, where the corresponding estimated body postures have been overlapped.

named *TwoEnterShop1front,* which offers a lateral view of a woman (subject $S0$) walking in a straight line from left to right in a shopping center. This sequence was selected because the 2D position of feet, hands and head of the subject were partially annotated, i.e. only the body parts which are visible at each frame have been annotated, leading to annotations with occlusions and noise. Therefore, ground truth data for this experiment is closer to realistic conditions as opposed to the ground truth data obtained from a HMC system used in the previous experiments. Specifically, the 2D position of both feet and the head is well annotated and available for each frame of the sequence. On the contrary, the left hand has no annotations available at all, and the right hand suffers multiple short occlusions from frames $\#303 - \#390$, and a severe occlusion from frames $\#217 - \#250$ and $\#291 - \#302$.

As in previous experiments, we ran our tracker with the posture filtering step with $N = 500$ particles, in order to track the subject $S0$ from frames #184 to #394. The available 2D ground truth data was used to update the predictions at each time

step. Frame #184 was chosen as the initial frame because it is the first frame with a full-body view of the subject. The global position of the subject was manually annotated, as well as the first postures of our body model in order to initialize the tracker.

Figure 6.10 shows the results obtained on every $14th$ frame of the sequence. The stick figures representing the final estimated 3D body postures have been superimposed over their corresponding frames, and the available 2D ground truth position of the annotated limbs is depicted as a purple circle centered at their actual position. On the one hand, we observe that on frames where ground truth data about hands position is not available, the estimated arms 3D posture is given by the strong motion prior. Then, on frames with the right hand next to the subject's face, the tracker is unable to track its true position due to the restrictive motion prior used. However, on the last frames the quality of the arms tracking improves, since the hand's 2D ground truth position is more stable and the resulting posture is accepted as a feasible human posture by the action model. On the other hand, the approximate configuration of both legs is successfully estimated along all the walking cycles present, which shows the ability of the proposed approach to produce rough estimates of the performed motion even when a very reduced set of measurements are available.

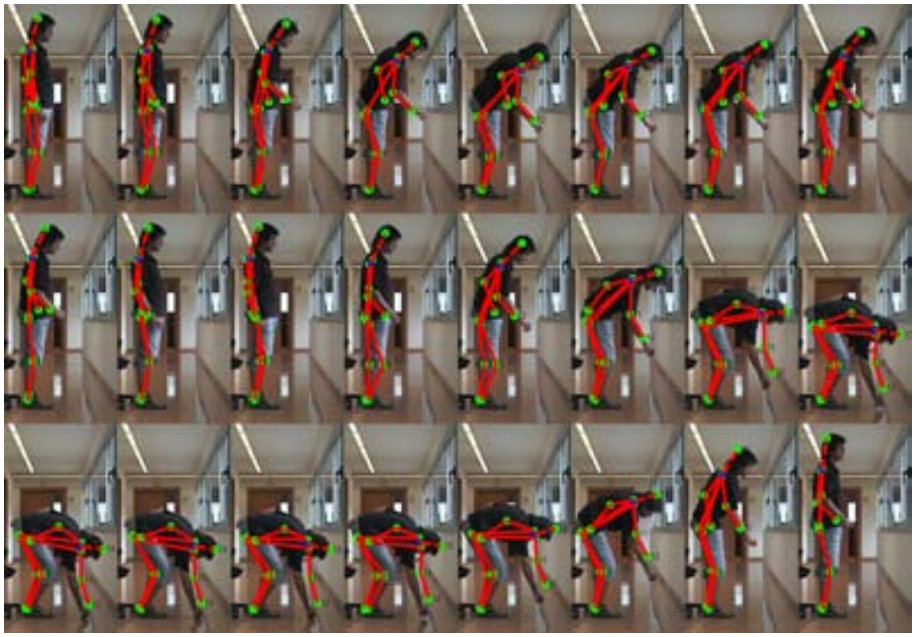
Lastly, two video sequences of a bending action from a lateral viewpoint were also tracked. The first sequence corresponds to 73 frames of a full bending performance and the second sequence has a total of 289 frames where a subject begins to bend, then he returns to a standing position and finally he completes a full bending execution. To carry out this test, we manually annotated the 2D position of the following body parts: the head, one hand, one foot, one knee and the hip. In addition, the 3D position and orientation of the full body was also initialized by hand, as well as the length of each body limb to fit each subject. The parameters used were $N = 1000$ particles, $\alpha = 30$, $\gamma = 80$, $b = 6$, $d = 10$, and $k = 3$ and $b_{filt} = 3$.

