Accurate Human Pose Tracking

Using Efficient Manifold Searching

Alexandros Moutzouris

Submitted in partial fulfilment of the requirements of

Kingston University for the degree of

Doctor of Philosophy

Digital Imaging Research Centre

Kingston University

MARCH 2013

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

APF Annealed Particle Filter

BC-GPLVM Back Constraint Gaussian Process Latent Variable Model

CSP Colored Surface Points

CSS Covariance Scaled Sampling

GLE Generalised Laplacian Eigenmaps

GMSPPF Gaussian mixture sigma-point particle filter

GPAPF Gaussian Process Annealed Particle Filter

GPDM Gaussian Process Dynamical Models

GPLVM Gaussian Process Latent Variable Model

GRBF Generalized Radial Basis Function

H-APF Hierarchical Annealing Particle Filter

H-GPLVM Hierarchical Gaussian Process Latent Variable Model

HE Human Eva Dataset

HMS Hierarchical Manifold Searching

HMS-MA Hierarchical Manifold Search - Multi Activity

HOG Histogram of Oriented Gradients

HTLE Hierarchical Temporal Laplacian Eigenmaps

IBVH Image-Based Visual Hull

IMS Image and MOCAP Synchronized Dataset

Isomap Isometric Feature Mapping

KF Kalman Filter

LC Limb Correction

LE Laplacian Eigenmaps

LELVM Laplacian Eigenmaps Latent Variable Model

LLE Local Linear Embedding

LSH Locality Sensitive Hashing

MOCAP Motion capture

MP Manifold Projection

MPLC Manifold Projection Limb Correction

PCA Principal Component Analysis

PF Particle Filter

RBFN Radial Basis Function Network

RVM Relevance Vector Machine

SFS Shape-From-Silhouette

SGPLVM Scaled Gaussian Process Latent Variable Model

SMA Specialized mapping architecture

ST-Isomap Spatio-Temporal Isometric Feature Mapping

TLE Temporal Laplacian Eigenmaps

VH Visual Hull

Symbols

λ eigenvalues

ω_{h,l_2} mapping function between the hierarchy positions

φ mapping function from $\mathbb{R}^D$ to $\mathbb{R}^d$ dimensional space

φ' mapping function from $\mathbb{R}^d$ to $\mathbb{R}^D$ dimensional space

$\varphi_{h,l}$ mapping function from $\mathbb{R}^{D_{h,l}}$ to $\mathbb{R}^{d_{h,l}}$ dimensional space

$\varphi'_{h,l}$ mapping function from $\mathbb{R}^{d_{h,l}}$ to $\mathbb{R}^{D_{h,l}}$ dimensional space

ξ Number of frames that used for action classification

$D_{h,l}$ dimension of the $p^i_{h,l}$

$d_{h,l}$ dimension of the $q^i_{h,l}$

g^i Global rotation and translation of the human model M at frame i

H^i Obsevation at time i

h_j level j of the hierarchy

k Action type

M^i Human model at time i

p^i Pose of the human model that is expressed by joint angles between body parts at frame i

p^i pose of the model at time $i = 1, ..., n$

$P_{h,l,k}$ Set of poses of the training dataset for action A_k

$P_{h,l}$ set of poses of the training dataset that corresponds to the l-th pose subspace at the hierarchy level h

$p^i_{h,l}$ pose of the human model at time i

q^i points of the manifold $Q, j = 1, ..., n$

$Q_{h,l,k}$ Set of manifolds for action A_k

$Q_{h,l}$ manifold at the (h, l) position of the hierarchy

$q^i_{h,l}$ points of the manifold $Q_{h,l}, i = 1, ..., n$

s_1 First part of the obsevation function

s_2 Second part of the obsevation function

A Adjacent temporal neighbours for point p^i

D dimension of the p^i

d dimension of the q^i

f Obsevation function

G graph

g Global rotation and translation of the human model M

H observation

h level of the hierarchy

K Number of actions

L Number of parts of human model M

l	subspace of the hierarchy
M	Human model
m	Length and radius of the cylinders of the L body parts
N	total number of poses of the training dataset P
P	training dataset poses at the high dimensional space $\mathbb{R}^D$
p	Pose of the human model that is expressed by joint angles between body parts
Q	manifold representation of P
R	repetition temporal neighbours for point p^i
W	weights

Abstract

In this thesis we propose novel methods for accurate markerless $3D$ pose tracking. Training data are used to represent specific activities, using dimensionality reduction methods. The proposed methods attempt to keep the computational cost low, without sacrificing the accuracy of the final result. Also, we deal with the problem of stylistic variation between the motions seen in the training and the testing dataset. Solutions to address both single and multiple action scenarios are presented.

Specifically, appropriate temporal non-linear dimensionality reduction methods are applied to learn compact manifolds that are suitable for fast exploration. Such manifolds are efficiently searched by a deterministic gradient-based method.

In order to deal with stylistic differences of human actions, we represent human poses using multiple levels. Searching through multiple levels reduces the effect of being trapped in a local optimal and therefore leads to higher accuracy. An observation function controls the process to minimise the computational cost of the method.

Finally, we propose a multi-activity pose tracking methods, which combines action recognition with single-action pose tracking. To achieve reliable

online action recognition, the system is equipped with short memory.

All methods are tested in publicly available datasets. Results demonstrate their high accuracy and relative low computational cost, in comparison to state-of-the-art methods.

Declaration

I hereby declare that this thesis entitled "Accurate Human Pose Tracking Using Efficient Manifold Searching " is the result of my own research except as cited in the references. This thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Parts of this thesis have been published in the following:

- A. Moutzouris, J. Martinez-del-Rincon, M. Lewandowski, J.-C. Nebel, and D. Makris, "Human pose tracking in low dimensional space enhanced by limb correction," International Conference on Image Processing, 2011.

- A. Moutzouris, J. Martinez-del-Rincon, J.-C. Nebel, and D. Makris, "Human Pose Tracking by Hierarchical Manifold Searching," International Conference on Pattern Recognition, 2012.

Acknowledgement

I would like to thank my supervisor Dimitrios Makris for his support and guidance throughout my PhD.

Also, my second supervisor Jean-Christophe Nebel for his comments and critical view of this work. Special thanks to Jesus Martinez-del Rincon for the useful discussions and comments. Moreover, I would like to thank all my colleagues at the Digital Image Research Centre for their support and discussions. I would like to thank my parents for their love and support during all my education.

Finally, I would like to dedicate this work to my fiancée Lefkothea Andreou for her endless patience and love.

Alexandros Moutzouris

Chapter 1

Introduction

1.1 Context and Overview

In recent years, technology has become human-centric by evolving ways by which humans interact with electronic devices. Touch-screen interaction, eye-tracking, gesture recognition are examples that demonstrate the central role of humans in modern technological applications.

A rapidly growing research area in computer vision is articulated human motion analysis, not only because of the methodological challenges that are implied, but also because of its many applications. As described by Moeslund et al. [64], there are three categories of applications: surveillance, control, and analysis (Figure 1.1).

Surveillance applications aim to automatically understand human motions for monitoring and security reasons (Figure 1.1c), such as recognition of unlawful activities, elderly people fall detection, etc. Control applications allow the user to interact with a device through physical gestures and movements, e.g. in games or virtual reality (Figure 1.1b). Finally, analysis applications aim

to specify characteristics of articulated motion, e.g. sports analysis applications (Figure 1.1a).

Articulated human motion analysis consists of pose estimation, pose tracking and action recognition methodologies. Pose estimation deals with estimating the skeletal position of a person for a single image. Pose tracking aims to find the sequence of human poses in video footage. Pose tracking normally exploits the temporal coherence between consecutive frames, and, therefore, may be more accurate than pose estimation. Finally, action recognition aims at classifying the type of action of a human being,

Marker-based approaches have been used for pose tracking applications. Motion capture systems are used to provide the $3D$ position of a set of markers using mechanical, electro-magnetic or optical features. The results are accurate, but the invasive nature of technical equipment limits the applications to controlled environments. Marker-based approaches have been used in the movie industry (Figure 1.1a), where the pose is estimated and is used for featuring a virtual person or subject. Also, marker-based techniques have been popular for analysing human articulated motion in sports analysis applications.

Markerless approaches do not require subjects to wear special equipment for tracking. Human pose tracking methods that rely on markerless approaches are generally desirable because of their non-invasive nature, which significantly widens their potential application. Multi-camera systems are able to mitigate the complexity of markerless approaches and to deal with the inevitable limb occlusions. In addition, depth sensors, such as Microsoft Kinect, are easy to install and use while the captured depth information can be utilised efficiently for $3D$ pose estimation and tracking.

In this thesis we propose methods for accurate markerless $3D$ pose tracking, where human motion is recorded by multi-camera systems or by the Microsoft

Kinect device. Training data is used to represent specific activities such as walking, jogging, punching and kicking, using non-linear dimensionality reduction methods. The proposed methods attempt to keep the computational cost of pose tracking low, without sacrificing the accuracy of the final result. Also, we deal with the problem of stylistic variation between the motions seen in the training and the testing dataset. Solutions to address both single and multiple action scenarios are presented.

(a)

(b)

(c)

Figure 1.1: Motion analysis examples a) MoCap data for $3D$ movies (www.awn.com) b) Microsoft Kinect uses in medical (www.kinectwindows.org) c) Surveillance applications [12].

1.2 Aim and Objectives

The aim of this thesis is to deal with the problem of markerless $3D$ human pose tracking in single and multi-activity scenarios. We assume that the input to our system is either synchronised video sequences captured by multiple cameras, or synchronised sequences of RGB and depth images acquired by the Microsoft Kinect device, as shown in Figure 1.2. In addition, some offline data is provided for the training of the system. The output of pose tracking is a sequence of $3D$ poses, one pose for each set of synchronised input frames.

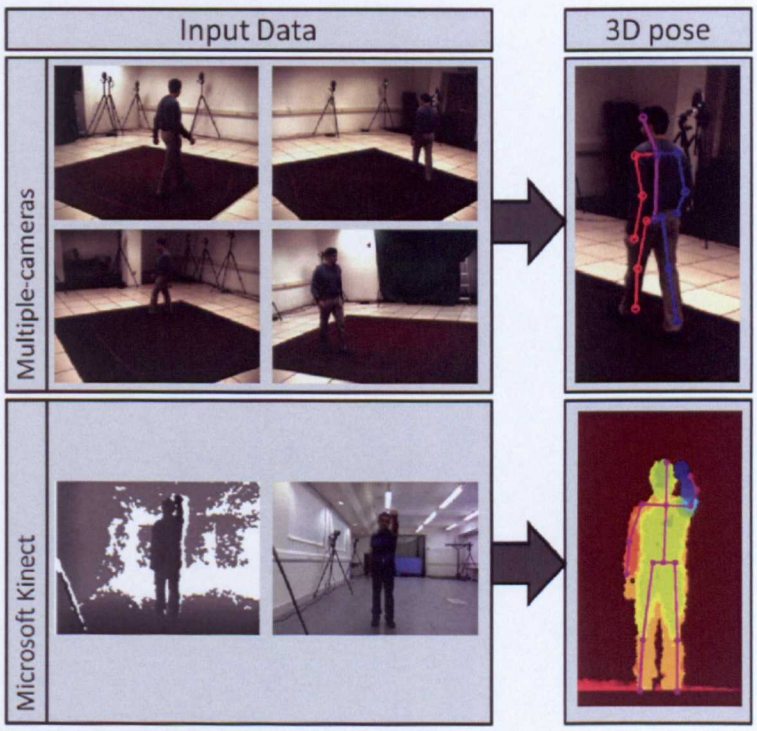


Figure 1.2: Input data and output $3D$ pose for pose tracking method.

The high dimensionality and non-linear space of human postures make the estimation of the optimal pose solution both difficult and computationally expensive. In this work, both the accuracy and the computational complexity will be considered in the evaluation of methodologies.

A specific aspect of the human pose space is the variety of human activities. While many markerless motion capture systems focus on a specific activity,

$3D$ pose tracking in multi-activity scenarios is a challenging problem, which is investigated here.

Another aspect of the complexity of the human pose space is attributed to the stylistic differences of activities, as performed by various human subjects. Such stylistic variations may be affected by anatomical, environmental, cultural or other differences. Therefore, a specific requirement for the proposed methodologies is the ability to track poses that may be stylistically different from any training data.

1.3 Contributions of this Thesis

The main contributions of this thesis are as follows:

- A novel $3D$ human pose tracking method for a specific action (MPLC): Firstly, the Manifold Projection (MP) module searches in low dimensional space for the optimal pose. In order to move beyond the boundaries of the training dataset and generate new poses, the Limb Correction (LC) module is used to provide an improved pose estimate by refining individual limb poses. The novelty of the MPLC method is that combines the advantages of MP and LC; therefore, highly accurate and precise results are derived with low-computational cost.

- A novel hierarchical dimensionality reduction method (HTLE), and a tailored hierarchical $3D$ human pose tracking method for a specific action (HMS): The HTLE approach allows for modelling each level of a posture hierarchy separately, thus representing unseen poses. Furthermore, HMS efficiently searches for optimal poses through the hierarchical structure of HTLE.

- A novel $3D$ human pose tracking method for multi-activity scenarios (HMS-MA): Multiple activities are modelled by multiple hierarchies of manifolds, generated by HTLE. For every frame of unseen video footage, HMS-MA performs both pose tracking and action recognition. Specifically, an online action recognition method is used to reduce the problem to single-action pose estimation. Then, the pose is estimated based on the matched hierarchy of manifolds.

1.4 Structure of Thesis

In Chapter 2, an overview of previous work on human motion analysis, focusing on research related to this thesis, is discussed. The pose estimation and tracking category of human motion analysis relies on two main categories: discriminative and generative approaches. The two categories are discussed and compared in terms of applications, advantages and disadvantages in pose tracking systems. Finally, dimensionality reduction methods and their application in human pose tracking are presented.

In Chapter 3, background information that is important in the context of this thesis is presented. First, a description and the internal parameters of the devices that are used are discussed. Then, the datasets that are used to validate the contributions to the thesis are presented. After that, a $3D$ human body model is introduced. Finally, the generation of an observation function from the input data is presented.

In Chapter 4, a novel $3D$ human pose tracking method, Manifold Projection - Limb Correction (MPLC), is presented. First, a low-dimensional manifold is generated from a sequence of poses of a given training dataset. The manifold represents a specific action and is created using the TLE dimensionality reduc-

tion method. Then, the MPLC method is applied in order to track a sequence of poses of a human action. This action is the same type as the training dataset. The MP method is applied in order to search in the low-dimensional space for the optimal pose. A deterministic optimisation method is used to avoid computationally expensive particle filtering methods. The result of the MP method is a $3D$ human pose that is constrained by the training dataset. In order to evaluate unseen poses beyond the training dataset, the LC method is applied. The LC method first uses a criterion for detecting the body parts that have failed during the MP method, and then searches for those parts in the high dimensional pose space. The MPLC method is compared with the particle filter method in a publicly available dataset.

In Chapter 5, a hierarchical $3D$ human pose tracking framework is presented. First, the hierarchical dimensionality reduction method, Hierarchical Temporal Laplacian Eigenmaps (HTLE), is introduced. HTLE uses a training dataset for a human action to generate a hierarchy of manifolds in low-dimensional space. Moreover, the novel human pose tracking method, Hierarchical Manifold Search (HMS), is applied to estimate efficiently the position of the corresponding body parts. HMS searches into the hierarchy generated by HTLE. At every level of the hierarchy, different sets of joint body parts are tested. Such a hierarchy provides increasing independence between limbs, allowing higher flexibility and adaptability that result in improved accuracy. Finally, evaluation using public datasets demonstrates our approach outperforms state-of-the-art generative methods in terms of accuracy and computational cost.

In Chapter 6, the methodology of HMS is extended to cope with multi-activity scenarios (HMS-MA). The HMS-MA uses the HMS method for searching in different hierarchies of manifolds. Every hierarchy represents a single activity generated by HTLE. The optimal pose for the first level of the hierarchy characterises the action type. Then the HMS method is applied for the rest of the levels

of the chosen hierarchy. Finally, the HMS-MA method is evaluated using two types of public datasets derived by either multiple RGB cameras or the Kinect device.

Finally, in Chapter 7, conclusions and future work are presented.

Chapter 2

Literature Review

2.1 Introduction

This chapter presents an overview of previous work on human motion analysis, focusing on research related to the work presented in the following chapters. More analytic reviews of the literature can be found in a number of review papers, e.g. [5, 63, 64, 75, 74, 112, 92].

Human motion analysis methods have been categorised in three main classes: pose estimation, pose tracking and action recognition. Pose estimation and pose tracking aim at estimating the skeletal position of a person (body pose) for a single frame [39, 7, 108, 42, 65, 62] and a sequence of frames [1, 2, 87] respectively. Action recognition aims at classifying the type of action of a human being. In the following sections, we focus on the literature review of the pose estimation and tracking methodologies. In the last section a discussion about the methods presented is given.

2.2 Pose Estimation and Tracking

In this section the pose estimation and pose tracking categories of human motion analysis are presented. Human pose estimation deals with the problem of determining the $2D$ or $3D$ coordinates of human body parts from a single image. Similarly, human pose tracking deals with the problem of determining the locations of human body parts from a sequence of images (frames). Consequently, pose estimation is often integrated within a pose tracking framework. Pose tracking contributes towards a partial solution to the human motion analysis problem by using past information to estimate the current pose in a more efficient way. Tracking exploits the temporal coherence of video sequences to estimate pose parameters over time. However, due to the complexity of human actions, the localisation of each body part separately is a challenging task. A human pose estimation method may be applied in a tracking scenario but without using the temporal coherence. That is, however, firstly, computationally expensive because in every frame the pose estimation method has to search all the space for the correct pose, and, secondly, there is a loss in accuracy as the information of the previous frames is not being used, e.g. in corrupted frames or frames with limb occlusions. Overall, pose tracking is faster and more accurate in tracking problems but pose estimation can by used for initialisation.

There are two main classes of human pose estimation and tracking methods [74]: discriminative (model-free) and generative (model-based) approaches. Generative approaches use an a priori human body model to generate pose hypotheses. From the input data an observation is produced. The pose hypothesis is compared with the observation using a likelihood or observation function. Discriminative approaches do not use an a priori model, but use a mapping function instead which directly compares the observation space to the pose space.

2.2.1 Discriminative Approaches

Discriminative or model-free approaches model and predict human poses directly from observation. An explicit human body model is not required for these methods. Instead, a mapping function from image space to pose space, learnt from selected training data, is given. Discriminative approaches can be used for initialisation because they do not need a pre-defined human model. Therefore, they have the ability to automatically reinitialize in a tracking application. Discriminative approaches can also deal with poses with less information, e.g. frames with limb occlusions or missing parts. This makes them appropriate for monocular applications. There are two main classes for discriminative estimation: learning-based and example-based [74].

2.2.1.1 Example-Based

In example-based approaches [70, 81, 73] a large database of exemplars is used that describe poses in image space and pose space. Applying an observation function, the optimal matching image is used to give the associated pose. The image descriptors that can be used vary in the literature, usually image descriptors based on edges [3, 70, 94] or silhouettes, [41, 30] or a histogram of oriented gradients (HOG) [76, 73]. Also, the number of cameras is an important parameter. The accuracy of the pose estimation increases with the number of cameras that are used [37]. A drawback of example-based approaches is that, for the provision of satisfactory accuracy and generalization properties, they require the storage and searching of large training datasets [66]. Previous works [85] addressed this problem using locality sensitive hashing (LSH) for faster retrieval of matching exemplars.

2.2.1.2 Learning-Based

In learning-based approaches [1, 4, 94] a continuous mapping is learned between image space and pose space using training data. Agarwal et al. [1] use both regularized least squares and relevance vector machine (RVM) [100] to generate a mapping between histograms-of-shape-contexts and pose. Rosales et al. [82] cluster the training data by generating several forward mapping functions from image to body pose parameters and an inverse mapping function using specialized mapping architecture (SMA), a nonlinear learning model. Dimensionality reduction methods are often used to learn the mapping between the image space and pose space. For example, Elgammal and Lee [31], in order to learn a non-linear manifold from datasets, use the local linear embedding (LLE) dimensionality reduction method. Grauman et al. [37] describe a distribution over both multi-view silhouettes and $3D$ joint locations with a mixture of probabilistic Principal Component Analysis (PCA). The main problem that they address is the generation of the mapping between image and pose, and the ability to connect the two spaces. An advantage of learning-based approaches is that the training dataset is represented through the mapping; therefore, there is no need for storage and searching of large training datasets, as in the example-based approaches. However, the accuracy of discriminative approaches, either example-based or learning-based, rely on the similarity between the unseen poses and the training dataset [110]. For this reason, the selected training dataset must be carefully selected to match the testing scenario.

2.2.2 Generative Approaches

The generative or model-based approaches use a human body model to compare the input image observation with the pose hypothesis using an observation function. The body model is projected into the image observation and the aim

is to maximise the observation function between the hypothesis and the observation. The model that can be used varies from $3D$ kinematic tree [57, 10] to individual limbs, like cylinders [29], blobs [20, 19] or superquadrics [34]. The high-dimensional space of the parameters of the human body model makes the problem quite complicated and computationally expensive, especially for pose estimation. Therefore, generative approaches are used for tracking tasks where pose is initialised for the first frame of the sequence. This gives the advantage of searching for small changes in every frame. On the other hand, generative approaches seem more suitable than discriminative approaches for multi-camera scenarios and produce more accurate tracking results, especially when the testing dataset differs significantly from the training dataset [9, 11, 89]. There are two main approaches for model-based estimation, top-down and bottom-up [74].

2.2.2.1 Bottom-up

In bottom-up approaches, individual body parts are found and then brought together into a human body. For every part an observation function is defined in order to compare the image space with the model parts. Mori et al. [67] perform image segmentation based on contour, shape and appearance cues. The data set that is used is a collection of sports news photographs of baseball players, varying dramatically in pose and clothing. Features like colour, corners or edges are used to detect body parts. Similary, Ren et al. [80] detect candidate body parts using the assumption that parts of the human body can be characterized by a pair of parallel line segments. Sigal et al. [91] propose a body model in which the limbs have elastic connections. For every node a likelihood function is defined. The tracking method takes into account the constraints and the observations to estimate the distribution over the parameters. Bottom-up approaches have the advantage that no initialisation is needed for pose tracking problems. A drawback of bottom-up methods is that many false positives appears on an image, as there

are many regions in an image with limb-like appearance.

2.2.2.2 Top-down

In top-down approaches an a priori human body model is used. The model is predefined based on the application and the scope of the task, and different body parts may be used. The problem is to match the hypothesis of the human body to the image observation. Because of the high dimensional space of the human body model and the high number of degrees of freedom between body parts, the top-down approaches are computationally expensive. For that reason, the problems are limited in the human pose tracking tasks, so the previous pose can limit the searching space for the next prediction. Searching in a local area can give good results, but it is still computationally expensive [35, 16].

To deal with this problem, gradient descent optimisation algorithms have been used. For instance, Delamarre and Faugeras [26] use gradient descent and physical forces between extracted silhouettes and the projected model. These forces guide the minimization of the differences between the pose of the $3D$ model and the pose of the real object in the video images. However, methods based on gradient descent optimisation algorithms may fail to find the global optimum solution, as they may converge to a local optimum. On the other side, methods based on Kalman filtering (KF) [45, 36] and particle filtering [28, 43] use a dynamic model to predict the current pose, based on the motion history. In order to avoid local optimum traps, multiple hypothesis tracking is adopted. In such approaches, multiple pose hypotheses are evaluated and propagated, either by a set of Kalman filters [21] or by particle filter (PF) [8]. Clam et al. [21] represent the modes of the state distribution as a mixture of a few Gaussian functions. The particle filter method uses multiple random predictions (particles), obtained by drawing samples of pose and location priors, then propagating them using the

dynamic model, which is refined by comparing them with the local image data using the likelihood. However, the high dimensionality of this space makes it difficult to sample the solution space efficiently [89] and prevent divergence.

Deutsher et al. [29, 27] proposed an annealed particle filter (APF) that improves the efficiency of the particle filter in order to search the high dimensional human pose space. APF attempts to recover the single pose that maximises the observation function. The algorithm employs a number of re-sampling stages or layers each time. According to a comparative study, [89], APF outperforms all other competitors and is considered state of the art. A drawback of particle filters is that satisfactorily accurate solutions may only be reached by deploying a large number of particles and, therefore, a large number of evaluations of the observation function are performed [89, 11], which increases the complexity and computational cost of pose tracking.

2.2.3 Dimensionality Reduction

In order to deal with the high complexity of modeling articulate human motion, dimensionality reduction methods have been used in either discriminative or generative tracking pipelines. Dimensionality reduction is defined as the process of reducing the number of dimensions of a set of data points in a high dimensional space to a meaningful and compact representation of a reduced dimensional space.

Linear dimensionality reduction techniques were applied for human pose tracking. Ormoneit et al. [71] use Principal Component Analysis (PCA) and particle filter for tracking cyclic motion actions. Urtasun et al. [102] used PCA in combination with a simple hill-climbing optimisation method to avoid computationally expensive multi-hypothesis probabilistic methods. Similarly, Sidenbladh et al. [88] use PCA and local optimisation for human pose tacking. However, lin-

ear dimensionality reduction techniques fail to model properly the non-linearity of human motions.