The tracking results are depicted in Fig. 6.11. As we might observe, for both sequences, the full 3D body posture is successfully tracked throughout the bending performance using only measurements from five known 2D joint positions. On the first hand, the sequence on Fig. 6.11.(a) is easier, because the performed action bears a lot of similarity with the prototypical bending action learnt from training data. Nevertheless, if we look carefully at the images provided, the limb representing the head and the shoulder complex are reconstructed slightly twisted as if the subject's head was turned to the left. This is due to the fact that our body model approximates the human body as a rigid articulated chain, and thus the limbs' length is fixed and cannot vary over time. However, during a bending performance, and specially when the subject is totally bent, the back appears slightly curved, and thus, its length varies resulting in twists and missalignments in the tracked postures to accommodate these phenomena. Also, as expected, the left arm and leg don't follow the truly performed motion because no information is provided to update their state. Hence, their motion is mostly guided by the motion prior resulting in plausible postures overall.

On the other hand, the second bending sequence illustrated in Fig. 6.11.(b) is more challenging. Here, the subject begins to bend but then, he returns to an upright position and completes a full bending performance from there. Notice, that the dataset used for training is composed only of bending sequences going downwards.

As a result, the motion prior guides the predicted postures downwards as the subject begins to bend, and despite the inertia of the particle set, the subject is successfully tracked as it returns to a standing position. Mainly, this is due to two reasons. From the one hand, the ability of the particle filter tracker to keep multiple hypotheses of the state of the tracked object, and from the other hand, the nature of the probabilistic matching step. Hence, at a given time step, some motion subsequences are recognized as a bending going downwards, while some other subsequences are matched as a bending going upwards, leading to multiple hypotheses about the state of the subject. Artifacts due to the fixed limbs length are also present, but overall, the subject is tracked successfully throughout the sequence.

(a)



(b)

Figure 6.11: Tracking results for the aBend action: (a) 16 frames from a short bending sequence and (b) 24 frames of a longer bending sequence where the subject starts bending, then stands up again, and finally completes a full bending performance.

Chapter 7

Concluding Remarks

The work presented in this Thesis is all part of the efforts for automated visual human motion analysis. Towards this end, we focused on the definition of strong priors for human motion to ease the problem of tracking the approximate parameters of a full body 3D model from a monocular image sequence. Hence, rather than aiming to recover the accurate configuration of a body model over time, the key idea is to exploit a series of learnt motion priors for a predefined set of actions, so the approximate body motion can be recovered from noisy and incomplete image measurements suitable for generating coarse approximations about how humans move and behave for scene understanding applications.

The overall problem has been faced as a model-based tracking approach formulated as a recursive Bayesian filter. In particular, the particle filtering framework has been chosen to tackle the problem of tracking a human body model over time because their ability to keep multiple hypotheses about the state of the object, and the ease to incorporate a priori knowledge on human motion within their prediction stage. However, most PF approaches for monocular full-body tracking suffer from issues related to PFs' discrete nature and must deal with the lack of robustness of image-based likelihood functions. Hence, our approach exploits learnt strong motion priors to improve the overall tracking performance, compared to a generic PF tracker, in terms of number of needed particles, computational cost, and robustness against ambiguous and incomplete measurements from images. These measurements are provided to the tracker by means of an external detection stage which is assumed to feed our PF tracker with the 2D positions of a variable set of body joints on the image plane. Then, the state of a full 3D human body model is inferred over time guided by our human action models.

For building our action-specific motion models, first, we defined a suitable yet simple human body model consisting of a stick figure body composed of 12 limbs structured in a hierarchical manner. Direction cosines have been chosen to represent limbs orientations thus avoiding singularities and abrupt changes in the representation space at the expense of extra parameters. As a result, our human body model is simple and compact, and the use of direction cosines enables useful statistical analysis over human body postures since a smooth representation is guaranteed, i.e. near

configurations of the body limbs account for near positions in the parameter space.

On the second place, Principal Component Analysis has been applied to the training data for an action to perform dimensionality reduction over the highly correlated input data, thus leading to a coarse-to-fine representation of the human body, i.e. the so called *aSpace*, which relates the precision of the model with its complexity by means of the main modes of gait variation within each action, i.e. the principal components found for each action.

The parameters of the action-specific model are learnt from examples of motion-captured data. In particular, they have been trained with pre-recorded motion sequences from our own action dataset, i.e. the CVC dataset, and from the publicly available Carnegie Mellon University's Graphics Lab Motion capture database. Both datasets' motion sequences present different speeds and accelerations. Therefore, we developed a dense matching algorithm based on DP, which has been used to synchronize human motion sequences of the same action class. The algorithm finds an optimal solution in real-time. Additionally, we automatically select from the training data the best pattern for time synchronization following a minimum global distance criterion.

Then, the synchronized version of the training set has been subsequently used overall the whole learning approach. Hence, a probabilistic action model is learnt based on these examples which captures the variability and temporal evolution of full-body motion within a particular action. The motion model allows to predict feasible 3D body postures given a small motion history of a particular action, as well as determine which configurations of the representation space correspond to feasible human body postures given a particular action. In particular, the parameters of the resulting action model consists of: a representative manifold for the action, namely the mean performance, the standard deviation from the mean performance, the mean observed direction vectors from each motion subsequence of a given length and the expected error at a given time instant.

Subsequently, the action model has been embedded into a motion prior which is used as a priori knowledge within the particle filter tracking framework. From the one hand, we introduced a dynamic model responsible of predicting new body postures given the previously estimated ones, which has proven to drastically improve the efficiency of the PF tracker compared to a constant velocity model for the body model's parameters. On the other hand, a posture filtering step has been added to discard predictions which correspond to non-feasible body postures. If the motion performed belongs to the trained action class, this adaptive constraint of the search space improves the overall tracking reliability, stabilizes the overall error and avoids misstracks due to ambiguous and incomplete measurements from the real world, as supported by the tests performed. To define a suitable likelihood function, a detection stage, which has not been considered in this work, has been assumed to provide approximate 2D position of some body joints on each image. The predictions given by the motion prior are then updated according to these 2D measurements.

The performance of the overall approach has been measured using testing sequences from two databases with ground truth data available, namely, the same one used for training (the CMU MoCap dataset) and the HumanEva-I dataset. The results showed that the approach generalized well within the tested action class, and that it

is robust against incomplete and noisy measurements. Towards this end, several tests have been carried out varying the number of joints considered and the viewpoint used to update the predicted postures from the dynamic model. In the worst tested case, results point out that our tracker is able to estimate the 3D configuration of a full-body model providing only the known 2D positions of 3 joints from a totally frontal viewpoint as measurements to compute the likelihood of the predicted postures.

In addition, further tracking experiments were also run to test qualitatively the output of the proposed approach under more realistic conditions. Hence, video footage from the CAVIAR project and two bending sequences were successfully tracked from the known 2D positions of a reduced set of joints.

Alternatively, we also carried out tests regarding the performance of the sequence synchronization algorithm. Hence, we manually synchronized several action training sets using key-frames and compared them against the results obtained automatically with our method. Tests showed that both the manually synchronized training set and the one obtained with the proposed automatic synchronization method were very similar. In addition, the algorithm computed the optimal solution in near real-time due to the Dynamic Programing formulation of the problem.

Lastly, a gait recognition test scenario was set out defining a set of experiments to test the behaviour of the probabilistic matching methodology of motion subsequences. The experimental results pointed out that we were able to recognize which subject performed a motion subsequence from a set of 11 tested subjects using a very reduced number of motion samples.

On the contrary, the approach has the following limitations. First, general non-constrained motion cannot be tracked, since the use of a strong motion prior limits the approach to the subset of actions learnt beforehand. Second, the body's absolute position and orientation estimation hasn't been addressed in this work. Then, the PCA-based posture representation has proven to be effective in reducing the dimensionality of the state space for highly correlated motion such as the action datasets used, i.e. the motion between arms and legs while walking and running, or the symmetry between the left and right part of the body in most of the tested actions. However, more complex motions with less linear correlation, like break dancing, wouldn't benefit from this amount of reduction. Finally, although the probabilistic matching step provides some flexibility, it is assumed that the framerate of both training and testing sequences do not differ too much.