In order to deal with this problem, non-linear dimensionality reduction techniques have been suggested for pose tracking problems. Non-linear dimensionality reduction techniques are grouped into two categories: embedded-based and mapping-based approaches.

Mapping-based approaches employ probabilistic nonlinear functions in order to map the embedded space on the data space. In this category, methods like Gaussian Process Latent Variable Model (GPLVM) [48, 40] and Scaled Gaussian Process Latent Variable Model (SGPLVM)[38, 48] are included. A drawback of mapping-based approaches is the high computational cost of the learning process, which limits their usage to small datasets [53].

Tian et al. [99] use GPLVM and particle filtering for $2D$ body pose tracking. This method is able to track poses that are similar to the poses in the training dataset. Therefore, the method may fail when the poses deviate significantly from the training data. Urtasun et al. [104] use SGPLVM to learn prior models of human pose for pose tracking of two motions: golfing and walking. The SGPLVM is used because the manifold can be learned from a much smaller amount of training data than by using competing techniques such as LLE [31], LE [93]. Darby et al. [25] use GPLVM and APF for the pose tracking of unknown human motions to reduce the computational cost of the APF method.

Embedded-based or spectral approaches provide an estimate of the structure of the underlying manifold by means of approximating each data point according to their local neighbours on the manifold. Embedded-based approaches can learn the non-linear mapping from the pose space to low-dimensional space, but they cannot be inverted. However, they can handle large and high dimensional datasets with an acceptable computational cost. In this category, methods

like Isometric Feature Mapping (Isomap) [98], Locally Linear Embedding (LLE) [83], Laplacian Eigenmaps (LE) [13] are included.

The inverse mapping from the embedding space to the full pose space is required for evaluation of the observation function of the full pose representation. A possible solution of this is to first learn the embedding space and then do the inverse mapping. Wang et al. [111] used Isometric Feature Mapping (Isomap) for learning the embedding space and a method based on nearest neighbours to learn a mapping of the full pose space. However, this mapping is generally discontinuous and therefore inappropriate for continuous optimisation. Elgammal et al. [31] used LLE to learn activity manifolds from visual input data, and to learn mapping functions between manifolds and both visual input space and $3D$ body space using the Generalized Radial Basis Function (GRBF) [72]. Lewandowski et al. [53] used an unsupervised Radial Basis Function network (RBFN) in order to generate mapping functions between low and high dimensional spaces.

Sminchisescu and Jepson [93] used LE to learn the embedding and Covariance Scaled Sampling (CSS) [95] for tracking. Lu et al. [56] used the Laplacian Eigenmaps Latent Variable Model (LELVM), an extension of LE, to produce a probabilistic latent variable model and the Gaussian mixture sigma-point particle filter (GMSPPF) [106] for pose tracking by using monocular video.

Since human motion may be described by time series, the temporal dependencies between consecutive poses can assist human pose tracking. These temporal constraints ensure that points that are close in time will be close in the low-dimensional space. Spatio-temporal Isomap (ST-Isomap), [44] an extension of Isomap, changes the original weights in the graph of local neighbours in order to emphasize the similarity between temporally related points. Also, Back Constraint GPLVM (BC-GPLVM) [49] uses temporal coherence constraints to generate smooth mapping between spaces. Gaussian Process Dynamical Mod-

els (GPDM) [108] integrate time information using Gaussian Process priors to represent dynamics in the low-dimensional space.

Urtasun et al. [103, 105, 104] use GPDM for learning human poses and motion priors for $3D$ people tracking. They formulate the method as a nonlinear least-squares optimization problem. Hou et al. [40] use BC-GPLVM, which makes particle propagation more efficient. However, most of these methods are person dependent; that is, they are not able to efficiently track new people and their corresponding style from the training set and, therefore, the applicability of the method is reduced.

Alternatively, Temporal Laplacian Eigenmaps (TLE) [53] was specifically designed to address the issue of modelling activities of different people by suppressing their stylistic differences and producing a coherent manifold. As seen in Figure 2.1, TLE is able to suppress stylistic variation and produce more compact manifolds which may be considered almost $1D$ in most cases and, therefore, is suitable for fast exploration. Moreover, the low-computational cost and the generalisation abilities make it appropriate for larger datasets. On the other hand, Rincon et al. [58] proposed a similar method, the Generalised Laplacian Eigenmaps (GLE), that explicitly represents stylistic variations using extra dimensions. They control the balance between the temporal and repetition temporal neighbours by introducing a weighting factor. Low values of it discard the stylistic variations and high values discard temporal information. Since TLE is adopted in this thesis, a further discussion is presented in section 4.2.1.

Hierarchical dimensionality reduction techniques have been proposed to extend the pose space by decoupling the motion of individual body parts, which allows them to deal with unseen activities. First, a hierarchy of the human body model is defined and then a dimensionality reduction method is applied at every level of the hierarchy. An example of hierarchical dimensionality re-

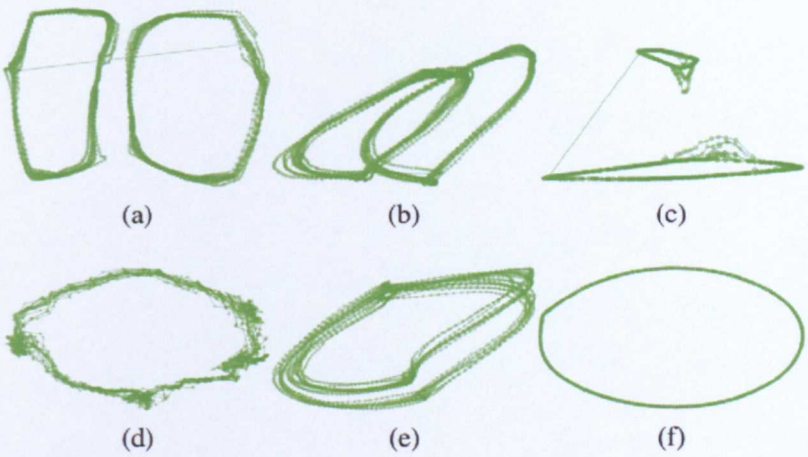


Figure 2.1: Low-dimensional space for walking action (2 subjects) using a) Isomap, b) BC-GPLVM, c) LE, d) ST-Isomap, e) GPDM and f) TLE [54].

duction method is the Hierarchical Gaussian Process Latent Variable Model (H-GPLVM) [50]. The H-GPLVM is an extension of GPLVM with a hierarchical low-dimensional space representation.

H-GPLVM has been used to create a hierarchy of manifolds trained using different activities [77, 24]. Darby et al. [24] used H-GPLVM for training two different activities and the APF method to search for poses that result from combinations of these activities. Using this learnt hierarchical model for multiple activities they can recover novel poses joining activities. An example of that is given training data for a person walking and a person standing and waving; they are able to detect a person who is walking whilst waving. The hierarchy is able to detect the upper body for the first training action and the lower body for the other one. The combination of those two actions can give novel poses for the training datasets. Similarly, Raskin et al. [77] presented the Hierarchical Annealing Particle Filter (H-APF) method, an extension of the Gaussian Process Annealed Particle Filter (GPAPF) method. They use the H-GPLVM nonlinear dimensionality reduction method to generate a hierarchy of manifolds in the low-dimensional space, and the APF method to generate particles in the latent space. In addition, H-GPLVM has been used in multi-activity scenarios, where the ac-

tion of every frame is estimated before pose estimation. Specifically, the average distance between the estimated pose points and the action manifold is calculated using Frèchet distance [6]. The model with the smallest distance was chosen to represent the type of the action. The advantage of these hierarchical approaches is that new poses can be generated where individual body part postures originally belonged to different activities. However, their main drawback is the high computational cost, since APF is used to search through the whole hierarchy.

2.3 Discussion

In this thesis we deal with the problem of $3D$ human pose tracking. More specifically, we use generative top-down approaches in order to achieve high accuracy. As the top-down approaches are computationally expensive, we use non-linear dimensionality reduction method in order to address the high complexity of articulate pose space.

We explicitly select the temporal dimensionality reduction method TLE [53] to take advantage of the temporal dependencies between consecutive poses in the training dataset. TLE is also used because it is able to produce compact manifolds that may be searched efficiently. However, the drawback of using low-dimensional manifolds is that they are unable to model stylistic variations that are not present in the training dataset, [73], therefore accuracy may be low when pose tracking is applied to sequences of unseen subjects.

Hierarchical extensions of dimensionality reduction [77, 24] are able to represent stylistic variations that have not been seen in the training dataset and, therefore, improve the accuracy of pose tracking. In addition, at the end of the pose search constrained by low-dimensionality manifolds, an extra search step of refining the pose of individual limbs in the original high dimensional space is

adopted. Pose search through different spaces is driven by an observation function to minimise the computational complexity of the proposed methods.

Although particle filters are popular in generative pose tracking pipelines [29, 27, 77, 43, 28], their computational cost is very high, as they generate a large number of random hypotheses to estimate the optimal pose. Alternatively, this thesis adopts the use of deterministic gradient-based optimisation techniques to improve the efficiency of the proposed methods. Although such techniques may be sensitive to initialisation and may be trapped in local optima, in our approach searching is performed through different spaces, therefore the effect of local optima is reduced.

Finally, multi-activity pose tracking is addressed by combining action recognition and single-activity pose tracking. The characteristic of TLE of suppressing stylistic variations has already been exploited in offline action recognition applications [54, 52]. In this work, an online action recognition is based on TLE along with a short memory of observation functions to assign an activity label to each frame.

Chapter 3

Background

3.1 Introduction

In this chapter, background information that is important in the context of this thesis is presented. First, a description and the internal parameters of the devices that are used are presented. Two types of data acquisition systems are used for our experiments: a system of synchronised multiple cameras and the Microsoft Kinect device. Then the datasets that are used to validate the contributions to the thesis are presented.

Since this thesis proposes generative $3D$ pose tracking approaches, a $3D$ human body model is required. The human model that is employed is represented as an articulated kinematic tree model, which is appropriate for top-down methodologies (Section 2.2.2.2). Pose hypotheses are generated based on the human model for every frame. An observation must be provided based on the input data in order to evaluate pose hypotheses. In our framework, first, a background subtraction method is used for removing the background pixels from the input images in order to extract the human subject silhouettes. In the case of multiple-

camera data, the silhouette images are used for the creation of the coloured visual hull of the human subject. In the case of Kinect data, the observation is represented by the foreground colour image and depth map. Finally, an observation function is used to compare the observation with the pose hypothesis. Within the context of this thesis, we assume that the ground truth pose of the first frame is known.

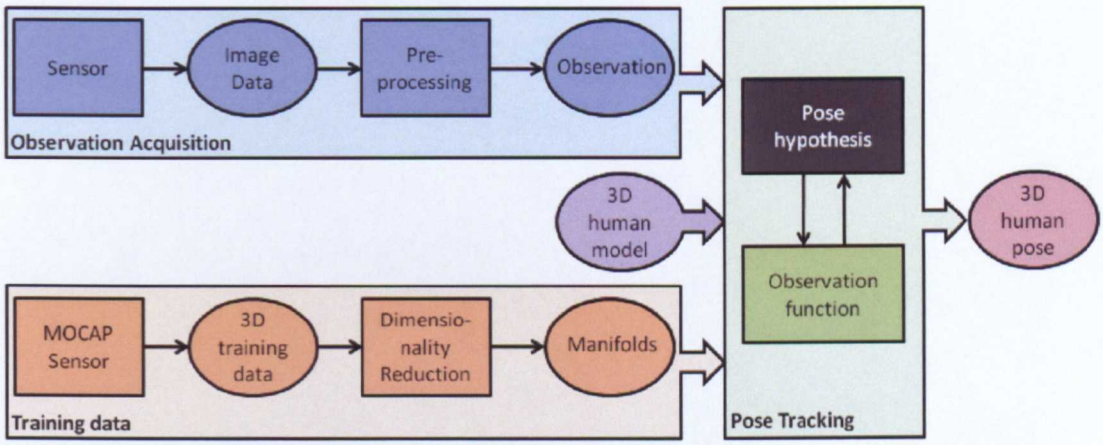


Figure 3.1: Pipeline of pose tracking methods.

In Figure 3.1 we see the general pipeline that we use in this thesis for the pose tracking problem. Dimensionality reduction is applied offline to training data to produce low-dimensional manifolds that represent the action(s) of a given scenario. When applying the pose tracking process on unseen sequences, an observation is constructed for every set of synchronised frames. $3D$ pose hypotheses, based on a $3D$ human model, are normally constrained by the learnt manifolds and are evaluated by an observation function that quantifies the extent to which they match the observation. Finally, the output of pose tracking is the hypothesis that maximises the observation function.

3.2 Data Acquisition

In this section we present the internal parameters of two types of devices, i.e. the multiple cameras and the Microsoft Kinect device. A camera model provides us with the geometric relationship between the image and the real world coordinates. Those parameters are useful not only for understanding the input data, but also for their utility in the next sections.

3.2.1 Multiple Cameras

Multiple cameras are used for the scenarios which are presented in this thesis. Multiple cameras are required to provide multiple views of the human subject and may resolve occlusions between different body parts that may appear on some views. The cameras are normally located around the subject that performs an action inside the captured space, i.e. the area that all cameras can view.

We assume that K cameras are used for the capture of an action. As described in [32, 101], each camera $k, k = 1, \ldots, K$ is characterised by its intrinsic and extrinsic parameters. The intrinsic parameters depend on the internal structure of the camera and are the following:

f^k : Focal length (2×1 vector)

c^k : Principal point (2×1 vector)

γ^k : Skew coefficient

The extrinsic parameters depend on the position and the orientation of the camera with respect to a world coordinate system and are the following:

R^k : Rotation (3×3 matrix)

T^k : Translation (3×1 vector)

In this work, we assume that these parameters are given for each camera. Therefore, we will not deal with their estimation (camera calibration). The above parameters are used to calculate the following parameters.

The intrinsic characteristics of a camera are represented by the camera calibration matrix, defined as the 3×3 matrix A^k :

$$A^k = \begin{bmatrix} f^k(1) & \gamma^k & c^k(1) \\ 0 & f^k(2) & c^k(2) \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.1}$$

Based on the calibration data (A^k, R^k, T^k), a 3×4 projection matrix P^k is produced

$$P^k = A^k[R^k|T^k] = A^k \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R^k & T^k \\ 0_{1,3} & 1 \end{bmatrix}. \tag{3.2}$$

The projection matrix describes the mapping of a camera from $3D$ points in the world to $2D$ points in an image. As seen in [32], we express the relationship between $2D$ pixel m^k and $3D$ point M (Figure 3.2), as

$$m^k = P^k \cdot \begin{bmatrix} M \\ 1 \end{bmatrix} = \Pi^k(M) \tag{3.3}$$

where

$$\Pi^k() : \mathbb{R}^3 \rightarrow \mathbb{R}^2 \tag{3.4}$$

is the perspective projection function, C^k is the centre of the camera, $m^k =$

$[u^k, v^k, 1]^T$ is measured in the image coordinate system (x_{c^k}, y_{c^k}) and $M = [X, Y, Z]^T$ is measured in the world coordinate system $(X_{C^k}, Y_{C^k}, Z_{C^k})$. A camera for which P^k or Π^k is known, is said to be calibrated.

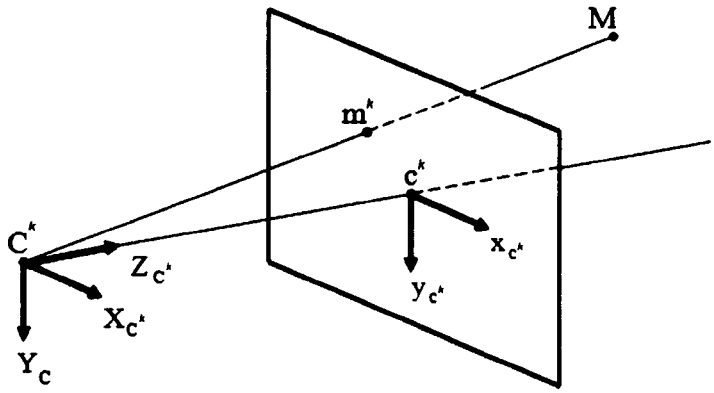


Figure 3.2: Camera model.

3.2.2 Microsoft Kinect Device

Microsoft Kinect [61] is a device that can provide synchronised sequences of colour images and depth images. Although Kinect was designed for computer game applications, it has also been used in other applications [84, 109, 59]. Kinect contains an RGB sensor and a depth sensor (infrared camera), as seen in Figure 3.3. The RGB sensor is actually a camera, as described in the previous section. The depth sensor provides a depth map image for every frame. Every pixel of the depth map represents the distance from the corresponding $3D$ point to the sensor. It produces satisfactory depth results when the subject is within 1 to 3 meters distance of the sensor [46]. Kinect has also integrated a state-of-the-art pose estimation method [86] based on the acquired depth data, which provides the position of joints of observed humans.

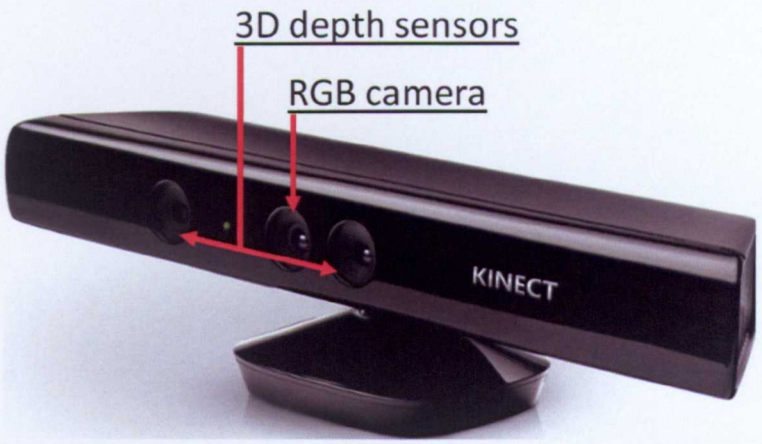


Figure 3.3: Microsoft Kinect devises: RGB camera and $3D$ depth sensors [61].

3.3 Datasets

In this thesis, we use three data sets to evaluate our contributions: a) the Image & MOCAP Synchronized Dataset (IMS), b) the HumanEva (HE) dataset, and c) the G3D dataset. The first two are used for comparing our methods with the state-of-the-art methodologies that use the same datasets. We use the G3D dataset to test our work in a multi-activity scenario. All datasets are captured in indoor environment using fixed viewpoint on the captured devises and illumination conditions.

3.3.1 Image & MOCAP Synchronized Dataset

The Image & MOCAP Synchronized Dataset (IMS) [18] is a dataset depicting a walking human. Synchronised data was derived by a motion capture (MoCap) Vicon system [107] and four grey-scale, calibrated cameras. Each camera captured 530 frames of pixel resolution 640×480 at 60 Hz. (Figure 3.4). The Vicon system captured the $3D$ positions of markers, which were then used to estimate the ground truth positions of 15 joints in $3D$.

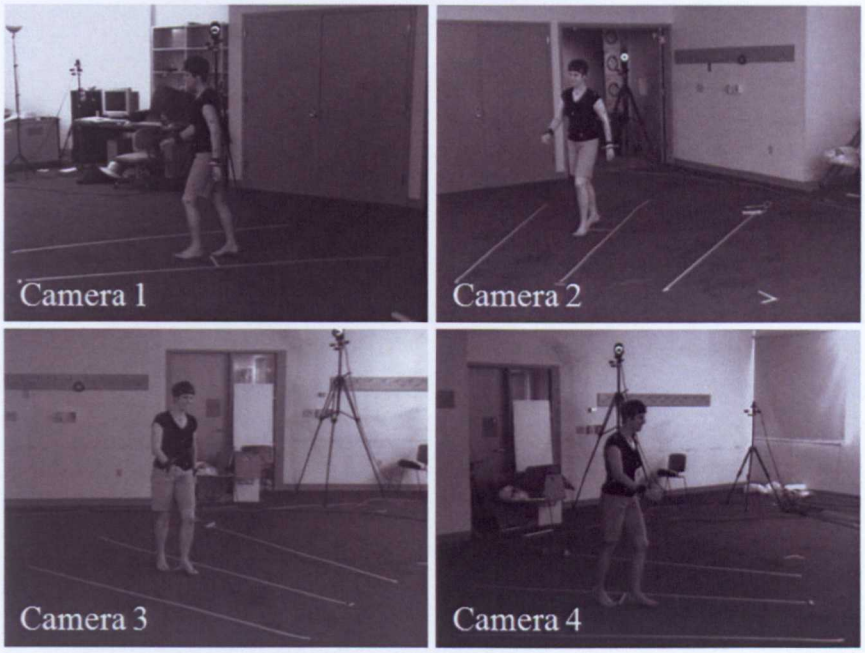


Figure 3.4: "Image & MOCAP Synchronized Dataset". First frame from 4 grey-scale cameras.

3.3.2 HumanEva

HumanEva-I and HumanEva-II datasets [90] represent multiple subjects performing multiple activities (Figure 3.5).

HumanEva-I contains software synchronised data from 7 video calibrated cameras (4 gray-scales and 3 colour) and Vicon motion capture (MoCap) system [107] Gray-scale camera resolution is 644×488, while colour camera resolution is 656×490, and both video systems captured at 60 Hz. HumanEva-I contains 4 subjects performing 6 common actions (e.g. walking, jogging, gesturing, etc.). Similarly, the second dataset, HumanEva-II, contains synchronised data from 4 calibrated colour video cameras and a Vicon system. HumanEva-II contains 2 subjects performing a continuous sequence of actions (walking, jogging, balancing). This dataset provides $3D$ ground truth of the human posture, i.e. $3D$ positions of 15 joints, for some sequences for training and validation purposes, while ground truth for testing sequences is not publicly available. An online system at the Human Eva website [17] provides results on the testing datasets. In

this thesis, we use training and testing sequences for the walking and jogging actions.

A standard metric proposed by Sigal [89] is applied for quantitative evaluation: for each of the joints of the skeleton representation the error is calculated as the average absolute Euclidean distance between M markers of the estimated pose $\hat{X}$ and M markers of the corresponding ground truth pose X, provided by the motion capture system

$$D\left(X, \hat{X}, \hat{\Delta}\right) = \sum_{m=1}^{M} \frac{\delta_m \left\| x_m - \hat{x}_m \right\|}{\sum_{i=1}^{M} \hat{\delta}_i} \tag{3.5}$$

where $x_m \in X$, $\hat{x}_m \in \hat{X}$ and $\hat{\Delta} = \left\{\hat{\delta}_1, \hat{\delta}_2, \ldots, \hat{\delta}_M\right\}$is a binary selection variable per-market where $\hat{\delta}_m = 1$ if the proposed algorithm is able to recover marker m, and 0 otherwise.

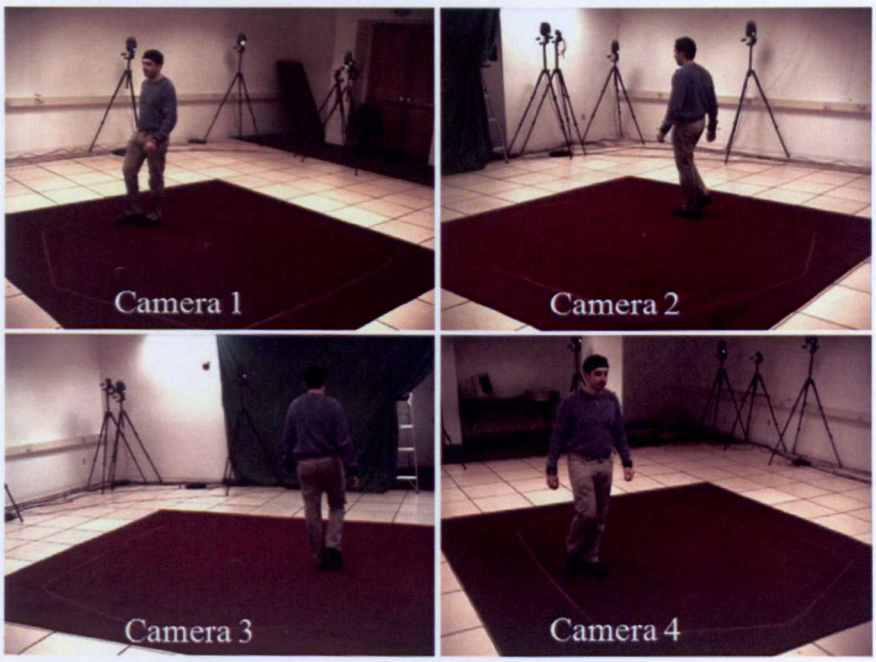


Figure 3.5: HumanEva-II dataset. First action(S2). First frame from 4 colour cameras.