Therefore, future research lines include the estimation of the absolute body orientation from the image sequence, and dealing with the initialization of the tracker for practical applications [58]. In addition, our tracker must be combined with approaches to track or identify the 2D body parts, i.e. the head, hands and feet [65, 85, 51, 74, 46, 47, 29] within each image implementing a proper detection stage. Towards this end, an interesting solution for this problem consists of learning a mapping between body silhouettes obtained by background subtraction techniques [39] and the viewpoint at a given height of the camera w.r.t. the subject [65].

Finally, the overall tracking approach should be trained and tested for a wider set of actions, and add multiple action support by selecting appropriate training sets and dealing with transitions between actions. Towards this end, it is worth noting that our method for matching motion subsequences probabilistically is specially well-

suited for this purpose. Hence, predictions taking into account multiple actions could be implicitly considered by only using a common representation *aSpace* for multiple action models. Thus, similar postures or subsequences between actions would lead to stochastically propagating these postures according to the most relevant action models.

At present, in the context of free-viewpoint media content creation, work is being done to apply the motion tracking framework to soccer scenes in combination with a commercial player tracking system which provides the bounding box of each player in the field accurately.

Appendix A

Acronyms

Given the presence of numerous acronyms through the text we have found convenient to summarize them in Tables A.1, A.2.

Symbol	Description
KF	Kalman Filter
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
IEKF	Iterated Extended Kalman Filter
fps	frames per second
PF	Particle Filter
pdf	Probabilistic Density Function
HCI	Human-Computer Interaction
HSE	Human-Sequence Evaluation
MSE	Mean Squared Error
PFT	Posture Filtered Tracker
NPFT	Non Posture Filtered Tracker
MoCap	Motion Capture
DOF	Degrees Of Freedom
PCA	Principal Component Analysis
MPCA	Multivariate Principal Component Analysis
CMU	Carnegie Mellon University
DP	Dynamic Programming
DSI	Disparity Space Image
MDL	Minimum Description Length

Table A.1: Acronyms (I).

Symbol	Description
GPDM	Gaussian Process Dynamical Model
MAP	Maximum A Posteriori
WAGN	White Additive Gaussian Noise
NN	Nearest Neighbor
MoG	Mixture of Gaussians
ROI	Region Of Interest
STD	Standard Deviation
HMC	Human Motion Capture

Table A.2: Acronyms (II).

Appendix B

Symbol List

In order to aid the reader comprehension, the symbols used throughout this work are here summed up, split in Table B.1 and Table B.2.

Symb.	Description	Symb.	Description
$\mathbf{R}$	rotation matrix	$Xy'z$"	Euler angles convention
(i, j, k)	quaternion's imaginary parts	θ_l^x	direction vector component for limb l
(x, y, z)	3D Cartesian coordinate	X_t	body posture's positional data
M	number of markers	$\mathbf{A}_k$	action k
Ψ_i	action performance	ψ_i^j	posture of a perf.
F_i	number of postures of a perf.	P	number of training perfs.
N_{A_k}	number of training postures for an action	$\mathbf{u}_b$	eigenvector
λ_b	eigenvalue	b	number of selected eigenvectors
$\bar{\psi}$	mean body posture	$\tilde{\psi}$	projected posture within the *aSpace*
$\tilde{\mathbf{\Psi}}$	projected perf. within the *aSpace*	$\mathbf{x}_i(t)$	interpolated expansion of a perf.
$\delta(\bullet)$	Dirac's delta	T	period
ρ	synchronization rate	$\tau(t)$	distance-time function
$\Delta_{n.m}(t)$	sought function synch.	$D_{n,m}$	synch. distance
$E(d, p)$	DSI cost value	$\mathbf{x}_{n,m}(t)$	synchronized version of $\mathbf{x}_m(t)$ to $\mathbf{x}_n(t)$
$\hat{\psi}_i^j$	posture of a synch. perf.	$\hat{\Psi}_i$	synchronized perf.

Table B.1: List of symbols used within this work.

Symb.	Description	Symb.	Description
$\bar{\Psi}$	mean performance	σ_t	STD from the mean perf.
$\overline{\mathbf{v}}_t$	mean motion direction	Σ_t	error covariance matrix
e_i^t	prediction error	Γ_{A_k}	action model
d	size of a motion subsequence	$\mathbb{E}(\bullet)$	expectation of a distribution
$\breve{\psi}_i^{t+1}$	predicted posture	$\mathbf{I}_t$	sequence of images
I_t	image	$\hat{\phi}_t$	estimated state
Φ_t^n	estimated motion history	$p(\phi_t\|\mathbf{I}_t)$	*posterior* pdf
$p(\phi_t\|\phi_{t-1})$	temporal prior	$p(I_t\|\phi_t)$	likelihood
N	number of particles	ϕ_t^s	particle
π_t^s	particle's weight	$\overline{\pi}_t^s$	particle's normalized weight
b_{filt}	number of filtering dimensions	β	survival rate
$\mathcal{D}_{min}$	min. acceptable num. of particles	$\eta(\bullet)$	zero-mean Gaussian noise
$S(\bar{\Psi}, \Phi)$	similarity between subsequences	$p(\bar{\Psi}_i\|\Phi_t^n)$	probabilistic match
α	prob. match balancing exp.	D_M	sum of Mahalanobis distances between postures
γ	likelihood's exponent	$m_{j,d}$	mapping function
$D(\mathbf{X}^e, \mathbf{X}^g)$	MSE between body postures		

Table B.2: List of symbols used within this work.

Appendix C

Publications

C.1 Journals

1. Action-specific motion prior for efficient bayesian 3D human body tracking. Ignasi Rius, Jordi Gonzàlez, Javier Varona, F. Xavier Roca. In Pattern Recognition, volume 42, number 11, pp. 2907-2921, November, 2009.

2. Automatic Learning of 3D Pose Variability in Walking Performances for Gait Analysis. Ignasi Rius, Jordi Gonzàlez, Mikhail Mozerov, F. Xavier Roca. In International Journal for Computational Vision and Biomechanics, volume 1, number 1, pp. 33-43, January-June, 2008.

3. Nonlinear Synchronization for Automatic Learning of 3D Pose Variability in Human Motion Sequences. Mikhail Mozerov, Ignasi Rius, Xavier Roca, Jordi Gonzàlez. In EURASIP Journal on Advances in Signal Processing, volume 2010, article ID 507247, January, 2010.

4. Importance of detection for video surveillance applications. Javier Varona, Jordi Gonzàlez, Ignasi Rius, Juan J. Villnueva. In Optical Engineering, volume 47, issue 8, 087201, August, 2008.

C.2 Conferences

1. Virtual camera synthesis for soccer game replays. N.Papadakis, A.Baeza, Ignasi Rius, X.Armangué, A.Bugeau, O.D'Hondt, P.Gargallo, V.Caselles and S.Sagas. Submitted to 7th European Conference on Visual Media Production (CVMP'2010), London, UK, November, 2010.

2. Exploiting Spatio-temporal Constraints for Robust 2D Pose Tracking. Grégory Rogez, Ignasi Rius, Jesús Martínez del Rincón, Carlos Orrite. In Second Workshop Human Motion - Understanding, Modeling, Capture and Animation, pages 58-73, Rio de Janeiro, Brazil, October, 2007.

3. Hierarchical Eyelid and Face Tracking. Javier Orozco, Jordi Gonzàlez, Ignasi Rius, F.Xavier Roca. In 3rd Iberian Conference on Pattern Recognition and Image Analysis (ibPRIA'2007), Girona, Spain, June, 2007.
4. 3D Human Motion Sequences Synchronization using Dense Matching Algorithm. Mikhail Mozerov, Ignasi Rius, Xavier Roca, Jordi Gonzàlez. In 28th Symposium of the German Association for Pattern Recognition (DAGM'2006), Berlin, Germany, September, 2006.
5. Action Spaces for Efficient Bayesian Tracking of Human Motion. Ignasi Rius, Javier Varona, Jordi Gonzàlez, Juan José Villanueva. In 18th International Conference on Pattern Recognition (ICPR'2006), Hong Kong, China, August, 2006.
6. Posture Constraints for Bayesian Human Motion Tracking. Ignasi Rius, Javier Varona, F. Xavier Roca, Jordi Gonzàlez. In 4th International Workshop on Articulated Motion and Deformable Objects (AMDO-e'2006), Andratx, Mallorca, Spain, July, 2006.
7. Robust Particle Filtering for Object Tracking. Daniel Rowe, Ignasi Rius, Jordi Gonzàlez, Juan J. Villanueva. In 13th International Conference on Image Analysis and Processing (ICIAP'2005), Cagliary, Italy, September, 2005.
8. Improving Tracking by Handling Occlusions. Daniel Rowe, Ignasi Rius, Jordi Gonzàlez, Juan J. Villanueva. In 3rd International Conference on Advances in Pattern Recognition (ICAPR'2005), Bath, United Kingdom, August, 2005.
9. 3D Action Modeling and Reconstruction for 2D Human Body Tracking. Ignasi Rius, Daniel Rowe, Jordi Gonzàlez, F.Xavier Roca. In 3rd International Conference on Advances in Pattern Recognition (ICAPR'2005), Bath, United Kingdom, August, 2005.
10. Probabilistic Image-based Tracking: Improving Particle Filtering. Daniel Rowe, Ignasi Rius, Jordi Gonzàlez, F.Xavier Roca, Juan J. Villanueva. In 2nd Iberian Conference on Pattern Recognition and Image Analysis (ibPRIA'2005), Estoril, Portugal, June, 2005.
11. A 3D Dynamic Model of Human Actions for Probabilistic Image Tracking. Ignasi Rius, Daniel Rowe, Jordi Gonzàlez, F.Xavier Roca. In 2nd Iberian Conference on Pattern Recognition and Image Analysis (ibPRIA'2005), Estoril, Portugal, June, 2005.