3.3.3 G3D Dataset

The G3D dataset [15] was captured using the Microsoft Kinect device. The dataset contains 10 subjects performing 20 gaming actions. Each subject performs 3 repetitions of each action. The camera captured images with pixel resolution 640×480 at 30 Hz. The actions are grouped into 5 scenarios e.g. boxing, golf, tennis. In our experiments, we use the boxing scenario, which consists of five actions, i.e. punch right, punch left, kick right, kick left and defend (see Figure 3.6). The dataset consists of sequences of three modalities: colour images, depth images and 20-joint poses estimated by the state-of-the-art method [86]. As the ground truth is not available for G3D dataset the latter is assumed as ground truth for training and testing our work. Therefore, the results for this dataset are compared to the Kinect results and not to the ground truth poses.

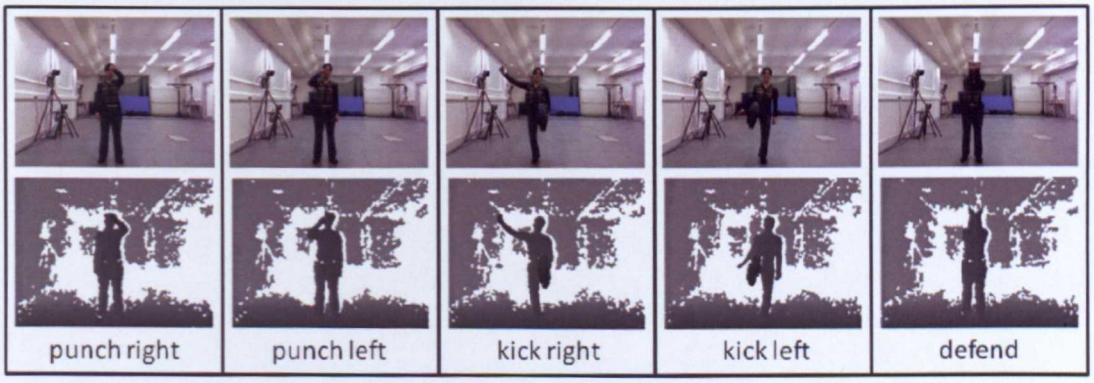


Figure 3.6: G3D dataset example. Colour images and depth maps for five actions.

3.4 Human Pose Hypothesis

A $3D$ human model is used for generating hypotheses of each pose and comparing them with the observation from the input data. The human model that we use is represented as a kinematic tree and it consists of L parts. We use a human model with $L = 10$, i.e. 1 for torso, 4 for leg parts, 4 for arm parts and1 for head. Each part must connect with the corresponding part and may move without being

restricted by the angle. The angles of the human body are not constrained in order to determine from the pose tracking method. In a more accurate description of the model, other parts like feet, hands and an extra part for the torso could be included. For the purposes of this work, two representations are used, i.e. the volumetric one and the skeleton one (Figure 3.7).

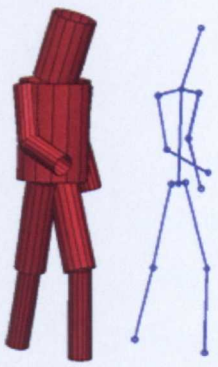


Figure 3.7: Human model and corresponding skeleton representation.

We define the volumetric representation as a $3D$ articulated human model M that consists of L cylindrical parts. The cylindrical definition of the model allows us to compare pose estimates against the observation using simple mathematical models. This is to allow faster evaluation of the observation function without losing the basic structure of the $3D$ human shape.

The human model M is defined as a set of three independent parameters

$$M = \left\{ \{g, p, m\}, g \in \mathbb{R}^6, p \in \mathbb{R}^D, m \in \mathbb{R}^{2L} \right\} \tag{3.6}$$

where $g \in \mathbb{R}^6$ describes the global rotation and translation of the body into the $3D$ Euclidean space, $p \in \mathbb{R}^D$ the pose of the model that is expressed by joint angles between body parts, and $m \in \mathbb{R}^{2L}$ represents the human volumetric model expressed by the length and the radius of the cylinders of the L body parts. Joint angles are represented by quaternions, as a consequence of which every body part

requires four parameters, i.e. $D = 4 \cdot L$.

The Skeleton Representation is a model that is extracted from specific points of the Volumetric Representation. Every part of the human body is represented by a straight line that connects two points as seen in Figure 3.7.

3.5 Observation

From the input data an observation is generated for every frame. Since our input may be acquired by multiple syncronised sensors, the term frame may also mean a set of synchronised frames in this thesis. The observation includes the information that will be used in the pose tracking methodologies. In this section we describe the observation that is generated by pre-processing the input data. The type of observation depends on the input dataset, and it is compared with the pose hypotheses. First, a background subtraction technique is applied to the input data to locate the object of interest, i.e. the silhouette of the human body. Then, in the case of multiple-cameras a coloured visual hull is generated. For Kinect data, the foreground colour image and depth map are extracted. For every dataset the observation is first generated for every frame and then is used in pose tracking method.

3.5.1 Foreground Mask

A foreground mask or silhouette is a pixel-wise binary representation of the area of an object of interest. For Image & MOCAP Synchronized and HumanEva datasets, the standard background subtraction method suggested by HumanEva, a static version of a mixture of Gaussians similar to [96], is used to ensure a fair comparison with other methods. For the G3D dataset, the background subtraction process is simpler as the depth map explicitly gives the depth information

of every pixel. Pixels whose distance from Kinect is larger than a threshold are classified as background.

A foreground colour image may be obtained by applying the foreground mask on the original colour image. Similarly, the foreground depth map is derived by applying the foreground mask on the original depth map.

3.5.2 Visual Hull

The visual hull is a voxel-wise binary representation of the volume of the subject of interest. It may be reconstructed from multiple cameras by the Shape-From-Silhouette (SFS) method, which estimates the shape of an object from its silhouette images.

Every image is segmented into foreground and background areas, as discussed before. When the foreground area is projected into the $3D$ space, using camera models a $3D$ geometric shape is defined that contains the target object. The intersection of all silhouette geometric shapes is the visual hull of the object. As described in [55], there are two main categories of visual hull construction methods, i.e. the voxel-based and the boundary-based methods.

The voxel-based methods result in a $3D$ volumetric visual hull composed of voxels. If a voxel place is inside all silhouette cones, it will be preserved, otherwise it will be cleared. These methods can reconstruct very complex objects, but they cannot get smooth modeling results. Every voxel must pass two tests to be classified as a member of the visual hull: the silhouette cone test and the silhouette consistency test. The former test verifies whether a voxel belongs to a silhouette cone. The latter checks if a voxel passes the former test for all the views. Szeliski [97] used a tree data structure for these two tests. Kutalakos and Seitz [47] suggested an algorithm, called Space Carving, for computing the visual

hull.

In the boundary-based methods, the foreground cones are represented as boundary elements, such as surfaces and lines. The visual hull is represented by the intersection of these elements, and the results could be a group of surface patches, line segments, or points. Matusik et al. [60] presented an algorithm for visual hulls for surface points seen from a target view, called image-based visual hull (IBVH). Cheung et al.[22] reconstructed visual hulls as a set of line segments, which they call bounding edges. In this work, we adopt the Bounding Edge [22] method to estimate the visual hull as it is more accurate than voxel-based methods [22].

3.5.2.1 Bounding Edge Method

Let us assume that there are K fixed cameras positioned around a human body and let

$$\left\{S_i^k, k = 1, ..., K\right\} \tag{3.7}$$

be the set of silhouette images of the human obtained from the K cameras at time i, as described in the section 3.5.1 (Figure 3.8).

The cameras are calibrated and therefore their perspective projection functions Π^k (Eq. 3.3) and centres C^k of each camera k (Figure 3.2) are known. Consequently, $m = \Pi^k(M)$ are the $2D$ image coordinates of a $3D$ point M in the kth camera, and $\Pi^k(S)$ represents the projection of a volume S onto the image plane of camera k. The Visual Hull H^i with respect to a set of consistent silhouette images $\{S_i^k\}$ is defined as the intersection of the K visual cones, each formed by projecting the silhouette image S_i^k into $3D$ space through the camera centre C^k at time i.

Let u_i^j be a point on the boundary of the silhouette image S_i^k. By pro-

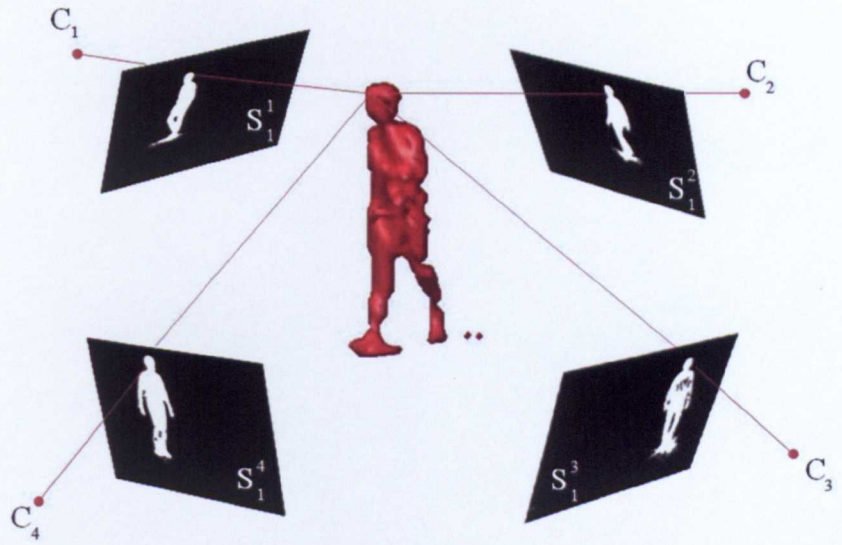


Figure 3.8: Silhouette images for 4 cameras (S_1^k), Visual Hull of the silhouettes and the centre of the cameras (C_k).

jecting u_i^j through the camera centre C^k, we get a ray r_i^j. A Bounding Edge E_i^j is defined to be part of r_i^j, so that the projection of E_i^j onto the l^{th} image plane lies completely inside the silhouette S_i^l for all $l \in \{1, \ldots, K\}$, therefore

$$E_i^j \subset r_i^j \text{ and } \Pi^l(E_i^j) \subset S_i^l \; \forall l \in \{1, \ldots, K\}. \tag{3.8}$$

The bounding edge can be computed by first projecting the ray r_i^l onto the $K-1$ silhouette images S_i^l, $l = 1, \ldots, K$; $l \neq k$, and then re-projecting the segments which overlap with S_i^l back into $3D$ space. The bounding edge is the intersection of the re-projected segments (Figure 3.9).

By sampling points on the boundaries of all the silhouette images $\{S_i^k; k = 1, ..., K\}$, we can construct a list of L_i Bounding Edges that represents the Visual Hull H^i.

We can describe the visual hull H^i at frame i as

$$H^i = \{(x, y, z) : h_i(x, y, z) = 1\} \tag{3.9}$$

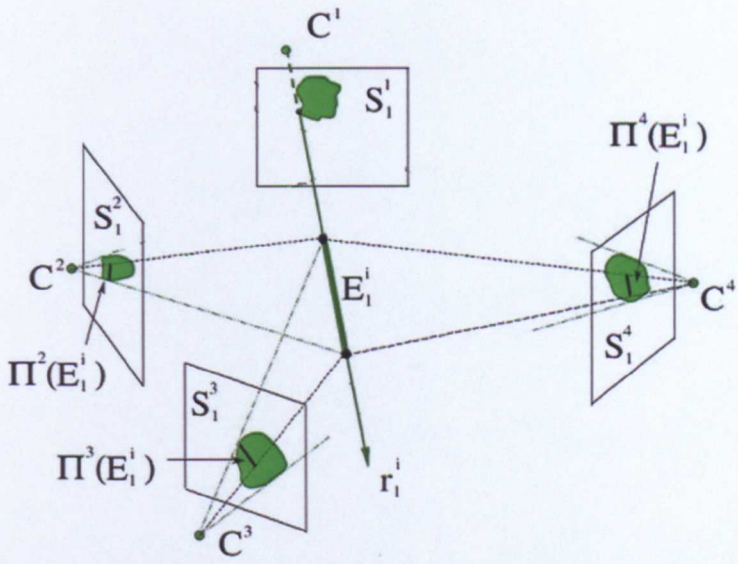


Figure 3.9: Bounding Edge method [22].

where

$$h^i : \mathbb{N}^3 \rightarrow \{0, 1\} \tag{3.10}$$

is represented by the list of the Bounding Edges. Every element h^i represents a voxel in $3D$ space. The Visual Hull consists of all the voxels whose values are 1.

In Figure 3.10 we see the visual hull for several frames of the HumanEva-II S2 dataset. The visual hull quality depends on the the quality of silhouettes and the number of cameras. Below, we present some indicative results from extracted silhouettes for 4 cameras of HumanEva-II S2 dataset.

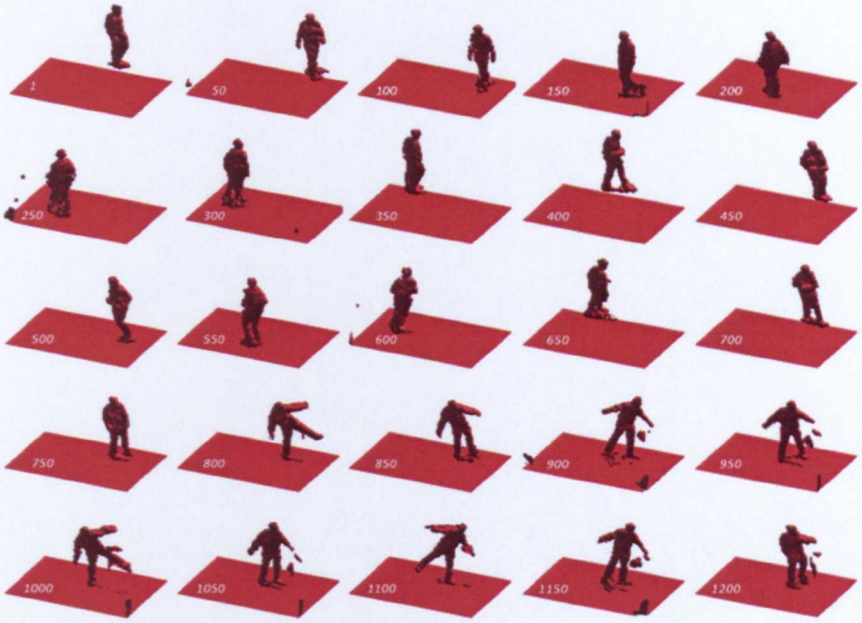
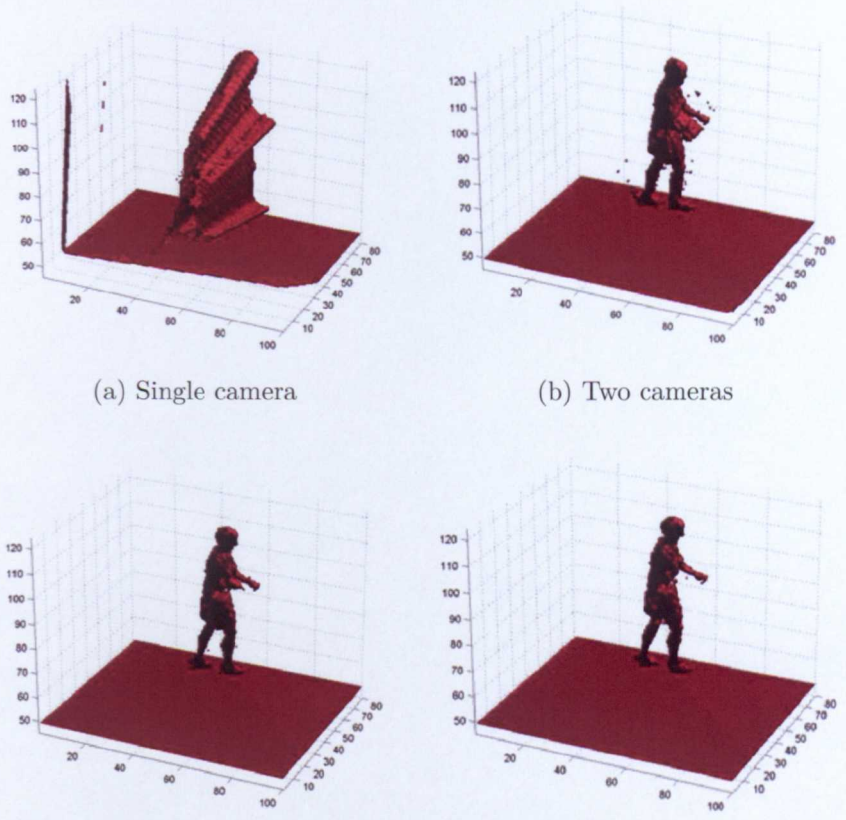


Figure 3.10: Visual hull for HumanEva-II dataset, for the action S2 and for 25 frames.

Using a different number of cameras, the visual hull is shown in Figure 3.11. Using a single camera, one can see the bounding edges from this view, but $3D$ representation is far from the real $3D$ volume of the subject. Using two cameras, limb representation may be inaccurate as some $3D$ volumes are unseen by the cameras. For example, in Figure 3.11b, an erroneous right arm appears in the visual hull. Finally, using three or more cameras, the quality of results is clearly improved and the visual hull can satisfactorily represent the human body for the purpose of this thesis.



(a) Single camera

(b) Two cameras

(c) Three cameras

(d) Four cameras

Figure 3.11: Visual Hull for HumanEva-II dataset, for the action S2 using 1,2,3 and 4 cameras.

3.5.2.2 Coloured Visual Hull

A Coloured visual hull results from the projection of colour information of the images onto the surface of the $3D$ visual hull. In order to generate coloured visual hull objects we use the Colored Surface Points (CSP) technique [23]. Each bounding edge touches the object at at least one point. However, this point is not known from the Bounding Edge method. For estimating this point we assume that any point on the visual hull should have the same projected colour for all the colour images. More specifically, for every point of a bounding edge we calculate the projected colour from camera k. Then the colour mean and the variance of that point are calculated according to the points which are visible by the camera. Finally, the point of the bounding edge with the minimum variance is chosen

(Figure 3.12).

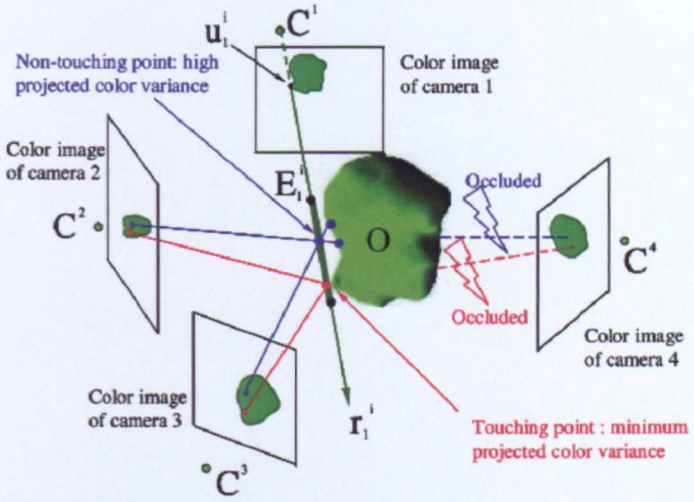


Figure 3.12: Estimation of the Colored Surface Point by searching on the Bounding Edge for the point with the minimum projected colour variance [23].

In Figure 3.13, we see the colour visual hull and one of the corresponding images from the HumanEva-II S2 dataset. The colour visual hull is calculated by projecting the colour information from 4 images of the same frame.

Figure 3.13: Colour Visual Hull and a corresponding frame for HumanEva-II S2 dataset.

3.5.3 Depth Map

The depth map of a $3D$ object is an image every pixel of which represents the distance of the corresponding $3D$ point from the sensor. The observation at Kinect dataset consists of the silhouette image and the depth map for every frame, as discussed in section 3.2.2. First, we apply a foreground mask method, as seen in section 3.5.1. We use the foreground colour image and the foreground depth map as the observation, as seen in Figure 3.14.

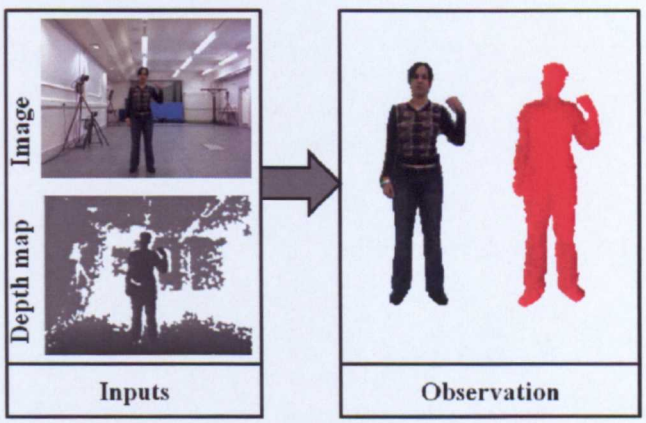


Figure 3.14: The observation generated by the Kinect input data, consisting of two parts: foreground colour image and depth map images.

3.6 Observation Function

In order to compare a pose hypothesis with the observation, we define an observation function or likelihood function. The observation function varies with the type of the dataset and the special needs of every situation. The specific observation function that is used in every dataset is presented in chapters 4, 5 and 6.

For multiple-camera datasets, the observation function compares a $3D$ human model (hypothesis) with a $3D$ coloured visual hull (observation). The observation function is based on the volume and the colour of the two $3D$ objects.

First, the $3D$ overlap between the pose hypothesis and observation is calculated. Then, the colour information, when it is available, is projected on the visual hull. The colour of every limb is compared with the colour of the corresponding limb of the initial frame.

In the Kinect dataset, the observation function is again based on $3D$ information and the colour, but expressed by the colour images and the depth map. First, the $2D$ overlap between the pose hypothesis and observation is calculated. Then, the colour information is matched with the colour of the initial frame. Finally, the depth map is compared with the generated depth hypothesis.

3.7 Discussion

In this chapter we present methods and data that are used in the pipelines proposed in this thesis. First, the datasets that are used and the different devices that are captured are first presented. Two types of devices are used in our experiments, i.e. a multi-camera system and the Microsoft Kinect device. Then the base of pose hypotheses and the process to produce observations are discussed. Pose hypotheses are generated by a $3D$ human model, which is defined here. The observation is generated by the input data. First, a background subtraction method is applied to the input images. Then the $3D$ information of the input data is extracted from the $2D$ input images or from the depth map. Finally, in order to compare the pose hypothesis with the observation, an observation function is required. In the next chapters 4, 5, 6, specific observation functions are defined and evaluated as parts of the proposed pose tracking methodologies.

Chapter 4

Human Pose Tracking in Low-Dimensional Space Enhanced by Limb Correction

4.1 Introduction

This chapter introduces Manifold Projection - Limb Correction (MPLC), a $3D$ human pose tracking method for a specific action. Specifically, a reliable method is required to estimate the pose, with low-computational cost, even when the execution of an action in the training dataset and the current sequence differs stylistically. We follow the general pipeline that was presented in section 3.1, where the pose tracking box in Figure 3.1 corresponds to the MPLC method. A manifold to represent a specific action is learnt by using the Temporal Laplacian Eigenmaps (TLE) dimensionality reduction method.

MPLC consists of two main modules: Firstly, the Manifold Projection (MP) module searches in the low-dimensional space for the optimal pose. However, the result of the MP module is constrained by the training dataset. In order to move beyond the boundaries of the training dataset and generate new poses, the Limb Correction (LC) module provides an improved pose estimate by

refining individual limb poses. The MPLC method combines the advantages of MP and LC and, therefore, highly accurate and precise results are derived with low-computational cost. The validation of our method uses publicly available datasets, and demonstrates its accurate and computational efficiency. Parts of this work have been published in [68].

4.1.1 Overview

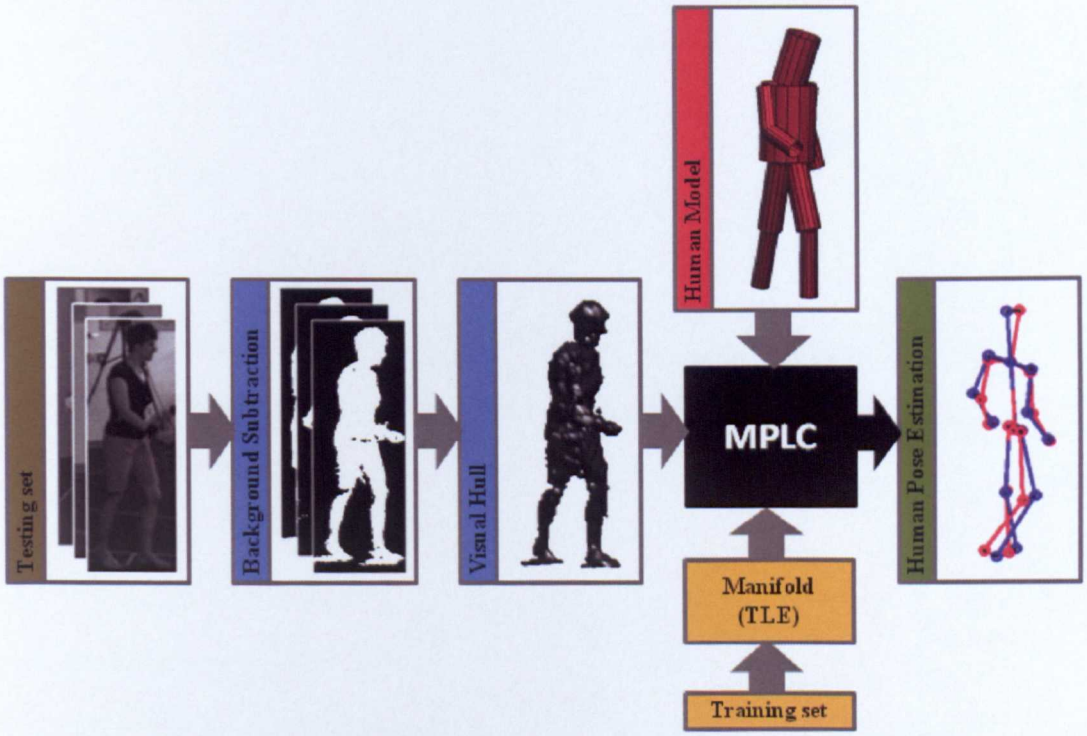


Figure 4.1: MPLC pipeline. Using the testing dataset we generate the observation. A low-dimensional manifold is generated using TLE from the training dataset. The human model is used to generate the pose hypothesis. MPLC method is applied. The output is the human pose estimation for the current frame.

In this section the MPLC human pose tracking pipeline is presented (Figure 4.1). MPLC, as a top-down generative method (see section 2.2.2), requires a human model and an observation of the input. Since top-down generative approaches are computationally expensive due to the high dimensionality of the human pose space, we use a dimensionality reduction method and a training dataset in order to constrain high-dimensional poses in a low-dimensional manifold.

The training dataset consists of a sequence, or a set of sequences of poses (typically MOCAP data) of a person performing a single action. Since accurate tracking requires a temporally smooth and consistent data model, a constraining manifold is generated by Temporal Laplacian Eigenmaps (TLE) [54], which aims for the preservation of the temporal topology present in high dimensional spaces. The choice of the TLE method enables us to suppress stylistic variation and generate more compact manifolds that are efficient for searching, as we have seen in Section 2.2.3. An observation is generated from the testing dataset inputs, as discussed in section 4.2. The poses of the testing dataset represent a human performing a single action. This action is the same as the one that is performed in the training dataset. Finally, a human model is used to generate pose hypotheses that are evaluated by an observation function (see section 4.4).

MPLC, which is introduced in section 4.3, guides pose estimation in two stages. First, in the MP stage, the search is constrained by a TLE low-dimensional manifold. A deterministic optimisation method is used for its computational efficiency, in contrast to computationally expensive particle filtering approaches. The result of the MP method is a $3D$ human pose estimate. However, since the testing and the training datasets are different, the results of the MP stage depend on the similarity of the two datasets. The LC stage allows us to search beyond the training dataset constraints and to evaluate completely unseen poses. The LC module firstly uses a criterion for detecting the body parts that have been erroneously determined during the MP method, and then refines those poses, searching in the high-dimensional pose space. LC is able to generate poses that do not appear in the training dataset, but remain close to them. This approach is advantageous when searching for different styles of a specific action. The MPLC method outperforms state-of-the-art methods for a given computational time when using a publicly available dataset, as presented in section 4.5.

4.2 Action Manifold Learning

4.2.1 Temporal Laplacian Eigenmaps (TLE)

In this section, the Temporal Laplacian Eigenmaps (TLE) dimensionality reduction method is presented. In order to generate the low-dimensional manifold a training dataset P, consisting of a sequence or set of sequences of data points, is used. Our notation in this work assumes only one sequence in the training dataset, without losing generality.

$$P = \left\{p^j, j = 1, ..., n\right\}, p^j \in \mathbb{R}^D \tag{4.1}$$

corresponds to n frames representing an action, where p^j is the pose of the model at time j. TLE produces a manifold Q, which is an equivalent representation of P in a low-dimensional space,

$$Q = \left\{q^j, j = 1, ..., n\right\}, q^j \in \mathbb{R}^d \tag{4.2}$$

where $D, d \in N$, $d \ll D$ and $q^j$the points of the manifold.

For each data point p^j two types of temporal neighbourhoods are defined. Adjacent temporal neighbours A: the $2m$ closest points in the sequential order of input (Figure 4.2a)

$$A^i \in \left\{p^{i-m}, \ldots, p^{i-1}, p^i, p^{i+1}, \ldots, p^{i+m}\right\} \tag{4.3}$$

and repetition temporal neighbours R: the s points similar to p^j, extracted from repetitions of time series fragment, defined by s adjacent temporal neighbours (Figure 4.2b)

$$R^i \in \left\{p^{i,1}(C), \ldots, p^{i,s}(C)\right\} \tag{4.4}$$

where $p^i\,(C)$ returns the centre point of p^i.

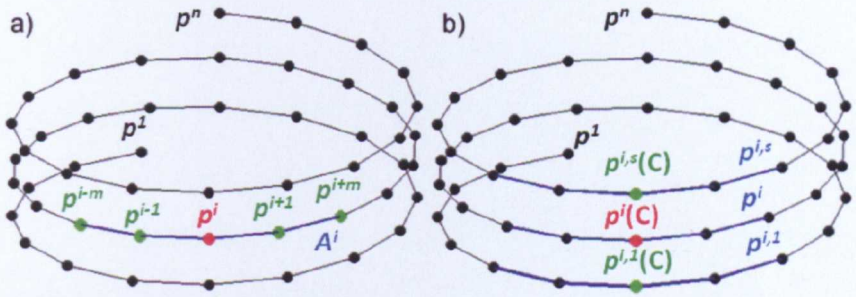


Figure 4.2: Repetition temporal a) and adjacent temporal b) neighbours (green dots) of a given data point, p^i, (red dots).

Using the standard LE formulation the weights W are assigned to the edges of each graph $G \in \{A, R\}$

$$W_{i,j}^G = \begin{cases} e^{-\left\|p^i - p^j\right\|^2} & i,j \text{ connected} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Then an extended cost function is introduced to combine information from the graphs

$$\arg\min_Q Q^T \cdot \left(L^A + L^R\right) \cdot Q \tag{4.6}$$

$$\text{subject to:} Q^T \cdot \left(D^A + D^R\right) \cdot Q = I \tag{4.7}$$

where $D^G = diag\left\{D_{11}^G, D_{22}^G, ..., D_{nn}^G\right\}$ is a diagonal matrix with $D_{ii}^G = \sum_{j=1}^n W_{ij}^G$, and $L^G = D^G - W^G$ is the Laplacian matrix. The minimum of the objective function can be found by the Lagrange function

$$\wedge\left(Q, \lambda\right) = Q^T \left(L^A + L^R\right) Q - \lambda\left(I - Q^T \left(D^A + D^R\right) Q\right) \tag{4.8}$$

$$\left(L^A + L^R\right) Q = \lambda \left(D^A + D^R\right) Q \tag{4.9}$$

The generalised eigenvalue problem is using to span the embedded space Q by the eigenvectors given by the d smallest nonzero eigenvalues λ.

Unlike the standard Laplacian Eigenmaps dimensionality reduction method (LE) that only preserves the manifold's local geometry [14], the temporal structure of the data manifold is preserved thanks to the inclusion of the graph R between time series. Consequently, TLE is able to preserve implicitly the local and global temporal topology of the data. This implies that TLE maintains the temporal continuity of time series during dimensionality reduction process and suppress stylistic variations displayed by different sources of time series by aligning them in the low dimensional space [54]. Experimental results also proof that as seen in Figure 5.7.

Although the manifold lies in the low-dimensional space, the observation function needs to be evaluated in the high-dimensional space. Consequently, a mapping function is required to find correspondences between the two spaces. Since spectral methods lack mapping functions to project data from one space to another, Radial Basis Function Network (RBFN), as suggested by [53], are trained to obtain these transformations φ and φ':

$$\varphi : \mathbb{R}^D \rightarrow \mathbb{R}^d \text{and } \varphi' : \mathbb{R}^d \rightarrow \mathbb{R}^D. \tag{4.10}$$

4.2.2 Application to Human Pose Modelling

We use Temporal Laplacian Eigenmaps (TLE) dimensionality reduction method to represent sequences of human poses for a given action. TLE generates a temporal representation of human postures, expressed as a single dimensional manifold, where style has been suppressed. TLE has been selected for the following reasons. Firstly, TLE explicitly preserves the temporal coherence of an activity, which is important for a tracking application. Secondly, TLE suppresses stylistic variation and is able to produce more compact manifolds (Figure 2.1). Searching in compact manifolds, such as those produced by TLE, is very efficient.

TLE is trained by a sequence of $3D$ poses of a training dataset P for a person performing a single action. A manifold Q is created in the low-dimensional space $\mathbb{R}^d$ as a result. Also, mapping functions φ and φ' between high and low-dimensional spaces are generated, as described in section 4.2.1.

4.3 Pose Tracking Framework

In this section we present a two-level $3D$ pose tracking approach namely Manifold Projection-Limb Correction (MPLC). In the first part of this method (MP), $3D$ human poses are constrained on a low-dimensional activity manifold by optimizing a full-body observation function. In the second part (LC) individual limb poses are refined by optimizing an observation function for each limb separately.

4.3.1 Manifold Projection

The first stage of MPLC is the Manifold Projection (MP) method, which is constrained to search poses that are similar to the training dataset. First the high-dimensional human pose of the previous frame (assuming that the initial pose is known) is projected on a point of the TLE low-dimensional space. Then, a deterministic optimisation method is applied to estimate a pose on the manifold.

Let assume the observation H^i of a person performing a single action A, derived as described in section 3.5, as input to the MP method. Also, let assume that the $3D$ pose of the previous frame p^{i-1} is known and the outputs of the action manifold learning part (section 4.2.2) Q, φ and φ' for a training dataset of a person doing the same action A are also given. The testing dataset is a sequence of frames where MP method is applied for every frame i.

Firstly, we estimate the global position and orientation g^i of the human

model exploiting the pose of the previous frame p^{i-1}. The observation H^i is compared with a human model hypothesis $M^i = \{g^i, p^{i-1}, m\}$ by maximising an observation function $s_1(M^i, H^i)$, defined later in section 4.4, varying the global position g^i. The values of the g^i depend on the type and the speed of the action.

$$\hat{g}^i = \arg\max_{g^i} s_1\left(\left\{g^i, p^{i-1}, m\right\}, H^i\right). \tag{4.11}$$

For the current frame i, the MP method consists of five steps (see Figure 4.3).

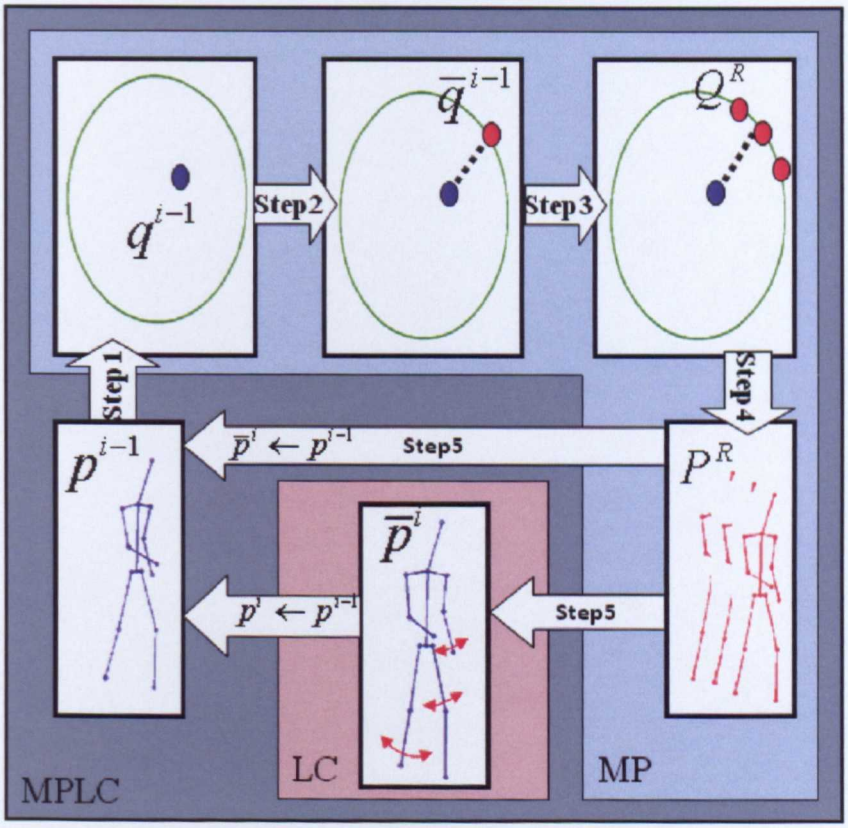


Figure 4.3: Flowchart of MP, LC and MPLC pipelines.

Step1: In order to move from the high-dimensional pose into the low-dimensional space we project the $3D$ pose to the low-dimensional space using the mapping function provided by the action manifold learning process i.e. the pose of the previous frame, p^{i-1} is projected to the low-

dimensional space $\mathbb{R}^d$ using the mapping function φ,

$$q^{i-1} = \varphi\left(p^{i-1}\right) \tag{4.12}$$

where q^{i-1} is the projection point into the low-dimensional space $\mathbb{R}^d$.

Step2: This point is projected on the manifold, in order to search for the optimal pose, using Euclidean distance i.e. the closest point $\bar{q}^{i-1}$ in the manifold Q to point q^{i-1}, is estimated.

At steps 3, 4 and 5 an optimization algorithm is applied to search for the point on the manifold that maximises the observation function.

Step3: Specifically, a sample of R points is selected from a neighbourhood of point $\bar{q}^{i-1}$ on the manifold Q:

$$Q^R = \{q^r, r = 1, \ldots, R\}, q^r \in \mathbb{R}^d \tag{4.13}$$

where the index r represents the temporal order of the points on the manifold.Step4: The selected points of the manifold are projected back to the high-dimensional space using the mapping function provided by the action manifold learning process i.e. all points of Q^R are back-projected to the high-dimensional space $\mathbb{R}^D$ using mapping function φ'. Let

$$P^R = \{p^r, r = 1, \ldots, R\} \tag{4.14}$$

be the set of candidate poses representations in $\mathbb{R}^D$, where

$$p^r = \varphi'\left(q^r\right), \forall r = 1, \ldots, R. \tag{4.15}$$

Step5: For the selected $3D$ poses the observation function is calculated using

the hypothesis of the $3D$ models and the observation of the input images i.e. every pose of P^R is compared with the observation of the input images H^i using the observation function s_1. The best pose $\bar{p}^i$ is chosen by maximising the function $s_1\left(P^R, H^i\right)$, i.e.

$$\bar{p}^i = \arg\max_{p^r} s_1\left(p^r, H^i\right), \forall p^r \in P^R \tag{4.16}$$

where $\bar{p}^i$ is the output $3D$ pose of the MP method for the frame i. This point corresponds to a point of the manifold Q that is projected on a $3D$ human pose.

The MP method can be visualised in the low-dimensional space. In Figure 4.4 we see the manifold that was created from the training dataset using TLE (highlighted in green), the ground truth data for the testing dataset that was projected into the low-dimensional space for 60 frames (red points), and the corresponding tracking points (blue points) generated by the MP method in the low-dimensional space $\mathbb{R}^2$. The positions of training and testing points differ because they originated from different subjects. The MP tracked points (blue) are on the manifold, because the MP method searches within the manifold. The MPLC tracked points are not usually on the manifold because the LC method moves the points out of the manifold if this leads to better accuracy.

4.3.2 Limb Correction (LC)

Since the manifold representation is constrained by the training data (section 4.2.1), there may be some discrepancy between the observed limbs and the manifold poses because of stylistic variations intrinsic to every subject. Therefore, the previous process needs to be refined to deal with this issue.

The second stage of MPLC applies Limb Correction (LC) for those limbs

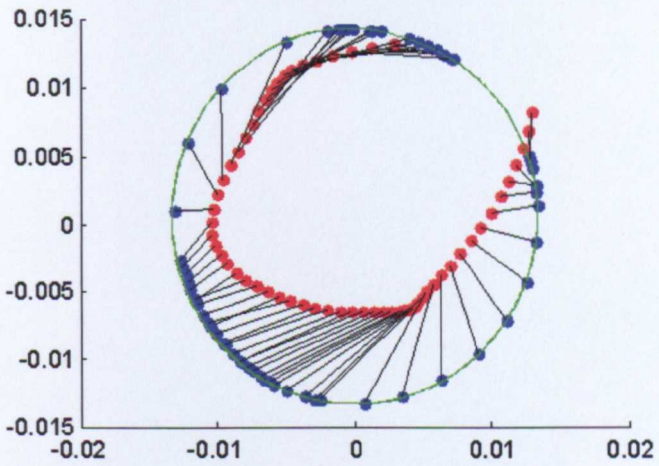


Figure 4.4: Low-dimensional space. Green: Manifold of the training data. Red: Ground truth. Blue: Tracking with MP method.

with significant error. The input of the LC stage is the $3D$ pose $\vec{p}^{\,i}$ from the equation 4.16. When a limb has been estimated, its evidence is removed from the observation. The evidence of the torso is removed in the beginning from the observation H^i to allow faster evaluation of the observation function but also to avoid errors in the estimation of the limbs that are near the torso. We apply the LC method for all limbs and head except the torso $\vec{p}^{\,i}(j), j = 2, \ldots L$.

For the current frame i, the LC method for each limb j consists of five steps, (Figure 4.5).

Step1: The hypothesis of the limb $\vec{p}^{\,i}(j)$ is compared to the observation using the observation function. If the $\vec{p}^{\,i}(j)$ derived through searching in P, is not satisfactory according to the threshold T, i.e.:

$$s_1\left(\vec{p}^{\,i}(j), H^i\right) < T \tag{4.17}$$

then we further search for the optimal solution in the high-dimensional limb pose space and proceed to Step2. Otherwise, the current limb estimate is considered to be sufficiently accurate and there is no further search.

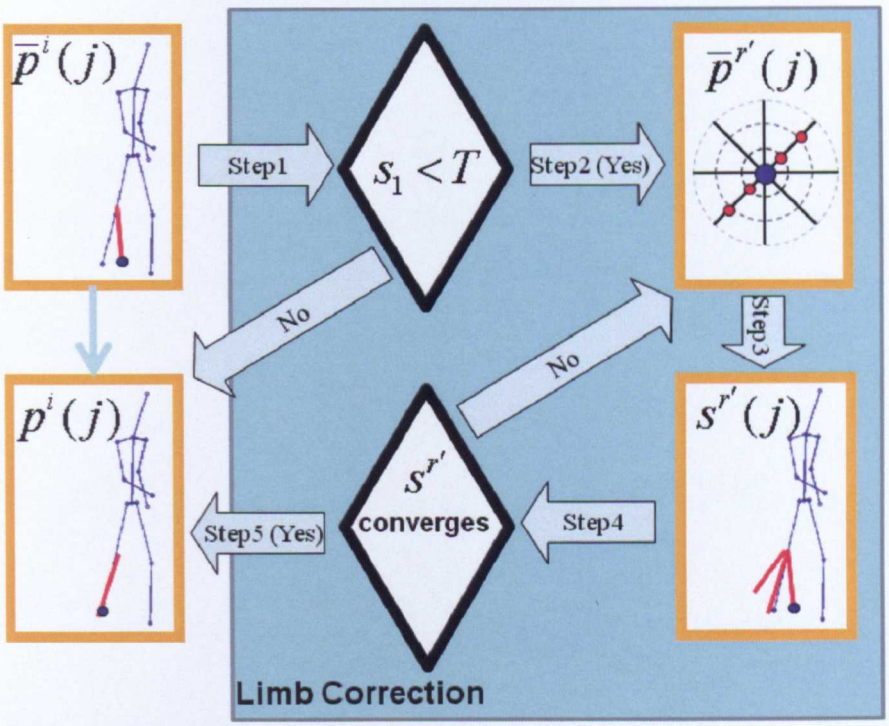


Figure 4.5: Limb error detection and correction pipeline.

Step2: Then a deterministic optimisation method is applied to detect the optimal position of the limb. We search for the r' rotation angle that optimises the observation function for the limb $\bar{p}^i(j)$. Since the solution space may be represented by the surface of a sphere, searching is performed on that surface using a gradient descent method. A point $\bar{p}^{r'}(j)$ is selected using a gradient-based optimisation algorithm.

Step3: The observation function of the limb pose $\bar{p}^{r'}(j)$ is estimated:

$$s^{r'}(j) = s_1\left(\bar{p}^{r'}(j), H^i\right) \tag{4.18}$$

Step4: The estimated pose is fed back to Step 3 until the observation function converges to a solution.

Step5: After maximising the function $s^{r'}(j)$ (Eq. 4.19) in steps 3 and 4, the final pose estimate $p^i(j)$ is the output of LC for limb j.

$$p^i(j) = \arg\max_{p^{r'}} s_1\left(\bar{p}^{r'}(j), H^i\right). \tag{4.19}$$

Before assessing the next limb, we remove the detected limb $p^i(j)$ from the observation H^i, and update the observation. The LC method is applied for all $L-1$ limbs and the final output is the $3D$ pose p^i. Thus, the estimated human model M^i is

$$M^i = \left\{m, g^i, p^i\right\} \tag{4.20}$$

where g^i is the global position and m is a known matrix representing $3D$ human model.

4.4 Observation Function

In this section the observation function that is used is presented. First we generate the observation from the input data and the human pose hypothesis. Then we define the observation function that is used to compare the observation and the pose hypothesis.

The MPLC method is evaluated for a testing set that comprises synchronised views of a human subject from multiple cameras as seen in Figure 4.6a. The observation H that we use is a $3D$ volumetric representation (visual hull) as described in section 3.5.2.1(Figure 4.6b). The human pose hypothesis M (Figure 4.6c) that is used, defined in section 3.4.

In order to evaluate a model hypothesis M, with the observed visual hull H we define an observation function s_1. We compare the volumes of the visual hull H and the human model M by using the relative overlap between them (Figure 4.6d). This observation function s_1 is defined by:

$$s_1(M, H) = \frac{\mid M \cap H \mid}{\mid M \mid}. \tag{4.21}$$

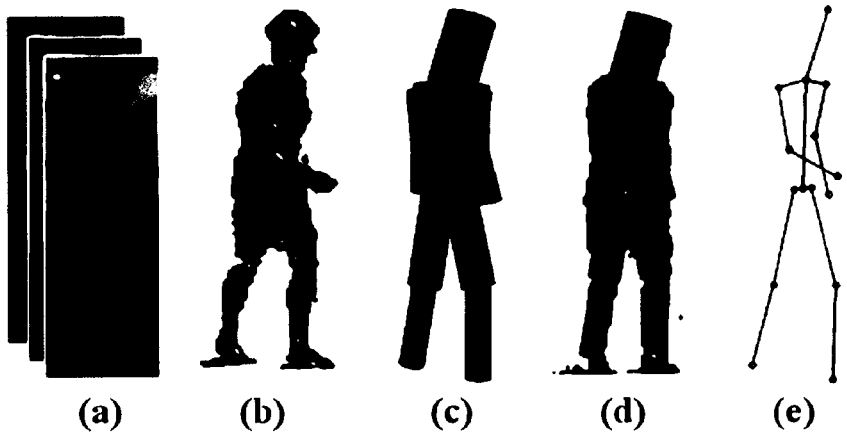


Figure 4.6: a) Images, b) computed Visual Hull, c) Human model, d) fitted human model to visual hull, e) extracted skeleton.

When the global position g and the size m are fixed we can use the term $s_1(p, H)$ where p is the pose of the model $M = \{g, p, m\}$ as described in section 3.4.

An advantage of the proposed observation function is that it allows comparisons of individual body parts of the human model to the visual hull as seen in Figure 4.7. Also, because of the $3D$ representation, individual body parts, like torso or arms, may be removed from the visual hull without affecting the observation of other body parts, making the search process more efficient. This contrasts with $2D$ image-based observation functions, such as the silhouette and edge likelihood and the bi-directional silhouette likelihood that are tested in [89] that do not allow comparison of individual body parts because of potential occlusions in image views.

4.5 Evaluation

In this section, results of the MPLC human pose estimation methodology are presented. We present the results of MPLC method and we compare it with an equivalent, i.e. of similar computational cost, particle filter approach (PF) [43] and other state-of-the-art methods [79, 77].