C.3 Technical Reports

1. Articulated 3D Human Motion Modeling for Tracking and Reconstruction. Ignasi Rius. In CVC Technical Report #91, CVC (UAB) , September, 2005.

Bibliography

[1] EC Funded CAVIAR project. http://homepages.inf.ed.ac.uk/rbf/caviar/.

[2] A. Agarwal and B. Triggs. Recovering 3 D human pose from monocular images. *PAMI*, 28(1):44–58, 2006.

[3] JK Aggarwal and Q. Cai. Human motion analysis: A review. *CVIU*, 73(3):428–440, 1999.

[4] M. Agrawal, K. Konolige, and M. Blas. Censure: Center surround extremas for realtime feature detection and matching. In *ECCV '08*, pages 102–115, Marseille, France, 2008.

[5] 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, February 2002.

[6] A.O. Balan and M.J. Black. An adaptive appearance model approach for model-based articulated object tracking. In *CVPR '06*, pages 758–765, Washington, USA, 2006.

[7] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and data association.* Academic Press, 1988.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[9] M. Brand. Shadow puppetry. In *ICCV '99*, pages 1237–1244, Corfu, Greece, September 1999.

[10] M. Braun. *Picturing time, work of Etienne-Jules Marey 1830-1904.* University of Chicago Press, 1995.

[11] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997.

[12] 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, May 2004.

[13] M.Z. Brown, D. Burschka, and G.D. Hager. Advances in computational stereo. *PAMI*, 25(8):993–1008, 2003.

[14] J. Chai and J.K. Hodgins. Performance animation from low-dimensional control signals. *SIGGRAPH 2005*, 24(3):686–696, 2005.

[15] J.C. Cheng and J.M.F. Moura. Capture and representation of human walking in live video sequence. *IEEE Trans. Multimedia*, 1(2):144–156, 1999.

[16] Q. Delamarre and O. Faugeras. 3D Articulated Models and Multi-View Tracking with Silhouettes. In *Proceedings of the International Conference on Computer Vision*, volume 2, page 716, 1999.

[17] Q. Delamarre and O. Faugeras. 3D Articulated Models and Multiview Tracking with Physical Forces. *Computer Vision and Image Understanding*, 81(3):328–357, 2001.

[18] J. Deutscher and I. Reid. Articulated Body Motion Capture by Stochastic Search. *IJCV*, 61(2):185–205, 2005.

[19] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[20] T. Drummond and R. Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *IEEE International Conference on Computer Vision*, Vancouver, 2001.

[21] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, volume 2, Madison, WI, USA, 2003.

[22] D. Fox. Adapting the Sample Size in Particle Filters Through KLD-Sampling. *International Journal of Robotics Research*, 22(12):985–1003, 2003.

[23] D.M. Gavrila. The visual analysis of human movement: A survey. *CVIU*, 73(1):82–98, 1999.

[24] D.M. Gavrila and L. Davis. 3d model-based tracking of humans in action: A multi-view approach. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 73–80, San Francisco, CA, USA, 1996.