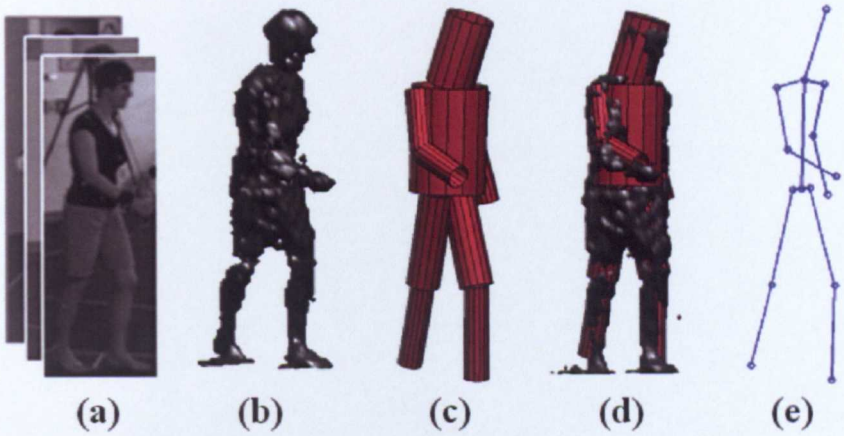


Figure 4.6: a) Images, b) computed Visual Hull, c) Human model, d) fitted human model to visual hull, e) extracted skeleton.

When the global position g and the size m are fixed we can use the term $s_1(p, H)$ where p is the pose of the model $M = \{g, p, m\}$ as described in section 3.4.

An advantage of the proposed observation function is that it allows comparisons of individual body parts of the human model to the visual hull as seen in Figure 4.7. Also, because of the $3D$ representation, individual body parts, like torso or arms, may be removed from the visual hull without affecting the observation of other body parts, making the search process more efficient. This contrasts with $2D$ image-based observation functions, such as the silhouette and edge likelihood and the bi-directional silhouette likelihood that are tested in [89] that do not allow comparison of individual body parts because of potential occlusions in image views.

4.5 Evaluation

In this section, results of the MPLC human pose estimation methodology are presented. We present the results of MPLC method and we compare it with an equivalent, i.e. of similar computational cost, particle filter approach (PF) [43] and other state-of-the-art methods [79, 77].

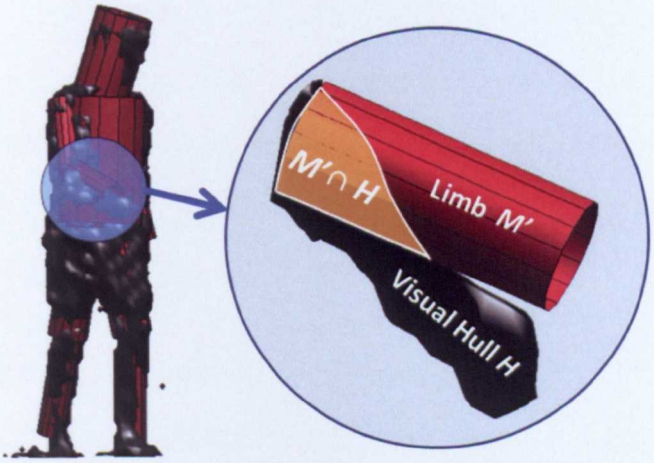


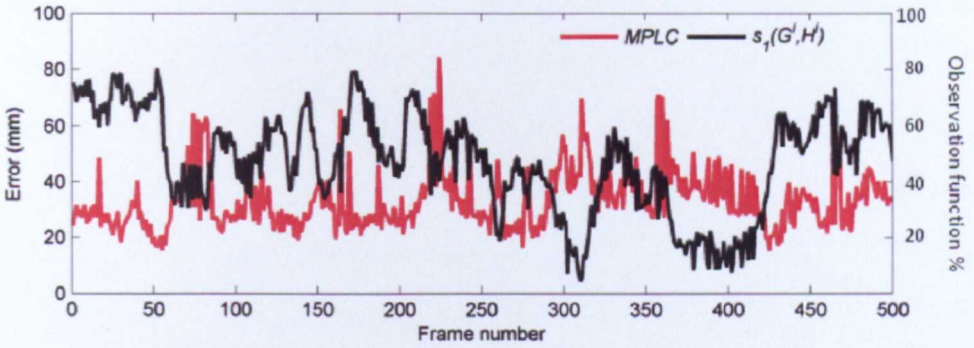
Figure 4.7: Calculation of observation function s_1 for individual body parts.

4.5.1 Datasets and Training

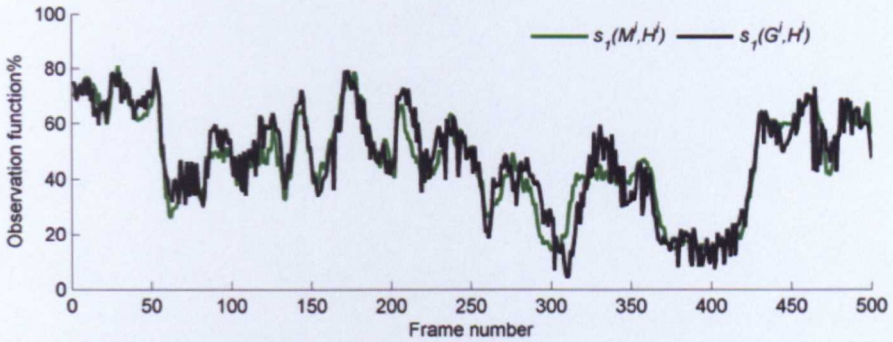
In order to compare MPLC method with other state-of-the-art methods we apply MPLC in public available datasets. The Image & MOCAP Synchronized Dataset (IMS) [18] and HumanEva (HE) Dataset [90] have been used for the experiments in this chapter. Our training set contains 1121 frames of the S3 walking sequence in trial 3 from HumanEva I. The IMS dataset (walking action), the HumanEvaI S1 and the HumanEvaII S2 and S4 (walking actions) are used for testing. In order to evaluate the pose tracking method (section 2.2), similar to [9, 89], we assume that the ground truth pose of the first frame is known for all experiments. In this work, the standard background subtraction method suggested by HumanEva [96] is used to ensure fair comparison with other methods.

4.5.2 Validation of Observation Function

In this section, we evaluate the observation function s_1. In order to calculate the s_1 we generate the volumetric model from the skeleton model, as seen in section 3.4. Figure 4.8a shows the inverse relationship between the average error per frame using MPLC configuration with threshold $T = 100\%$ (red line) and the values of the observation function s_1 for the ground truth poses G^i for every frame



(a)



(b)

Figure 4.8: (a) Error of MPLC and observation functions $s_1(G^i, H^i)$ per frame. (b) Observation functions $s_1(M^i, H^i)$ for our results and for ground truth $s_1(G^i, H^i)$ per frame.

values of parameter $R = \{4, 7, 10, \ldots, 24\}$ in equation 4.13 and T from 20% to 100% in equation 4.17.

MPLC is compared to the PF-TLE method that is also applied on the low-dimensionality space that was learnt by TLE. Performance with different numbers of particles $n = \{10, 15, \ldots, 50\}$ are conducted to ensure similar computational times with the MPLC method. For these experiments we used an Intel core 2 computer running Matlab implementations.

Figure 4.9 represents the average error for 100 frames for which the ground truth is known, as a function of the average computational time for each frame, for PF-TLE (blue line), MP (black line), and MPLC (red line) methods. As we can see MPLC is able to provide better results than MP and PF-TLE methods in all cases. By fixing the processing time we can obtain a direct com-

parison between MPLC and PF-TLE. For instance, if the average computational time of MPLC and PF-TLE methods is approximately $30sec$ per frame, the corresponding average error for PF-TLE is $44mm$ (standard deviation $\sigma = 12mm$) while MPLC's is $35mm$ (standard deviation $\sigma = 10mm$). The result justifies the LC part of the proposed method as the MPLC outperform MP in all cases. Also, PF-TLE has similar performance with MP, but reaches an accuracy limit around $45mm$ because of the TLE-constrained poses. MPLC overcomes this limit because LC is not constrained by TLE.

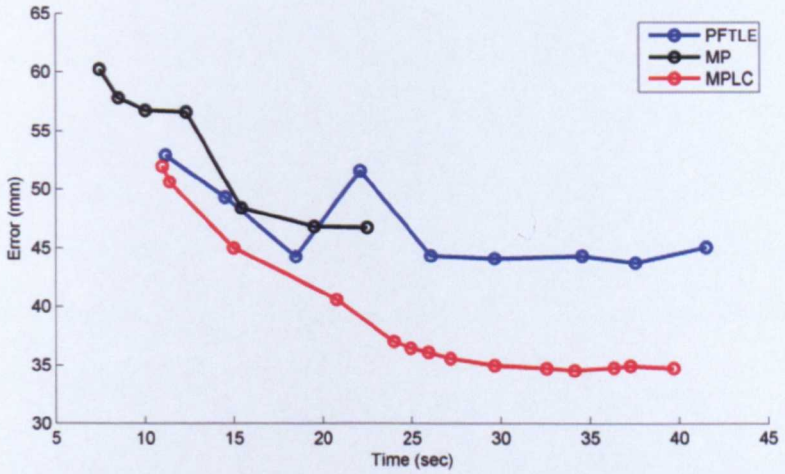


Figure 4.9: Comparison of average errors for 100 frames according to the average computational time for each frame for PF-TLE, MP and MPLC methods.

In the following experiments we use $R = 15$ in equation 4.13 and $T = 100\%$ in equation 4.17, while 35 particles are used for PF-TLE. Figure 4.10 displays the average error for every frame, as a function of the frame number, for the LC (green), MP (black), and MPLC (red) methods and for PF-TLE method (blue). The average computational time for MPLC and PF-TLE is approximately $30sec$ per frame and for MP and LC are approximately $15sec$ per frame and $20sec$ per frame respectively. Applying only LC, which is equivalent to searching the high-dimensional pose space, results to $72mm$ average error. Since LC is not constrained by TLE, tracking result diverges from the ground truth, i.e. the pose tracking error increases steadily over time, as seen in Figure 4.10.

On the other side, MP and PF-TLE avoid divergence issues because the TLE constraint and have similar performance: $48mm$ and $45mm$ average error respectively. Although PF-TLE spends twice the computational time as MP, performance improvement is minimal, which justifies the usage of deterministic gradient search instead of particle filter for searching in the low-dimensional space. Even if more particles are used for PF-TLE, no further improvement is expected, because of the difference the training dataset, represented by the TLE manifold and the testing dataset.

Such a restriction is overcome with MPLC that results in an average error of only $35mm$. The inclusion of the LC module leads to a significantly advantage regarding the accuracy with a relative small computational load increase. Therefore, MPLC combines the advantage of keeping pose estimates close to the TLE-manifold thanks to MP with the advantage of searching beyond the training dataset thanks to LC.

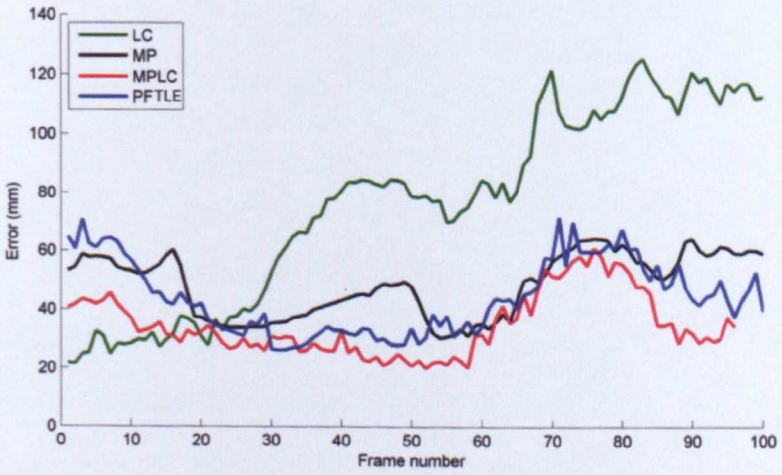


Figure 4.10: Average error per frame for 100 frames processed by methods MPLC, MP, LC and PF-TLE.

In Figure 4.11 we can see visual results of skeleton models generated by MPLC method (blue poses) and the corresponding ground truth poses (red poses).

For the last experiment we calculate the global position of the human

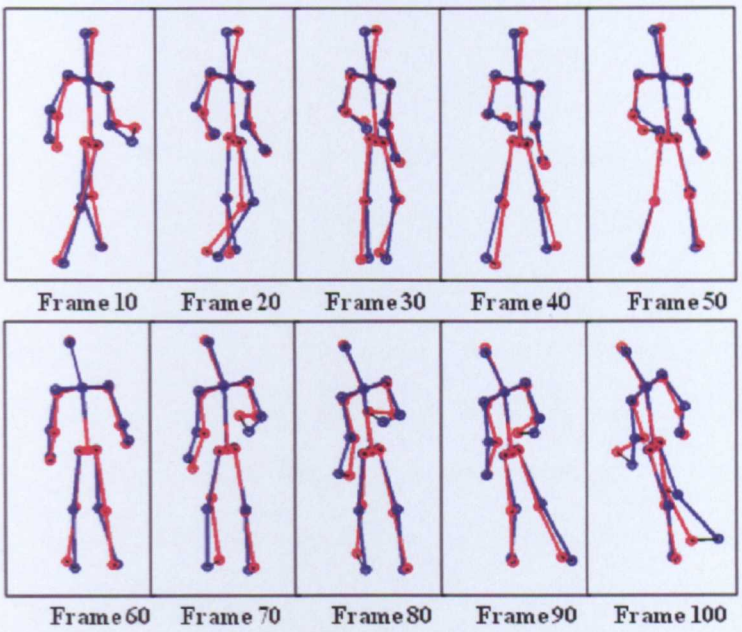


Figure 4.11: Skeleton models for Red: ground truth and for Blue: our method (MPLC15)

	HEIIS2walk	HEIIS4walk	HEIS1walk	Comp.
GPAPF	86.6	89.0	86.3	500
H-APF	75.2	81.8	75.4	500
MP	74.0	96.2	72.0	10
MPLC	71.4	75.6	68.8	60

Table 4.1: Average error in mm for GPAPF, H-APF, MP and MPLC methods.

body at every frame as described in section 4.3.1. At Table 4.1 we compare our method with the state-of-the-art methods GPAPF [79] and H-APF [77]. The GPAPF uses APF for searching in a low-dimensional space generated by the GPLVM dimensionality reduction method as described in section 2.2.2.2 and the H-APF is a hierarchical extension of GPAPF as described in section 2.2.3. For MPLC and MP results, we use $R = 15$ in equation 4.13 and $T = 100\%$ in equation 4.17, while 500 particles were used for the particle filter approaches. We can see that in all cases MPLC outperforms GPAPF and H-APF, although it performs only 12% of observation function evaluations.

4.6 Discussion

In this chapter we presented a novel human pose tracking methodology called MPLC. The MPLC method has two stages: MP and LC. In the MP stage, the observation pose is compared with the model hypothesis constrained by a low-dimensional manifold to avoid divergence of pose tracking. The manifold is trained by the TLE dimensionality reduction method using a sequence of poses of the training dataset. The MP method searches for the best match between the observation and the training points using a deterministic optimisation method, instead of particle filter methods, to provide efficiently an initial pose estimate. The LC stage deals with the problem of stylistic variations of human activity by refining each limb individually. The LC method is able to search for the optimal position of the body parts that have been erroneously determined during the MP method.

This chapter demonstrates that the MPLC method provides better accuracy than particle filter approaches. Although particle filter methods are popular techniques for human tracking, they are computationally expensive because of the large number of particles that they require [89]. In our experiments, we applied PF in the low-dimensionality space, which was learnt by TLE. Although PF-TLE can achieve satisfactory accuracy, MPLC's accuracy is even better for the same processing time. Also MPLC clearly outperforms particle filter methods that were applied in GPLVM-generated manifolds, despite the generalisation properties of GPLVM.

The bottleneck of our implementation is the evaluation of the observation function, which leads to high-computational times. Real-time performance may be achievable if an optimised version of the observation function, programmed in C/C++, is deployed in appropriate hardware. However, compared to other generative methods, MPLC has lower complexity, i.e. fewer evaluations per frame

and lower overall computational cost. We can conclude that the combination of MP and LC provides significant advantages in terms of accuracy, stability and computational cost.

Chapter 5

Human Pose Tracking by Hierarchical Manifold Searching using Hierarchical Temporal Laplacian Eigenmaps

5.1 Introduction

In this chapter we introduce Hierarchical Temporal Laplacian Eigenmaps (HTLE), a novel hierarchical dimensionality reduction method, and Hierarchical Manifold Search (HMS), a human pose tracking methodology. Both HTLE and HMS fit in the general pipeline that was presented in section 3.1 (Figure 3.1), and, in particular, in the pose tracking and the dimensionality reduction processes respectively.

The TLE dimensionality reduction method, which was used in the previous chapter, represents only poses that appear in the training data. In order to expand this space into its components, we introduce HTLE, a hierarchical dimensionality reduction method. The HTLE approach allows us to search in each level of a posture hierarchy separately, thus modeling new, unseen poses.

Furthermore, HMS searches for optimal poses through the hierarchical structure of HTLE. HMS-HTLE performs better than MPLC, as discussed theoretically in section 5.2.2 and confirmed by experimental results in section 5.5.5. Parts of this work have been published in [69].

5.1.1 Overview

The framework that is presented in this chapter operates on a two-phase approach: first, a sequence of poses from a training set are used in order to generate a hierarchy of low-dimensional manifolds using HTLE and, second, pose tracking is performed in a hierarchical manner using HMS. The pipelines of both phases are presented in Figure 5.1. More specifically, the training dataset consists of a sequence of poses (typically MOCAP data) describing the action of interest which is given. Hierarchies of action manifolds are learned by the proposed Hierarchical Temporal Laplacian Eigenmaps (HTLE) (Figure 5.1a), as described in section 5.2. We propose to use TLE [54] as the base of our hierarchy, because it suppresses stylistic variation and, therefore, generates more compact manifolds in comparison to other methods (Isomap [98], BC-GPLVM [49], LE [13], ST-Isomap [44], GPDM [108]), as discussed in section 2.2.3.

The pose tracking process is constrained by the hierarchy of action manifolds (Figure 5.1b), which is presented in section 5.3. In every cycle, an observation from the input data is estimated. The observation and the previously learnt action manifolds are fed to our novel search method, i.e. Hierarchical Manifold Search (HMS) (section 5.3), which efficiently explores the pose space described by HTLE. We minimise computational costs by using a deterministic optimisation method, instead of searching the whole hierarchy using particle filtering approaches [24, 77]. The final output is a sequence of poses.

An observation function is introduced to match the observation from

multiple colour cameras to pose hypotheses (section 5.4). The performance of the proposed framework is evaluated for a range of parameters and compared to state-of-the-art human tracking methods using publicly available datasets, in section 5.5.

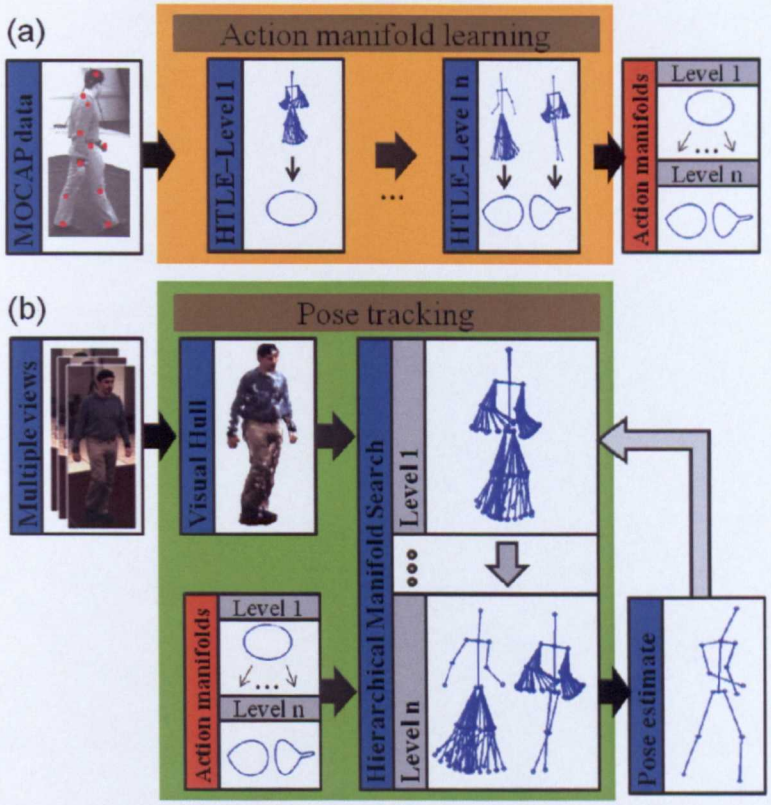


Figure 5.1: (a) Training and (b) pose tracking pipelines.

5.2 Action Manifold Learning

5.2.1 Hierarchical Temporal Laplacian Eigenmaps (HTLE)

In this section, we present the formation of Hierarchical Temporal Laplacian Eigenmaps (HTLE). TLE manifolds (Section 4.2.1) only represent poses seen in the training dataset, therefore TLE-constrained solutions may be biased because of stylistic variations between the training and testing datasets. In order to deal with this restriction we propose to expand the available pose space using HTLE,

a hierarchy extension of TLE. The advantages of such a structure are two-fold: firstly, fast searching is facilitated by a set of compact TLE manifolds, as we have already discussed in chapter 4; secondly, the hierarchy of manifolds models unseen poses to address the problem of stylistic variations between the training and the testing datasets. More specifically, the hierarchical structure of HTLE has been designed to allow searching each level of the hierarchy extending overall pose search range. This is achieved by exploring each level separately and then combining all of them, generating a new, unseen configuration.

HTLE uses a training dataset P to generate a hierarchy of manifolds in low-dimensional spaces. Let $P_{h,l}$ be the set of N poses of the training dataset that corresponds to the l-th pose subspace at the hierarchical level h

$$P_{h,l} = \left\{ p^i_{h,l}, i = 1, ..., N \right\}, \tag{5.1}$$

where $p^i_{h,l} \in \mathbb{R}^{D_{h,l}}$ is the pose of the model at the time i. As discussed in Section 4.2.1 TLE produces a manifold $Q_{h,l}$ representing $P_{h,l}$ in a low-dimensional space $\mathbb{R}^{d_{h,l}}$

$$Q_{h,l} = \left\{ q^i_{h,l}, i = 1, ..., N \right\}, \tag{5.2}$$

where $q^i_{h,l} \in \mathbb{R}^{d_{h,l}}$ and $d_{h,l} \ll D_{h,l}$.

At a given level h (Figure 5.2), mapping between the high- and low-dimensional spaces [54] is performed by the functions:

$$\varphi_{h,l} : \mathbb{R}^{D_{h,l}} \rightarrow \mathbb{R}^{d_{h,l}}, \varphi'_{h,l} : \mathbb{R}^{d_{h,l}} \rightarrow \mathbb{R}^{D_{h,l}} \tag{5.3}$$

where

$$\varphi_{h,l}\left(p^i_{h,l}\right) = q^i_{h,l}, \varphi'_{h,l}\left(q^i_{h,l}\right) = p^i_{h,l}. \tag{5.4}$$

We also define mapping functions (Figure 5.2) between the hierarchical

level points $p_{h-1,l} \in P_{h-1,l}, p_{h,l'} \in P_{h,l'}$

$$\omega_{h,l'} : P_{h-1,l} \rightarrow P_{h,l'}, \text{where } \omega_{h,l'}(p_{h-1,l}) = p_{h,l'} \tag{5.5}$$

These mapping functions permit evaluating hypotheses by projection to the high-dimensional space as well as propagating hypotheses through the hierarchy.

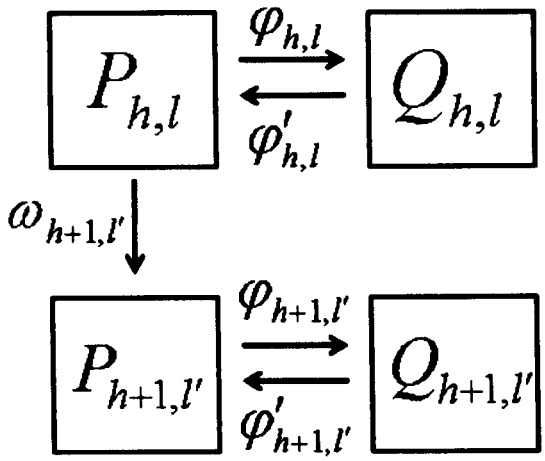


Figure 5.2: Pose subspaces P and submanifolds Q connected by mapping functions $\varphi, \varphi'$and ω.

5.2.2 Application to Human Pose Modelling

We define a hierarchy based on the division of the individual body parts as shown in Figure 5.3 and Figure 5.7. At the first level, h_1, the whole body is represented. At the next level, h_2, the variability of the previous level is expressed by two subspaces containing either the upper or the lower body. The division process is repeated for the next two levels, h_3 and h_4: firstly, four subspaces are created to model the four individual limbs, i.e. left and right arms and legs; secondly, each limb is divided into two segments, i.e. upper and lower arm and leg, to produce in total eight submanifolds. At the last level, h_5, each limb segment is allowed to move in an unconstrained manner similar to section 4.3.2. The levels

h_4 and h_5 have the same leaf nodes but the searching space is different in each of them. Nonetheless, we include it in the hierarchy for simpler representation of the pose tracking method (section 5.3). By introducing different levels with an increasing level of specificity, we incrementally vary the ability of generating new pose hypotheses while maintaining a certain level of constraints.