[25] D.M. Gavrila and L.S. Davis. 3-D model-based tracking of humans in action: a multi-view approach. *Proc. CVPR '96*, pages 73–80, 1996.

[26] J. Gonzàlez, J. Varona, F.X. Roca, and J.J. Villanueva. A comparison framework for walking performances using aSpaces. *ELCVIA*, 5(3):105–116, 2005.

[27] Jordi Gonzàlez. *Human Sequence Evaluation: the Key-frame Approach*. PhD thesis, Universitat Autònoma de Barcelona, May 2004.

[28] K. Grochow, S.L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. *ACM Transactions on Graphics (TOG)*, 23(3):522–531, 2004.

[29] M. Al Haj, A. Amato, F.X. Roca, and J. Gonzàlez. Face Detection in Color Images using Primitive Shape Features. In *CORES'07*, Wroclaw, Poland, 2007.

[30] B. Han, Y. Zhu, D. Comaniciu, and L. Davis. Kernel-based bayesian filtering for object tracking. In *CVPR '05*, pages 227–234, San Diego, USA, 2005.

[31] C. Harris. Tracking with rigid models. *Active vision*, pages 59–73, 1993.

[32] D. Hogg. Model-based vision: a program to see a walking person. *Image Vision Computing*, 1:5–20, 1983.

[33] N.R. Howe, M.E. Leventon, and W.T. Freeman. Bayesian reconstruction of 3D human motion from single-camera video. In *Advances in Neural Information Processing Systems*, pages 820–826, 2000.

[34] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[35] A.D. Jepson, D.J. Fleet, and T.F. El Maraghi. Robust online appearance models for visual tracking. *PAMI*, 25(10):1296–1311, October 2003.

[36] S.J. Julier and J.K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Proceedings of SPIE*, volume 3068, page 182, 1997.

[37] M.B. Kaâniche and F. Brémond. Tracking HOG Descriptors for Gesture Recognition. *2009 Advanced Video and Signal Based Surveillance*, pages 140–145, 2009.

[38] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(D):35–45, 1960.

[39] K. Kim, TH. Chalidabhongse, F. Harwood, and L. Davis. Background modeling and subtraction by codebook construction. In *ICIP'04*, Singapore, 2004.

[40] O. King and D.A. Forsyth. How does CONDENSATION behave with a finite number of samples? In *ECCV'00*, pages 695–709, Ireland, 2000.

[41] F. Korc and V. Hlavác. Detection and tracking of humans in single view sequences using 2D articulated model. *Computational Imaging and Vision*, 36:105, 2008.

[42] G. Kurtenbach and E.A. Hulteen. Gestures in human-computer communication. *The art of human-computer interface design*, pages 309–317, 1990.

[43] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[44] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, New York, NY, 2006.

[45] H.J. Lee and Z. Chen. Determination of 3d human body posture from a single view. *Computer Vision Graphics*, 30:148–168, 1985.

[46] M.W. Lee and I. Cohen. Human Upper Body Pose Estimation in Static Images. In *ECCV'04*, Prague, Czech Republic, 2004.

[47] M.W. Lee and R. Nevatia. Body Part Detection for Human Pose Estimation and Tracking. In *WMVC'07*, Austin, Texas, USA, 2007.

[48] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 17–32, 2004.

[49] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[50] J. MacCormick and M. Isard. Partitioned sampling, articulated objects and interface-quality hand tracking. In *ECCV'00*, pages 3–19, Dublin, 2000.

[51] A. Micilotta and R. Bowden. View-based location and tracking of body parts for visual interaction. In *BMVC'04*, volume 2, London, UK, 2004.

[52] I. Mikić, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53(3):199–223, 2003.

[53] T.B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2-3):90–126, 2006.

[54] P. Del Moral and L. Miclo. Branching and interacting particle systems. Approximations of Feynman-Kac formulae with applications to non-linear filtering. *Séminaire de probabilités de Strasbourg*, 34:1–145, 2000.

[55] Eadweard Muybridge. *The Human Figure in Motion*. Dover Publications, Inc., 1955, First published 1887.

[56] A. Nakazawa, S. Nakaoka, and K. Ikeuchi. Matching and blending human motions temporal scalable dynamic programming. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 1, 2004.

[57] H. Ning, T. Tan, L. Wang, and W. Hu. Kinematics-based tracking of human walking in monocular video sequences. *IVC*, 22:429–441, 2004.