The HMS method improves the results of the human pose tracking problem when compared with the MPLC method. Generally, results depend on how close the global optimal solution is to the initial pose, because gradient-based optimisation may be trapped in a local optimal. Fortunately, in both MPLC and HMS, this effect is reduced due to searching through multiple levels. Since HMS searches through more levels than MPLC, improved accuracy is expected. Therefore, when reaching the LC level, HMS provides a better initialisation for the LC process than does the MP method.

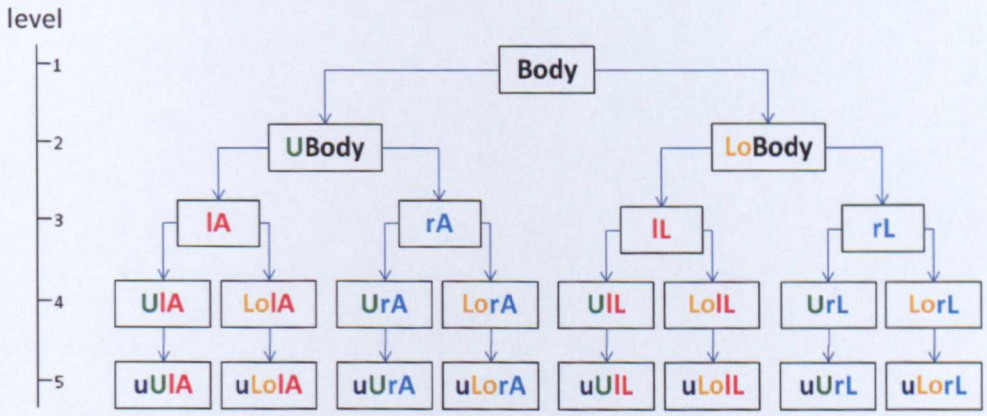


Figure 5.3: Five-level hierarchy of human model. Each level is represented horizontally in the figure. Level number increases by one progressively from top to bottom. Every level h is composed of pose subspaces l. U: Upper, Lo: Lower, l: left, r: right, A: Arm, L: Leg, u: unconstrained

5.3 Pose Tracking Framework-HMS

In this section, we introduce the Hierarchical Manifold Search (HMS) method, which is used to estimate the human pose through the hierarchy proposed in

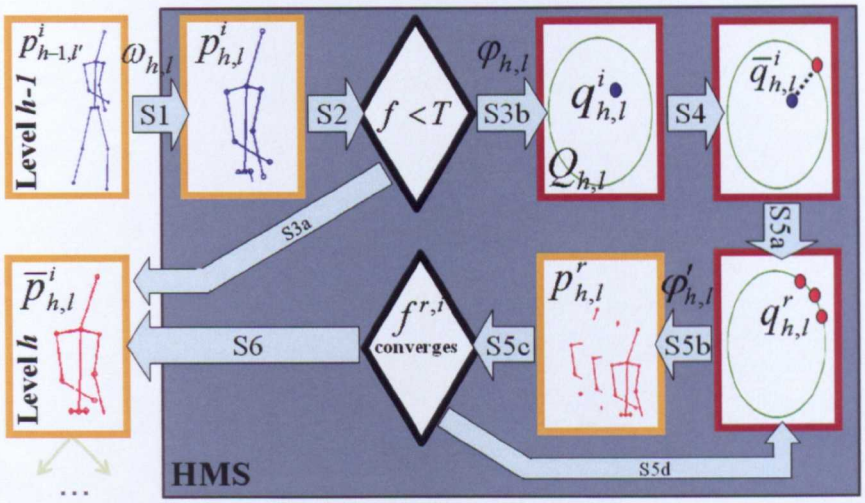


Figure 5.4: Flowchart of HMS at subspace (h, l) of the hierarchy. Transformations in the high- and low-dimensional spaces are represented in orange-framed and red-framed boxes, respectively.

section 5.2. Initially, we search the top level of the hierarchy, which represents the full body pose. Then, we search the rest levels of the hierarchy, each of them representing a different division of the human body. This procedure allows as to take full advantage of the hierarchy of manifolds which mitigates discrepancies between the testing and training dataset by permitting the estimation of unseen poses.

For every frame i, we optimise the observation function $f\left(\{g^i, p^i, m\}, H^i\right)$ in two steps. Firstly, we initialise the global position and orientation g^i of the human model with the previous frame p^{i-1}. The new global position $\hat{g}^i$ is estimated as described in section 4.3.1 and the corresponding body model is the $\{\hat{g}^i, p^{i-1}, m\}$. During this step the torso is removed from the observation H^i to allow faster evaluation of the observation function, and also to avoid errors in the estimation of the limbs that are near the torso.

Secondly, the pose p^i of the current frame i is estimated. Specifically, a process is applied through the hierarchy, as illustrated in Figure 5.4. We apply the following algorithm to each TLE manifold l, for each TLE-constrained level h.

Initially, a new hypothesis $p^i_{h,l}$ for frame i is generated (Figure 5.4, S1). If $h = 1$, the pose from the previous frame is projected to the pose subspace $P_{1,1}$, i.e.

$$p^i_{1,1} = p^{i-1} \tag{5.6}$$

otherwise if $h > 1$ the point from the pose subspace l', from the previous hierarchical level $h - 1$ is projected to the child pose subspace $P_{h,l}$ using the function $\omega_{h,l}$ (Eq.5.5) to restrict the part of the human model that is searched:

$$p^i_{h,l} = \omega_{h,l}\left(p^i_{h-1,l'}\right) \tag{5.7}$$

Then, the model hypothesis is compared to the observation using the observation function (Figure 5.4, S2) (Eq.5.17).

If the match between the hypothesis and the observation is sufficiently large (Figure 5.4, S3a), i.e.

$$f\left(p^i_{h,l}, H^i\right) > T, \tag{5.8}$$

where T is linked to the required accuracy, searching the current subspace (h, l) is omitted. Therefore, the final estimation for this subspace is given as: $\hat{p}^i_{h,l} = p^i_{h,l}$ and HMS proceeds with the following manifolds (S1).

Otherwise, the high-dimensional point $p^i_{h,l}$ is projected to the low-dimension space $\mathbb{R}^{d_{h,l}}$ to find a more accurate estimate (Figure 5.4, S3b):

$$q^i_{h,l} = \varphi_{h,l}\left(p^i_{h,l}\right). \tag{5.9}$$

Then, the solution is constrained using the action manifold. Specifically, HMS considers the closest point $\hat{q}^i_{h,l}$ to the point $q^i_{h,l}$ in $Q_{h,l}$ (Figure 5.4, S4).

Afterwards, the local maximum is searched by optimising the observation

function. A gradient descent optimisation algorithm is used in order to find a local maximum where putative solutions are evaluated in the high-dimensional space using the observation function. More specifically, this is achieved by following the four following sub-steps (Figure 5.4, S5a-S5d).

A point $q^r_{h,l} \in Q_{h,l}$ is selected using a gradient-based optimisation algorithm.

The point $q^r_{h,l}$ is back-projected to the high-dimensional space $\mathbb{R}^{D_{h,l}}$ of human poses. Let $p^r_{h,l}$ be the point after the projection

$$p^r_{h,l} = \varphi'_{h,l}\left(q^r_{h,l}\right). \tag{5.10}$$

The observation function of the point $p^r_{h,l}$ is estimated:

$$f^{r,i}_{h,l} = f\left(p^r_{h,l}, H^i\right). \tag{5.11}$$

The estimated pose is fed back to the algorithm (Figure 5.4, S5a) until the observation function converges to a solution. Finally, the output of the algorithm is the optimal point $\hat{p}^i_{h,l}$ that maximises the observation function $f^{r,i}_{h,l}$ (Figure 5.4, S6)

$$\hat{p}^i_{h,l} = \left\{p^r_{h,l} : \max_r f^{r,i}_{h,l}\right\}. \tag{5.12}$$

At the last level h' of the hierarchy, Limb Correction may be applied to refine the solution in an unconstrained space as described in section 4.3.2.

The output of this process is the $\hat{p}^i_{h',l}$ pose for every subspace l of the level h' of the hierarchy.

Finally, the pose of the model p^i is estimated by concatenating the body

parts estimated at the last level of the hierarchy

$$p^i = \left\{\hat{p}^i_{h',1}, \hat{p}^i_{h',2}, ..., \hat{p}^i_{h',n}\right\}. \tag{5.13}$$

Thus, the estimated human model M^i is

$$M^i = \left\{m, g^i, p^i\right\} \tag{5.14}$$

where g^i is the global position and m is a known matrix (section 3.4) from the initial frame.

HMS allows a data-driven efficient search of the hierarchy of manifolds, compared to previous hierarchical approaches, [77, 24]. The threshold T controls this search, i.e. the lower the threshold, the less accuracy is needed, and the faster the search will be performed, as demonstrated in section 5.5. Although our approach is based on gradient-descent optimisation, the hierarchy structure minimises the problem of being trapped into a local optimum, by searching again limb configurations at different levels, as shown in the results presented later.

5.4 Observation Function

In order to evaluate the HMS method we use a testing set comprises synchronised views of a human from multiple colour cameras. First, a $3D$ volumetric representation (visual hull) of the observed human is generated to allow evaluation of human model hypotheses as described in section 3.5.2.1. The colour from the input images is also back-projected on the visual hull in order to discriminate between body parts and improve accuracy as described in section 3.5.2.2. The final observation is the coloured visual hull H (Figure 5.5). The human pose hypothesis M (Figure 4.6c) that is used, is defined in section 3.4.

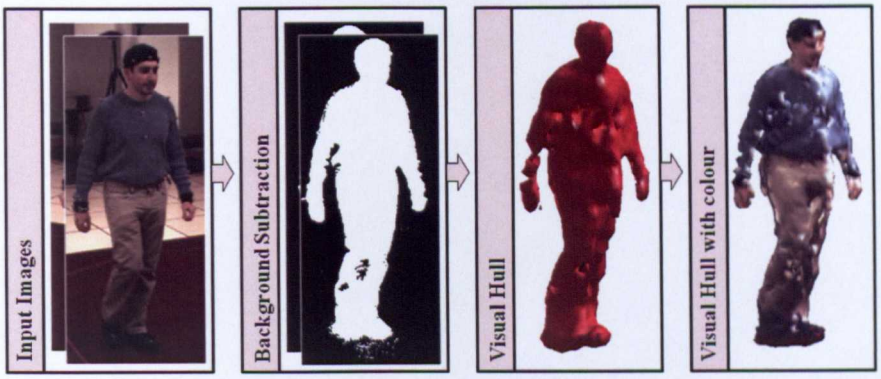


Figure 5.5: The pre-processing pipeline. From left to right: the input images, the corresponding silhouettes, the visual hull and the visual hull with colour.

The proposed observation function takes into account two features of the observation: volume and colour. Firstly, we compare the volumes of the visual hull H and the human model M by using the relative overlap between them. This part of the observation function has already defined in section 4.4 (s_1 in Eq. 4.21).

The second part of the observation function exploits the colour information of the visual hull. This is important since it complements the first part of the observation function especially for poses where the limbs are close to the torso as the colour of the torso is normally different than this on the limbs. At the initial pose the colour of the limbs $c_j^1, j = 1, ..., L$ is estimated, using the voxels of the initial visual hull H^1, matched by the limb j. Then, this colour information is used for comparing the colour of the corresponding areas of frame i with the initial one. Specifically, $c_j^1, j = 1, ..., L$ is estimated as the average of the hue values of all the matched voxels, assuming an HSV colour space. The hue value of HSV colour space is used for comparing the colour without affecting the saturation and the brightness in every frame. The Figure 5.6 illustrates the HSV colour space. Then, at the frame i the colour information of the visual hull H^i of every voxel v, $c_j^{i,v}, j = 1, ..., L$, matched by the limb j, is compared to the initial limb colour c_j^1. A binary colour similarity variable, $C_j^{i,v}$, is introduced to emphasise significant colour differences and at the same time suppress noise in

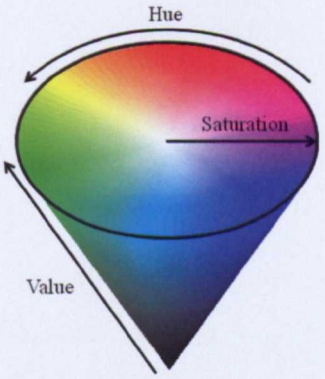


Figure 5.6: The HSV colour space. Hue, Saturation and Value are illustrated in the figure.

the hue channel

$$C_j^{i,v} = \begin{cases} 1, & \text{if } \left|c_j^{i,v} - c_j^1\right| \leq a \\ 0, & \text{if } \left|c_j^{i,v} - c_j^1\right| > a \end{cases} \tag{5.15}$$

where $a \in [0, 1]$ is an appropriate threshold. Then the observation function s_2 is defined by:

$$s_2(M, H) = \frac{1}{L} \sum_{j=1}^{L} \frac{\sum_{v=1}^{V_j} C_j^{i,v}}{V_j} \tag{5.16}$$

where V_j is the total size in voxels of each area j and L is number of the body parts.

The observation function of the model M and the coloured visual hull H is given by the weighted mean

$$f(M, H) = \sum_{k=1}^{2} w_k s_k(M, H) \tag{5.17}$$

where w_k is a weight that allows us to change the balance between observation functions, where $\sum_{k=1}^{2} w_k = 1$.

The proposed observation function allows comparisons of individual body parts of the human model to the visual hull. This property is important when moving down through our hierarchy in section 5.2.

5.5 Evaluation

5.5.1 Overview

In this section we analyse the parameters and depict the results of the HMS method. Firstly, the publicly available datasets that are used are presented. Then the training process is discussed. After that, the observation function is evaluated. Then, the HMS method is tested for different accuracy thresholds and different hierarchy levels. Those parameters are analysed in order to calculate the trade-off between computational cost and accuracy. Finally, HMS is tested using a variety of datasets and compared with state-of-the-art human pose tracking methods.

5.5.2 Datasets and Training

In order to facilitate the comparison of HMS with other methods, we apply HMS to 4 standard walking sequences: HumanEva (HE) II-S2 (frames 1 to 390), HEII-S4 (frames 4 to 370), Image & MOCAP Synchronized Dataset (IMS) (frames 1 to 150) and HEI-S1walking (frames 1 to 590) [18, 89] and 2 jogging sequences: HEII-S2 (frames 391 to 710), HE-II S4 (frames 371 to 710). For all sequences, we used human actions captured by 4 cameras and calibration information for each of them. For the walking sequences the tracker is initialised by the first pose using ground truth and for the jogging the tracker is initialised by the last estimated pose of the corresponding walking sequence. Coloured Visual hulls are created using the calibration data and the silhouettes provided with the datasets.

A training dataset is used to generate the HTLE models as discussed in section 5.2. Walking and jogging HTLE models are estimated using 1443 skeleton poses from the HEI-S2 walking, trial-3 and 795 skeleton poses from the HEI-S2

jogging, trial-3 sequences respectively. The same training dataset is used for all experiments for each action to demonstrate the generalisation properties of the HMS method. In Figure 5.7 human poses that correspond to the training data set $P_{h,l} \in \mathbb{R}^D$ and the corresponding manifolds in $2D$, $Q_{h,l} \in \mathbb{R}^2$ are shown for different levels of the hierarchy h and pose subspace l.

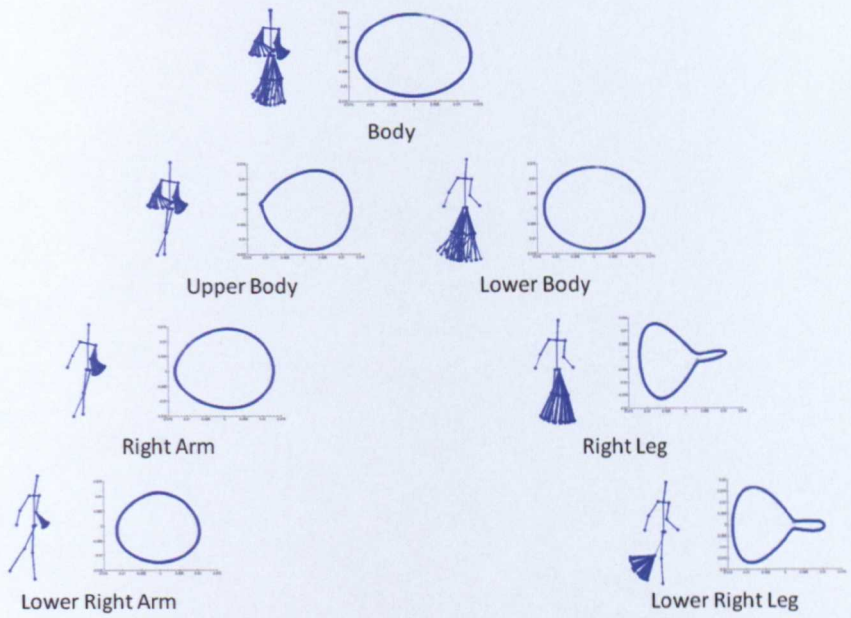


Figure 5.7: Different levels of the hierarchy. Human poses and the corresponding manifolds are represented in $2D$ for a walking activity.

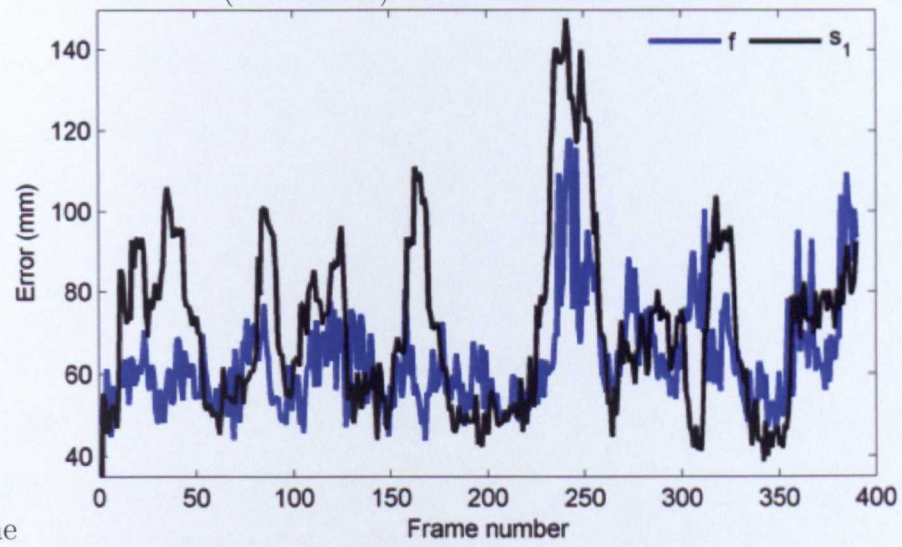
5.5.3 Validation of Observation Function

In this section, we evaluate the observation function relative to the error of our methodology results. Note that in these experiments, only colour of the end-limbs (lower arms and lower legs) is used for the observation function s_2 to take advantage of colour discrimination of hands/shoes. In all experiments we use a threshold $a = 0.2$ in equation 5.15 and $w_1 = w_2 = 0.5$ in equation 5.17.

The observation function s_1 is compared to the observation function $f = (s_1+s_2)/2$. In Figure 5.8 the error per frame using the observation function f with colour information (Eq. 5.17) is presented in blue (average error $63.1mm$), and

without using colour information s_1 (Eq. 4.21) in black (average error $70.2mm$), for HEII S2 dataset. We can see that colour information improves results in most frames.

(section 3.4) from the initial



frame

Figure 5.8: Error per frame of HMS(1,2,3,4,5) using observation function f with colour (blue) and observation function s_1 without colour (black), for HEII S2 dataset.

5.5.4 HMS Configuration

In this section, we investigate different configurations of the HMS method by evaluating different sets of levels in the hierarchy and different values of the threshold T. We denote as $HMS(h_1, h_2, h_3 \ldots)$ the HMS method applied for levels $h_1, h_2, h_3, \ldots$ as seen in Figure 5.3.

Figure 5.9 shows the average error and the computational time per frame for 150 frames of the IMS dataset for different HMS configurations. As shown in Figure 5.9(a) by increasing the levels of the hierarchy, the estimated error decreases for every threshold. Furthermore, by increasing the threshold the error decreases in all configurations. Likewise, as shown in Figure 5.9(b), the computational cost (mean number of observations per frame for all frames) rises with increasing levels of hierarchy. That is because of the increase in the number of

subspaces that are searched in every level, as seen in Figure 5.3. Finally, computational cost rises for increasing thresholds. Figure 5.9(c), shows the mean number of observation evaluations per frame for different levels of the hierarchy and different thresholds in the HMS(1,2,3,4,5) configuration. The mean number of observations per frame for every level increases for rising thresholds. Therefore, different configurations of HMS provide flexibility on compromising between computational cost and accuracy, demonstrating the value of the hierarchy.

In this experiment, after the 80% threshold, error and computational cost are almost constant. That is because the maximum value of the observation function is near to 80%, as shown in Figure 4.8a. For these experiments we used an Intel core 2 laptop with code written in Matlab. The computational costs vary from $4sec$ to $55sec$ per frame.

Table 5.1 also shows the usage of different levels of the hierarchy for different thresholds in the HMS(1,2,3,4,5) configuration. For small thresholds, the contribution of the first level to the final solution is dominant and that keeps the computational cost low. Higher accuracy is achieved by increasing the contribution of the lower levels of the hierarchy. The contribution of the last level (unconstrained limb poses) is relatively high even for small thresholds, since the subject (and therefore the style) in training and testing data are significantly different.

5.5.5 Comparison with the State-of-the-Art

In order to compare the HMS method with state-of-the-art methodologies we apply HMS to the Walking action of HEII-S2 (frames 1 to 390) and HEII-S4 (frames 4 to 297), IMS (frames 1 to 150) and HEI-S1walking1 (frames 1 to 590) and to the Jogging action of HEII-S2 (frames 391 to 710) and HE-II S4 (frames 371 to 790). For every action we use the corresponding training dataset as discussed

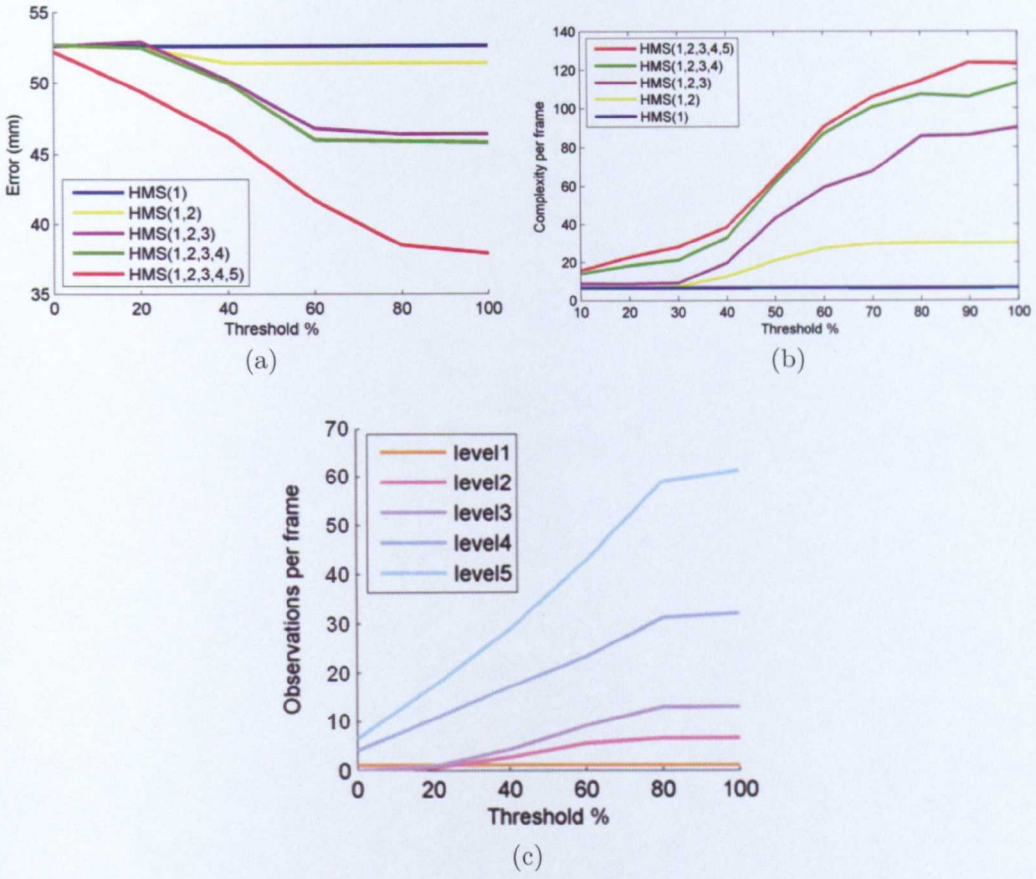


Figure 5.9: HMS performance for different thresholds and configurations (different numbers of hierarchy levels). (a) Average error of different configurations of HMS for 150 frames and different thresholds ($0 - 100$ and (b) average number of evaluations of the observation function per frame for HMS method for increasing thresholds ($0 - 100\%$). (c) Mean number of observation evaluations per frame for different levels of the hierarchy and different thresholds in the HMS(1,2,3,4,5) configuration.

in section 5.5.2. For all sequences, 4 cameras are used and the ground truth for the first frame initialises the tracker. Since ground truth is not known for the full length of the sequences, the results of the HMS method were evaluated using the online evaluation system of the Human Eva website [17]. Our method is quantitatively evaluated against the MP (Manifold Projection) method, MPLC (Manifold Projection Limb Correction) method presented in the previous chapter 4, APF [9] that demonstrates state-of-the-art performance according to [89] and applications of APF in low-dimensional spaces, i.e. GPAPF [79], H-APF [77].