[58] R. Okada and B. Stenger. A single camera motion capture system for human-computer interaction. *IEICE Transactions on Information and Systems*, 91(7):1855–1862, 2008.

[59] E. Ong and S. Gong. A dynamic human model using hybrid 2d-3d representation in hierarchical pca space. In *10th British Machine Vision Conference*, United Kingdom, 1999.

[60] E.J. Ong, A.S. Micilotta, R. Bowden, and A. Hilton. Viewpoint invariant exemplar-based 3D human tracking. *CVIU'06*, 104(2-3):178–189, 2006.

[61] J. O'Rourke and N.I. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(6):522–536, June 1980.

[62] F.J. Perales, A. Igelmo, J.M Buades, P. Negre, and G. Bernat. Human motion analysis & synthesis using computer vision and graphics techniques. some applications., Benicassim, Spain, 16-18 May 2001.

[63] R. Plankers and P. Fua. Articulated soft objects for video-based body modeling. In *Proceedings of the Ninth International Conference on Computer Vision*, Vancouver, Canada, 2001.

[64] R. Ramamoorthi and A. H. Barr. Fast construction of accurate quaternion splines. In *24th Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*, pages 287–292, Los Angeles, California, USA, 1997.

[65] G. Rogez, C. Orrite, J., and J. Herrero. Probabilistic spatio-temporal 2d-model for pedestrian motion analysis in monocular sequences. In *AMDO'06*, pages 175–184, Andratx, Spain, July 2006.

[66] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59:94–115, 1994.

[67] A. D. Sappa, N. Aifanti, S. Malassiotis, and M. G. Strintzis. 3d human walking modeling. In *3rd International Workshop on Articulated Motion and Deformable Objects (AMDO'2004)*, pages 111–122, Palma de Mallorca, Spain, September 2004.

[68] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002.

[69] J.M. Selig. *Geometrical methods in robotics*. Springer, New York, 1996.

[70] H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV'00*, pages 702–718, Dublin, 2000.

[71] H. Sidenbladh, M.J. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *ECCV'02*, volume 1, pages 784–800, Copenhagen, Denmark, 2002.

[72] L. Sigal and M. Black. Predicting 3D People from 2D Pictures. In *AMDO'06*, pages 185–195, Mallorca, Spain, 2006.

[73] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical report CS-06-08, Brown University, 2006.

[74] L. Sigal and M.J. Black. Measure Locally, Reason Globally: Occlusion-sensitive Articulated Pose Estimation. In *CVPR'06*, volume 2, pages 2041–2048, New York, USA, 2006.

[75] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *CVPR'01*, Kauai Marriott, Hawaii, 2001.

[76] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *The International Journal of Robotics Research*, 22(6):371, 2003.

[77] Bjorn Stenger, Arasanathan Thayananthan, Philip H. S. Torr, and Roberto Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *PAMI*, 28(9):1372–1384, 2006.

[78] A. Thayananthan, R. Navaratnam, B. Stenger, P. H. S. Torr, and R. Cipolla. Multi-variate relevance vector machines for tracking. In *Proc. 9th ECCV*, volume 3, pages 124–138, Graz, Austria, 2006.

[79] C. Tomasi and J. Shi. Good features to track. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[80] R. Urtasun, L. EPFL, D.J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. In *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, volume 1, Beijing, China, 2005.

[81] R. Urtasun, D.J. Fleet, and P. Fua. 3D People Tracking with Gaussian Process Dynamical Models. In *CVPR'06*, pages 238–245, New York, NY, 2006.

[82] S. Wachter and H.H. Nagel. Tracking persons in monocular image sequences. *CVIU*, 74(3):174–192, June 1999.

[83] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003.

[84] A. Watt and M. Watt. *Advanced animation and rendering techniques.* Addison-Wesley Professional, 1992.

[85] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: real-time tracking of the human body. Master's thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1996.

[86] Ying Wu, John Lin, and Thomas S. Huang. Analyzing and capturing articulated hand motion in image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1910–1922, 2005.

[87] V.M. Zatsiorsky. *Kinematics of Human Motion.* Human Kinetics, 1998.

[88] T. Zhao, T.S. Wang, and H. Y. Shum. Learning a highly structured motion model for 3d human tracking. In *Proceedings of Fifth Asian Conference on Computer Vision*, Melbourne, Australia, 2002.