In Table 5.2 we present the average absolute $3D$ error [89], for GPAPF

Th.%	Level1	Level2	Level3	Level4	Level5	Time(s/f)	Error(mm)
10	100	0	0	8	8	8	44
20	100	0	0	17	16	14	43
30	100	1	0	25	23	18	41
40	100	19	7	31	28	24	40
50	100	64	40	41	37	32	38
60	100	92	77	58	53	38	34
70	100	100	92	79	73	42	35
80	100	100	99	95	90	45	34
90	100	100	100	100	99	48	33
100	100	100	100	100	100	49	33

Table 5.1: Search (%) of the hierarchy in every level for HMS(1,2,3,4,5) method.

	HEIIS2walk	HEIIS4walk	HEIS1walk	Comp.
GPAPF	86.6	89.0	86.3	500
H-APF*	75.2	81.8	75.4	500
MP	74.0	96.2	72.0	10
MPLC	71.4	75.6	68.8	60
HMS	63.1	62.5	65.0	130

Table 5.2: Average error in mm for GPAPF, H-APF, MP, MPLC and HMS methods (*the H-APF results are the average of whole sequence).

and MPLC, and the corresponding hierarchical methods, i.e. H-APF and HMS. We also present the complexity (mean number of observations per frame) for every method. In this experiment, a threshold $T = 100\%$ (Eq.5.8) is set for HMS to achieve optimal results. These results demonstrate the value of introducing hierarchy in dimensionality reduction based approaches, as hierarchical methods always perform better than the original ones, and improves computational efficiency and accuracy compared with GPAFP and H-APF. Our decision to base our dimensionality reduction framework on TLE is confirmed by the comparison between TLE-based and GPLVM-based representations. Specifically, MP and HMS outperform in most of the cases, GPAPF and H-APF, respectively.

In Figure 5.10 we show the average error per frame for HEI-S2 walking and HEI-S4 walking datasets for MP (blue line) and HMS(1,2,3,4,5) (red line) methods using threshold $T = 100\%$. HMS(1,2,3,4,5) clearly improves MP in all datasets (see Table 5.2). This demonstrates the value of using the hierarchy.

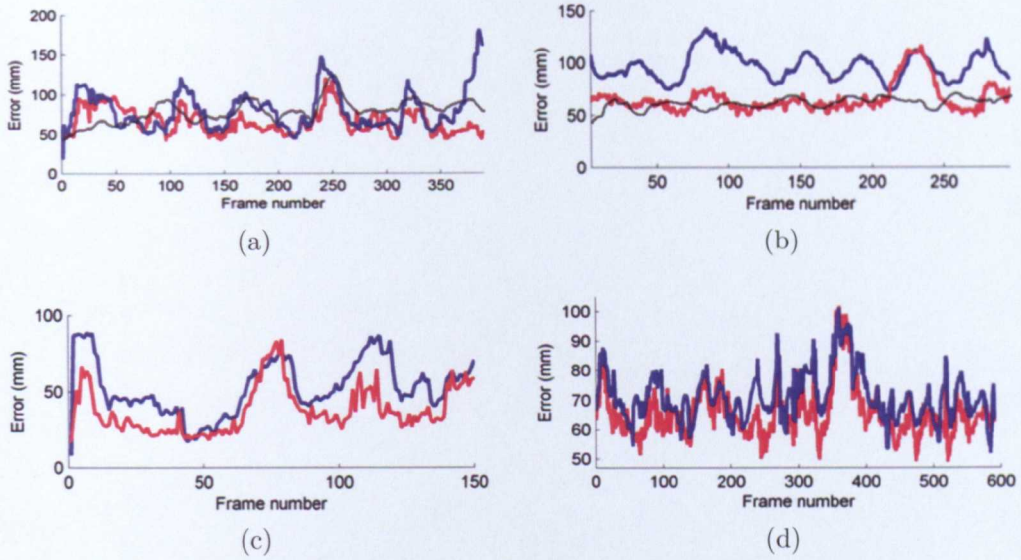


Figure 5.10: Results for (a) HEII-S2, (b) HEII-S4, (c) IMS and (d) HEI-S1walking1 sequence with MP (blue line), HMS(1,2,3,4,5) (red line) and APF (black line) methods when it is available.

Figure 5.11 and Table 5.3 displays the average absolute $3D$ error for APF and HMS using different particle numbers and thresholds respectively, and their computational costs per frame as measured on the same machine using Matlab implementations for both methodologies. Their level of complexity 'Comp.', i.e. the number of evaluations of their observation function, is also shown in Table 5.3. HMS using $T = 100\%$ generally outperforms APF both in terms of error and complexity. Moreover, the figure suggests that HMS is able to deliver similar accuracy to any APF configuration using only $5\% - 25\%$ of processing time. The low complexity of our method comes from the hierarchical searching strategy that is driven by the observation function. Furthermore, the combination of a hierarchical approach with a search that occurs beyond the training dataset results in improved accuracy. In summary, HMS methodology achieves the best overall accuracy with the lowest computational complexity.

In Figure 5.12 we show the average error per frame for HE-II S2 walking and jogging dataset using HMS(1,2,3,4,5) for lower body (red line), upper body (blue line) and full body (black line). The training dataset that is used is the

	HEIIS2walk	HEIIS4walk	HEIIS2jog	HEIIS4jog	IMS	Comp.
APF1000	76	60	85	93	41	1000
APF500	83	63	109	154	46	500
APF250	88	70	133	180	49	250
HMS100%	63.1	62.5	80.9	102.4	37.6	128
HMS60%	65.1	64.1	82.5	104.2	41.7	104
HMS40%	69.5	65.2	86.1	106.8	46.3	78
HMS 20%	74.2	67.5	87.3	107.5	49.5	57

Table 5.3: Average error in mm and complexity (number of evaluations) for different configurations of APF and HMS.

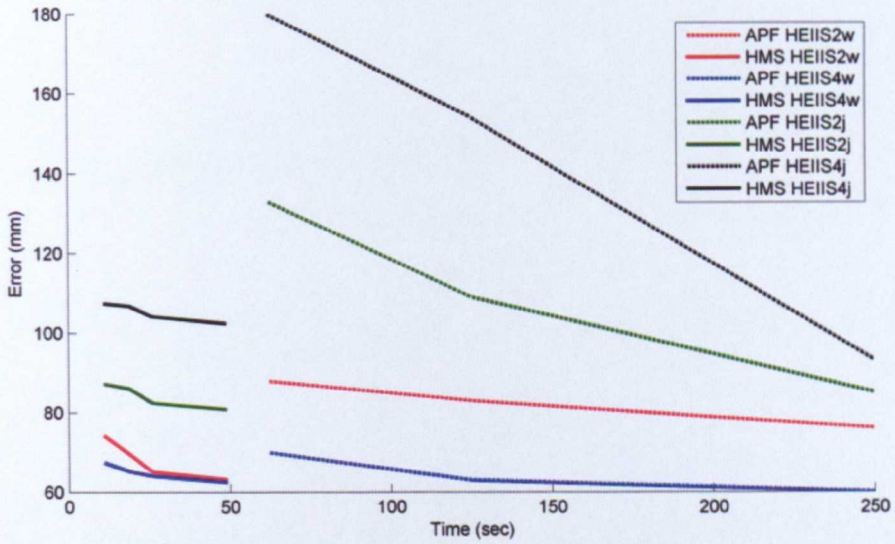


Figure 5.11: Average error in mm and computational cost per frame in seconds for different configurations of APF and HMS.

walking action as described in section 5.5.2. The average error for the lower body is $63.7mm$ ($57mm$ for walking and $70.4mm$ for jogging), for the upper body is $96.2mm$ ($69.2mm$ for walking and $123.2mm$ for jogging) and for the full body is $79.6mm$ ($63.1mm$ for walking and $96.7mm$ for jogging). The error in the walking sequence (frames $1 - 390$) is lower than that of the jogging action (frames $390 - 710$), since training was based on walking data. More specifically, error in the jogging action is higher mainly because of upper body error: in the tested jogging activity, arm positions are significantly dissimilar to those found in the walking dataset, especially when arms are near to the torso. Since the latter configuration is periodical over the jogging action, a cyclic pattern of error is observed in Figure 5.12. On the other hand, although a walking action was

used for training, leg positions were estimated accurately for both walking and jogging activities. These results suggest that our methodology is able to track different styles efficiently to the extend that these are not significantly dissimilar to the training set, so they can still be considered as a variation of the same given activity.

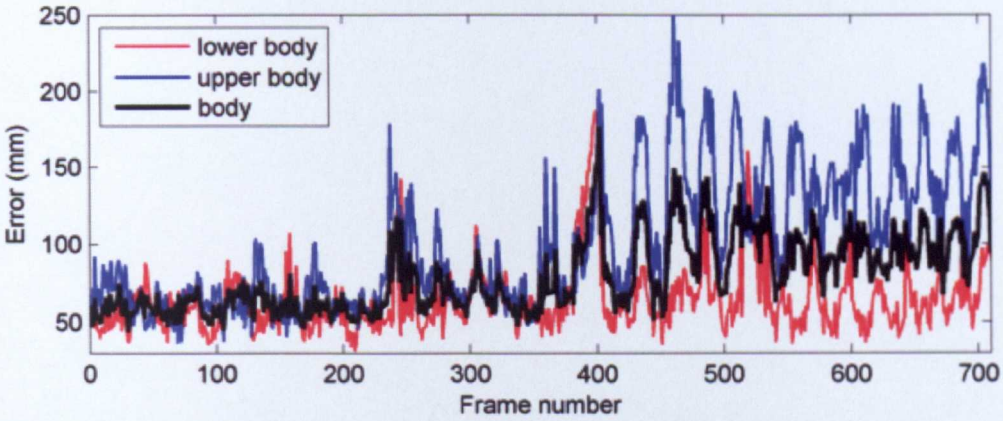


Figure 5.12: Average error per frame for HEII-S2 dataset with HMS(1,2,3,4,5) for lower body (red line) and upper body (blue line) and full body (black line).

In Figure 5.13 we show the average error per frame for HE-II S2 walking dataset, using the HMS(1,2,3,4,5) method with threshold 100% for individual limbs. As we can see in the graph, the shoulder has a lower error than the elbow or the wrist, and in the lower body the knee has more accurate results than the ankle, as we would expect. Also, similarly, we observe that the leaf nodes of the hierarchy for the lower body have better results than the leaf nodes in the upper body. That makes sense as hands are generally more challenging to track as they can be more easily confused with the torso, and they are less constrained by the specific activities (walking, jogging).

In the Figures 5.14, 5.15, 5.16,5.17 and 5.18 we display tracking results of HMS(1,2,3,4,5) with threshold 100% for the datasets used in the study. For IMS dataset only the first part of the observation function s_1 is applied because imagery is grey-scale. APF and HMS(1,2,3,4,5) result in $41mm$ and $37.6mm$ average error respectively.

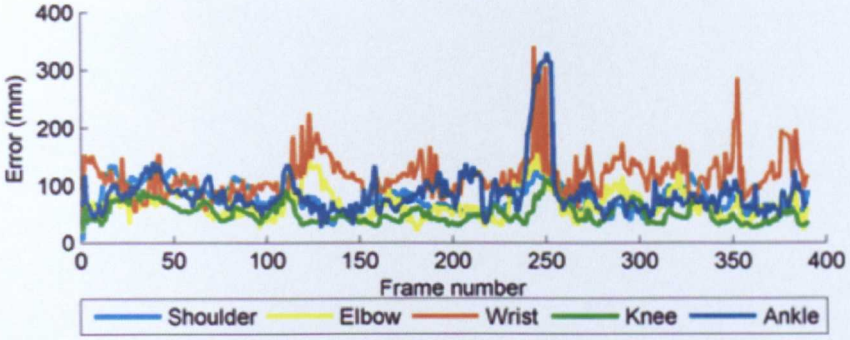


Figure 5.13: Error for selected individual joint locations. Average error per frame for HEII-S2 sequence and HMS(1,2,3,4,5) method for different body parts for 390 frames.

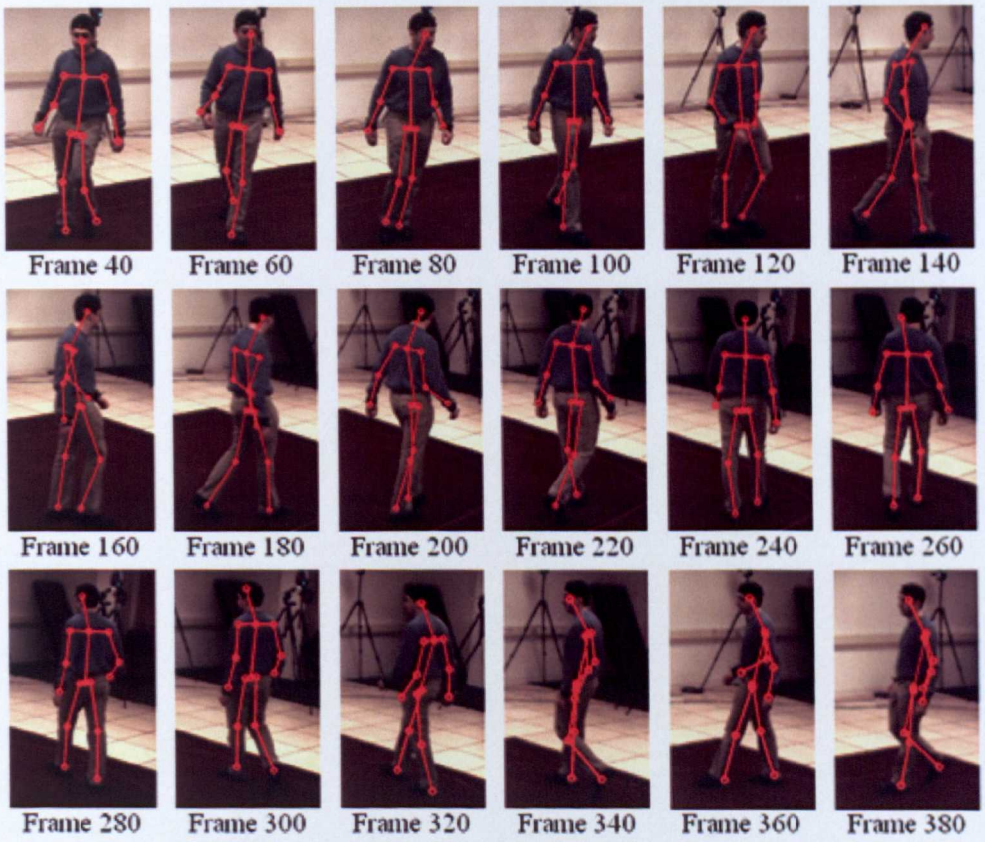


Figure 5.14: Results for HEII-S2 walking dataset with HMS(1,2,3,4,5).

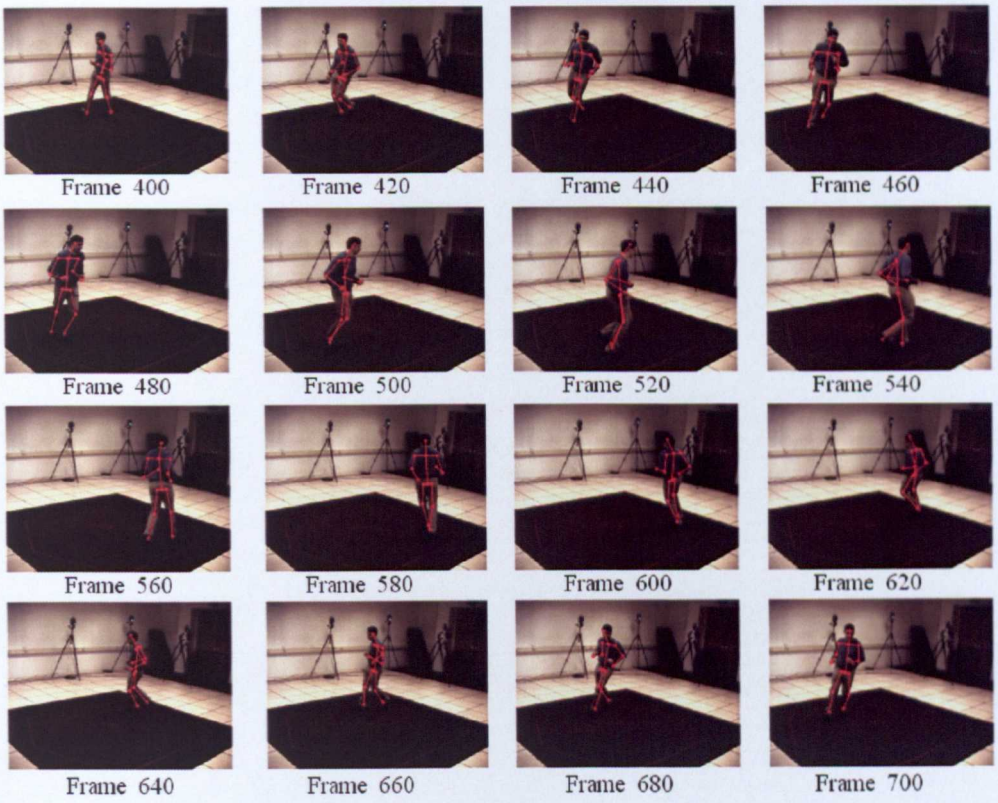


Figure 5.15: Results for HEII-S2 jogging dataset with HMS(1,2,3,4,5).

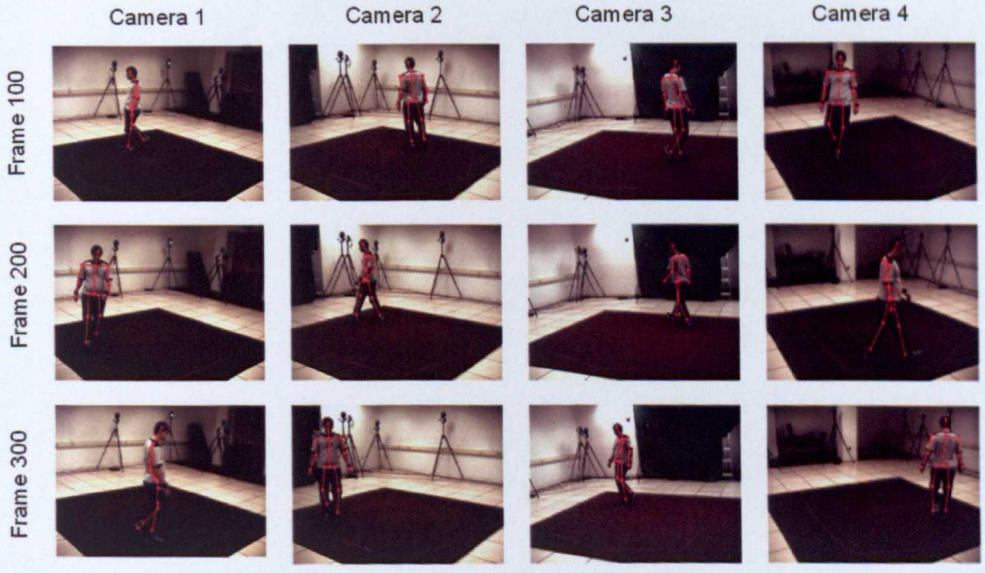


Figure 5.16: Results for HEII-S4 dataset with HMS(1,2,3,4,5) for four cameras.

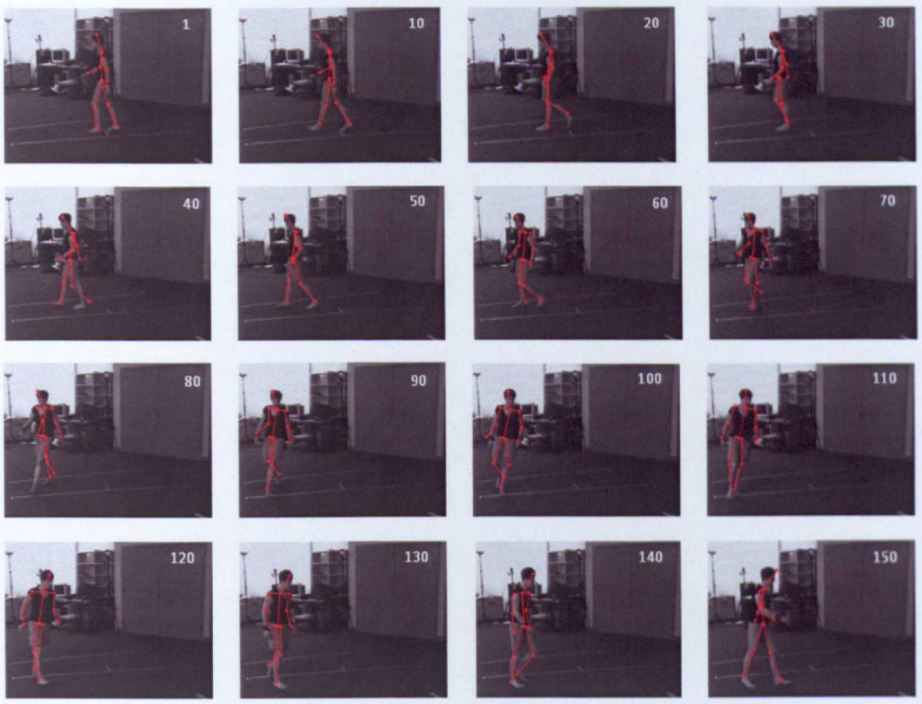


Figure 5.17: Results for IMS dataset with HMS(1,2,3,4,5).

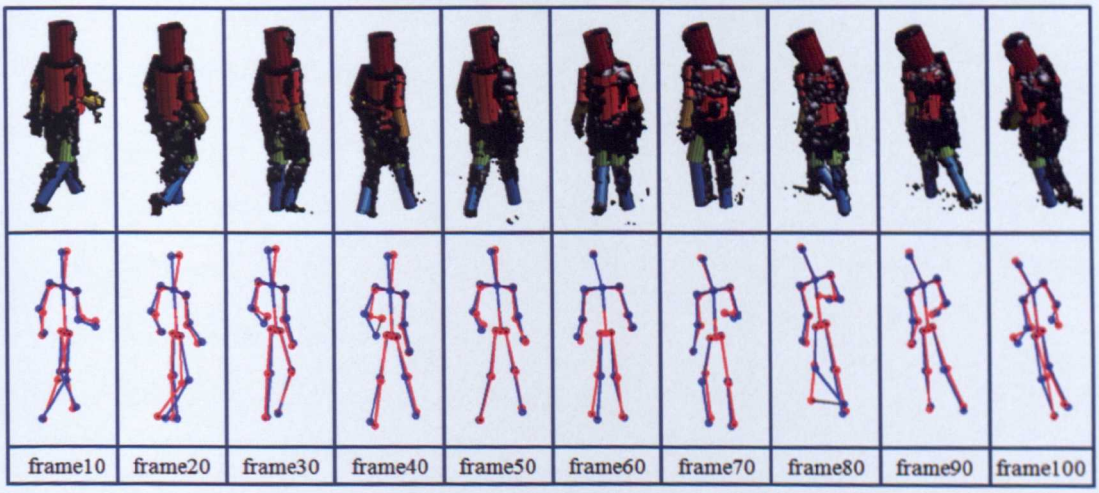


Figure 5.18: Results for IMS dataset with HMS(1,2,3,4,5).

5.6 Discussion

This chapter presented a human pose tracking methodology relying on two novel techniques. Firstly, a hierarchical method based on dimensionality reduction for human pose tracking was proposed. TLE is used as the basis for our hierarchy

as it suppresses stylistic variation and generates more compact manifolds in comparison to other methods such as ST-Isomap [44] and BC-GPLVM [49] (section 2.2.3). HTLE has been designed for human pose tracking as it takes into account the hierarchical representation of the human body. This allows the decoupling from the structure of the training dataset, and the exploration of unseen poses.

Secondly, we introduce a method, HMS, which deterministically searches through the hierarchy of low-dimensional manifolds and is driven by an observation function. HMS allows searching in a constrained space at every level of the hierarchy, so it requires a low number of evaluations of the observation functions and, therefore, low-computational resources. In addition, by searching through the hierarchy we are able to consider a wide range of unseen poses. Therefore, unlike conventional dimensionality reduction methods, which are restricted to the set of poses present in a training set, our framework is capable of moving beyond the training set and generating pose hypotheses that have never been seen before. Compared to MPLC, which can also search beyond the TLE-constrained manifold, HMS-HTLE demonstrates better performance, as explained theoretically and confirmed experimentally. In addition, instead of searching the whole hierarchy, as performed in previous studies using particle filtering [24, 77], we minimise computational costs by controlling this process using a deterministic optimisation method driven by the observation function which aims at fast convergence.

Experimental results were presented on publicly available datasets and comparisons to state-of-the-art methods were given. They demonstrate the accuracy and efficiency of our approach compared to other state-of-the-art methods. However, HMS, as presented here, may only be applied in single-activity scenarios, not in multi-activity scenarios.

Chapter 6

Human Pose Tracking for Multi-Activity Scenarios

6.1 Introduction

The pose tracking method that was presented in the previous chapter can only be applied in single-action scenarios. In this chapter, we introduce Hierarchical Manifold Search - Multi Activity (HMS-MA), a novel $3D$ human pose tracking methodology for multi-activity scenarios. With reference to the general pipeline (Figure 3.1) presented in section 3.1, the HMS-MA method corresponds to the pose tracking process, while multiple hierarchies of manifolds are estimated by HTLE.

HMS-MA properly extends the HTLE and HMS techniques that were presented in chapter 5, so they can be applied in multi-activity scenarios. First, using HTLE we generate a hierarchy of manifolds for each action relevant to a scenario. Then, we recognise the type of action for every frame using the HMS-MA method. Finally, the HMS-MA is applied to the whole hierarchy of the recognised action to estimate the pose. The validation of our method uses publicly available datasets captured by either a multi-camera system or a depth

camera (i.e. Microsoft Kinect), and demonstrates its accuracy and computational efficiency.

6.1.1 Overview

In this section the pipeline of the Hierarchical Manifold Search - Multi Activity (HMS-MA) pose tracking method is presented. HMS-MA relies on a training phase, which generates a model for each action of interest, and an online phase where $3D$ postures are recovered for each frame of a sequence representing an individual performing a variety of actions. Hierarchies of manifolds for the actions of a specific scenario are learnt using HTLE (Figure 6.1 (a)), as described in section 5.2. Each hierarchy represents a single action in the low-dimensional space, as described in section 5.2.2.

The HMS-MA method is applied to an unseen sequence of observations for a person performing multiple actions on a given scenario (Figure 6.1 (b)). In order to evaluate the observation at every frame, a $3D$ human body model is used to generate pose hypotheses. First, online action classification is performed based on the whole body manifold of the hierarchies. Then, the pose estimation is refined by searching through the hierarchy of the recognised action. The outcome of HMS-MA is an action classification label and a $3D$ pose estimate for each frame.

HMS-MA allows searching in a constrained space at every level of the hierarchy, so it requires a low number of evaluations of the observation functions and, therefore, low-computational resources. In addition, by searching through the hierarchy we are able to consider a wide range of unseen poses. Therefore, unlike conventional dimensionality reduction methods that are restricted to the set of poses present in a training set [104, 31, 56], our framework is capable of moving beyond the training set and generating pose hypotheses that have never

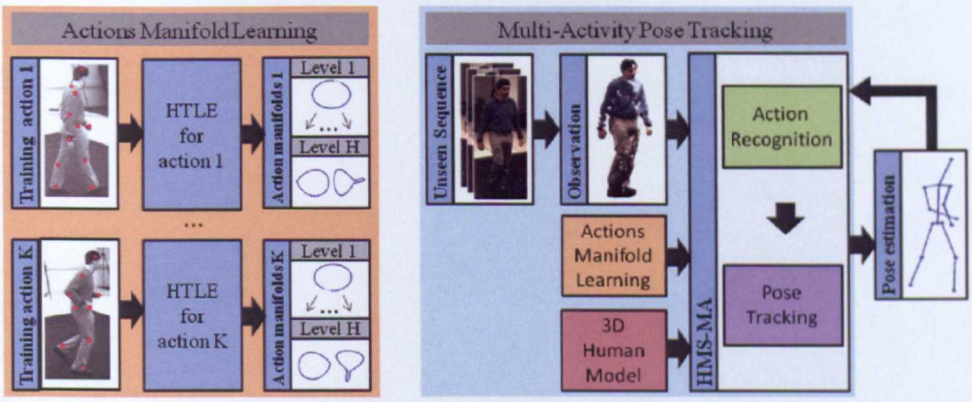


Figure 6.1: (a) Actions manifold learning and (b) human pose tracking pipelines for multi-activity scenario.

been seen before.

An interesting requirement of pose tracking in multi-activity scenarios is how stylistic variations are addressed. Specifically, they must be suppressed in action recognition [54, 52], but they must be expressed in pose hypotheses for efficient pose tracking [58]. Therefore, we use only the first level of the hierarchy, where style has been suppressed for recognising the current action, and all levels to generate a variety of hypotheses for tracking the pose of an individual accurately.

6.2 Action Manifold Learning

In this section we discuss the generation of a set of hierarchies of manifolds for K different actions for use in a multi-activity scenario tracking. For this purpose, we exploit in a multi activity context the Hierarchical Temporal Laplacian Eigenmaps presented in section 5.2.

We define a hierarchy based on the division of the individual body parts, similarly to [77, 24]. At the first level, h_1, the whole body is represented. At the next level, h_2, the variability of the previous level is expressed by two subspaces

containing either the upper or the lower body. The division process is repeated for the next two levels, h_3 and h_4: firstly, four subspaces are created to model the four individual limbs, i.e. left and right arms and legs; secondly, each limb is divided into two segments, i.e. upper and lower arm and leg, to produce in total eight submanifolds. Differently from previous work though, at the last level, h_5, each limb segment is allowed to move in an unconstrained manner. By introducing different levels with an increasing level of specificity, we incrementally vary the ability of generating new pose hypotheses while maintaining a certain level of constraints.

We assume that the given training dataset consists of K sets of sequences that correspond to K actions, each of them consisting of N_k poses in the high-dimensional space $\mathbb{R}^{D_k}$. Let $P_{h,l,k}$ be the concatenation of the sequences of the k-th action in the training dataset, represented in the high-dimensional space $\mathbb{R}^{D_{h,l,k}}$ that corresponds to the l-th pose subspace at the hierarchical level h

$$P_{h,l,k} = \left\{p^i_{h,l,k}, i = 1, ..., N_k\right\}, \tag{6.1}$$

where $p^i_{h,l,k} \in \mathbb{R}^{D_{h,l,k}}$ is the pose of the model at the time i. The HTLE method is applied to all $P_{h,l,k}$, for every level of the hierarchy h, subspace l and action k. As described in section 5.2 the resulting manifold is:

$$Q_{h,l,k} = \left\{q^i_{h,l,k}, i = 1, ..., N_k\right\}, \tag{6.2}$$

where $q^i_{h,l,k} \in \mathbb{R}^{d_{h,l,k}}$ and $d_{h,l,k} \ll D_{h,l,k}$.

Radial Basis Function Networks (RBFN) [53] are used to define mapping functions $\{\varphi_{h,l,k}, \varphi'_{h,l,k}\}$ between the high and low-dimensional spaces

$$\varphi_{h,l}\left(p^i_{h,l}\right) = q^i_{h,l}, \varphi'_{h,l}\left(q^i_{h,l}\right) = p^i_{h,l}. \tag{6.3}$$

We also define the mapping function between hierarchical levels

$$\omega_{h,l',k}\left(p_{h-1,l,k}\right) = p_{h,l',k}. \tag{6.4}$$

The result of the action manifold learning process is a set of K hierarchies of manifolds corresponding to the K action types. These mapping functions permit evaluating hypotheses by projection to the high-dimensional space as well as propagating hypotheses through the hierarchy.

6.3 Pose Tracking Framework - HMS-MA

In this section the Hierarchical Manifold Search - Multi Activity (HMS-MA) method for $3D$ pose tracking in a multi-activity scenario is presented. This method is an extension of HMS that was presented in section 5.3. For every frame of the testing dataset the HMS-MA method recognises the type of action that the person performs. Then the HMS method is applied using the hierarchy of manifolds of the selected action. The result is the $3D$ pose of the current frame.

6.3.1 Action Classification

In this section the first part of Hierarchical Manifold Search - Multi Activity (HMS-MA) method for online action classification is presented. We assume a set of K hierarchies of manifolds corresponding to the K types of actions has been generated using HTLE, as described in section 6.2. The unseen sequence represents a single subject performing a subset of the K actions. For every frame i we firstly estimate the global position and orientation of the human model by optimising the observation function applied on the previous pose, as described in section 4.3.1. The action recognition method consists of three steps as seen in

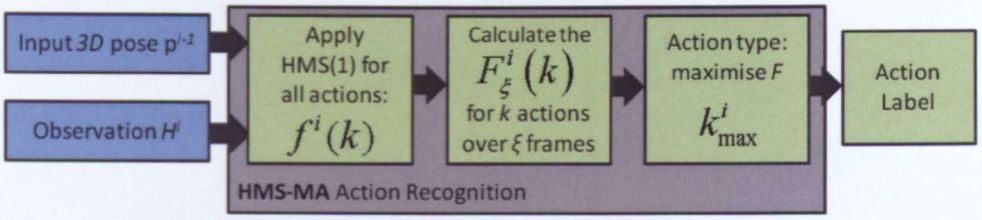


Figure 6.2: HMS-MA action recognition pipeline for multi-activity scenario.

Figure 6.2:

Step1: First, in order to recognise the type of action of frame i, we estimate the $3D$ pose according to the model of each of the possible activities by applying the HMS (chapter 5) model-based pose estimation algorithm to the whole-body manifolds of all K hierarchies for pose p^{i-1}. A $3D$ pose $p^i(k)$ is estimated for all $k = 1, \ldots, K$. For every pose $p^i(k)$ a corresponding observation function $f^i(k)$ is calculated (as defined in section 6.4).

Step2: Then, in order to exploit the information of previous frames we calculate the average value of the last ξ observation functions for each action k:

$$F_\xi^i(k) = \sum_{j=1}^{min(\xi,i)} \frac{f^{i+1-j}(k)}{\min(\xi, i)} \tag{6.5}$$

where $\xi \geqslant 1$ represents the memory of the system, i.e. the frames that will be used. If $\xi = 1$ we only use the current frame.

Step3: Finally, the action k_{max}^i that maximises $F_\xi^i(k)$ over the sliding window of length ξ is chosen to represent the type of action in frame i

$$k_{max}^i = \arg\max_k \left(F_\xi^i(k)\right). \tag{6.6}$$

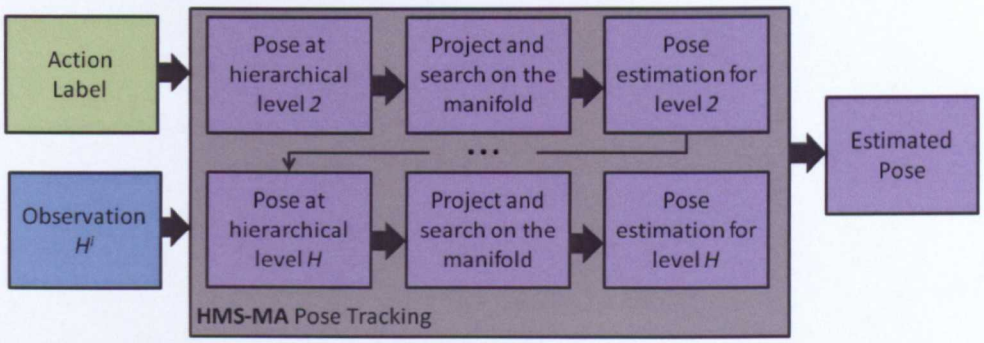


Figure 6.3: Pose tracking pipeline for multi-activity scenario.

6.3.2 Pose Tracking

HMS-MA searches the rest of the levels of the hierarchy of manifolds that correspond to the selected action $k = k^i_{max}$ where $p^i_{1,1} = p^i(k^i_{max})$ (Figure 6.3). Specifically, for every layer h ($h > 1$) and subspace l of the hierarchy first the point from the pose subspace l', from the previous hierarchical level $h-1$ is projected to the child pose subspace $P_{h,l,k}$ using the function $\omega_{h,l,k}$ to restrict the part of the human model that is searched:

$$p^i_{h,l} = \omega_{h,l,k}\left(p^i_{h-1,l'}\right). \tag{6.7}$$

Then the high-dimensional point $p^i_{h,l}$ is projected to the low-dimension space $\mathbb{R}^{d_{h,l,k}}$

$$q^i_{h,l} = \varphi_{h,l,k}\left(p^i_{h,l}\right). \tag{6.8}$$

Then, the solution is constrained using the action manifold. Specifically, HMS-MA considers the closest point $\hat{q}^i_{h,l}$ to the point $q^i_{h,l}$ in $Q_{h,l,k}$.

Afterwards, a gradient descent optimisation algorithm is used in order to find a local maximum where putative solutions are evaluated in the high-dimensional space using the observation function. The output of the algorithm

is the optimal point $\hat{p}^i_{h,l}$ that maximises the observation function $f^{r,i}_{h,l}$

$$\hat{p}^i_{h,l} = \left\{ p^r_{h,l} : \max_r f^{r,i}_{h,l} \right\}. \tag{6.9}$$

Finally, the pose of the model p^i is estimated by concatenating the body parts estimated at the last level of the hierarchy.

$$p^i = \left\{ \hat{p}^i_{h',1}, \hat{p}^i_{h',2}, ..., \hat{p}^i_{h',n} \right\}. \tag{6.10}$$

The estimated pose p^i is used as input in the next frame. The HMS-MA method is applied for every frame of the video sequence. The final results are the action labels k^i_{max} and the $3D$ poses p^i for each frame i of the sequence.

The HMS-MA method is an extension of HMS-HTLE methods to cover multi-activity scenario problems. Therefore, the pose tracking results are not expected to be an improvement upon the results presented in the previous chapter where action labels were considered known. The advantage of this method is that the proposed HMS and HTLE methods may be used in more complicated scenarios containing multiple actions. Finally, HMS-MA requires some extra computational resources to run HMS(1) for all activities and recognising the action for each frame.

6.4 Observation Function

In this section we discuss the observation function that is used to compare the observation and the pose hypothesis. We use two type of data for evaluation the HMS-MA method i.e. multi-camera and Kinect datasets. For multi-camera datasets we use the same observation function as described in section 5.4. For the Kinect dataset, the foreground colour image and depth map for each frame

are used to represent the sequence of observations, as seen in section 3.5.3. The human pose hypothesis, based on the $3D$ human body model M as defined in section 3.4, is projected on two different spaces to facilitate the comparison to the observation: Firstly, the human pose hypothesis is projected onto the Kinect image plane to enable comparison with the foreground colour image. Secondly, it is projected on depth map space to allow comparison with the foreground depth map, as seen in figure 6.4, where every point of the projected model represents the $3D$ Euclidean distance from the corresponding $3D$ point to the camera plane (x_{c^k}, y_{c^k}) as seen in section 3.2.1.

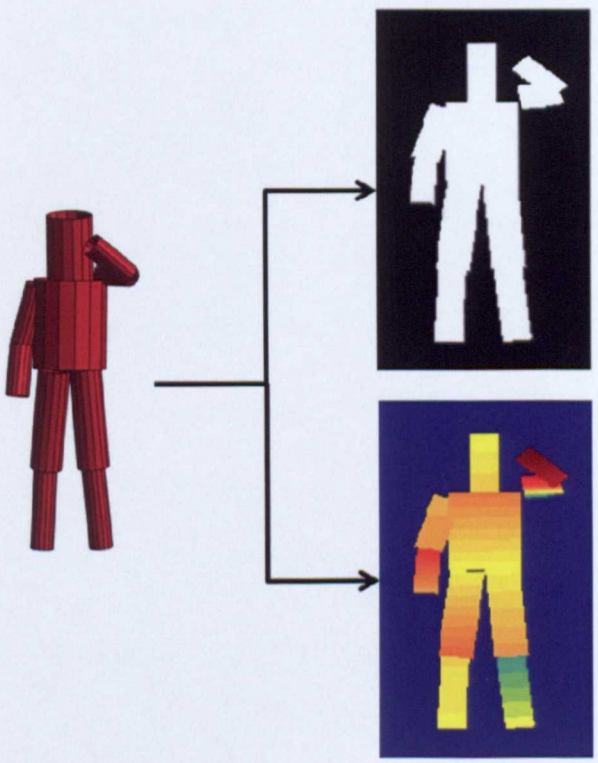


Figure 6.4: The $3D$ pose hypothesis projected on the image plane and the depth map space. In the latter projection, pseudo-colour is used to represent depth values.

In order to compare the observation with the pose hypothesis, an observation function is required. The observation function that is used for the HumanEva dataset is the same as the one that was presented in section 5.4. For the G3D dataset, the observation function is based on the colour image and the

depth map of each frame. The first part of the observation function is based on the $2D$ area of the silhouettes. The two areas of observation H and the projection of pose hypothesis are compared using function s_1, as defined below. Although the definition of s_1 is similar to the one in section 4.4, they differ as M and H represent areas instead of volumes in equation 6.11.

$$s_1(M,H) = \frac{\mid M \cap H \mid}{\mid M \mid}. \tag{6.11}$$

The second part of the observation function is based on colour similarity, similar to the one defined in section 5.4

$$s_2(M,H) = \frac{1}{L}\sum_{j=1}^{L}\frac{\sum_{v=1}^{V_j} C_j^{i,v}}{V_j} \tag{6.12}$$

where V_j is the total size in pixels of each area and $C_j^{i,v}$ is the binary colour similarity variable, which is introduced to emphasise significant differences between pixel colours.

The third part of the observation function compares the depth information between the observation and pose hypothesis M

$$s_3(M,H) = \frac{\sum_{v\in\tilde{M}} |m_v - h_v|}{\mid M \mid} \tag{6.13}$$

where $m_v \in M$ and $h_v \in H$ for every pixel v in the depth map.

Finally, the observation function is given by the weighted mean

$$f(M,H) = \sum_{k=1}^{3} w_k s_k(M,H) \tag{6.14}$$

where w_k is the weight that allows us to change the balance between observation functions, where $\sum_{k=1}^{3} w_k = 1$.

6.5 Evaluation

6.5.1 Overview

In this section we analyse the parameters and depict the results of the HMS-MA method. Firstly, the publicly available datasets that are used and the training process are presented. Then, the observation function for the G3D dataset is evaluated. Finally, HMS-MA is tested using a variety of datasets and compared with state-of-the-art multi-activity pose tracking methods.

6.5.2 Datasets and Training

We evaluate HMS-MA on two publicly available datasets. First, we use the multiple action sequences of the HumanEva (HE) II datasets which are presented in section 3.3.2, i.e. HEII-S2 frames 1 to 710 (1−390 walking, 391−710 jogging), and HEII-S4, frames 4 to 710 (4 − 370 walking, 371 − 710 jogging). For all sequences we used human actions captured by 4 cameras and calibration information for each of them. The HEI-S2 walking, trial-3 and HEI-S2 jogging, trial-3 sequences belonging to HumanEva I dataset are the datasets used for training the HTLE models, as discussed in section 5.2. Second, we use the boxing scenario of the G3D dataset which is presented in section 3.3.3 since it provides training data for a wide range of actions (punch left, punch right, kick left, kick right, defence). In between actions, subjects return to an "inaction" pose, i.e. standing still. Subjects $1 - 5$ are used for training, while the remaining subjects $6 - 10$ for testing our method. The range of frames of each action was extended by 20 frames in order to include some inaction frames of training dataset. In all experiments, the tracker is initialised with the first frame of the sequence using the ground truth pose.

The value of variable ξ is estimated by applying HMS-MA on the training dataset for different values of ξ and selecting the optimal one: they correspond to $\xi = 10$ and $\xi = 2$ for the HEII and the G3D datasets respectively. Such a difference of optimal ξ is justified as actions are shorter and change more frequently in G3D than in HumanEva.

6.5.3 Validation of Observation Function

The observation function that is used for the HumanEva datasets was evaluated in sections 5.5.3 and 4.5.2. Here we evaluate the observation function that was presented in section 6.4. In Figure 6.5 we see the inverse relationship between the average error per frame using HMS-MA configuration for G3D subject 7 (red line) and the values of the observation function f for every frame i (in black) using $w_1 = w_2 = w_3 = 1/3$. The correlation coefficient of the error and the observation function is -0.38, which implies negative linear correlation, and this is statistically significant because the p-value is sufficiently small ($p = 2 \cdot 10^{-11}$) [33]. This confirms that maximising the proposed observation function leads to minimising the pose tracking error.

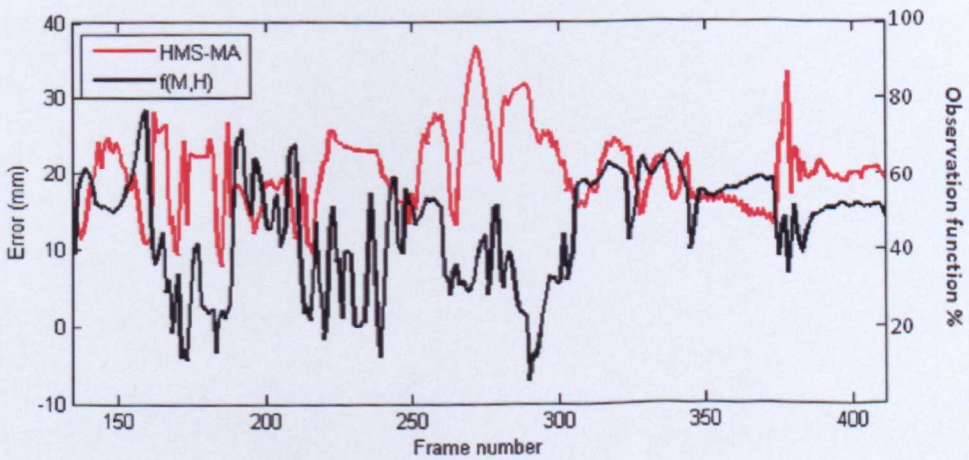


Figure 6.5: Error of HMS-MA and observation functions per frame.

6.5.4 Action Classification Results

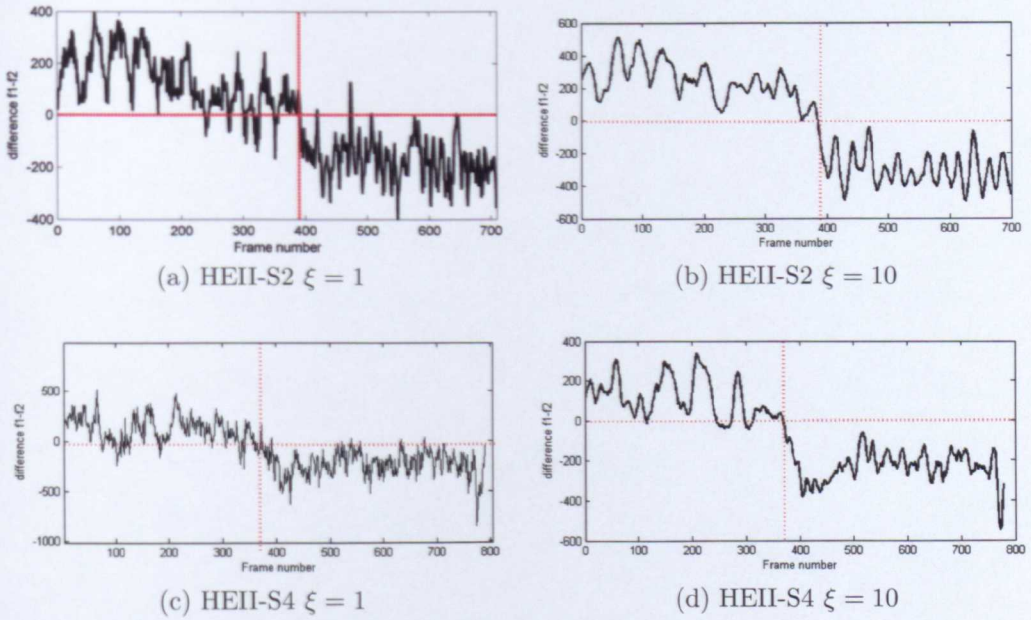
In this section we present action classification results produced by the HMS-MA pose tracking method presented in section 6.3. In all experiments we use threshold $T = 100\%$ in equation 5.8 and $a = 0.2$ in equation 5.15. All observation function weights are set as equal, i.e. $w_1 = w_2 = 1/2$ (equation 5.17) in multi-camera experiments and $w_1 = w_2 = w_3 = 1/3$ in Kinect experiments (equation 6.14).

In Figure 6.6, the difference between the two functions, $F_\xi^i(1)$ and $F_\xi^i(2)$, is estimated for every frame i, as described in section 6.3, for the HumanEvaII S2 (Figure 6.6a,b) and HumanEvaII S4 (Figure 6.6c,d) datasets. The $F_\xi^i(1)$ corresponds to the walking action, and the $F_\xi^i(2)$ to the jogging action. The horizontal red line is the zero axis and represents our activity decision boundary, while the vertical line at frame 390 for HEII-S2 and 370 for HEII-S4, depicts the time of change of activity type, i.e. the last frame of the walking action. When a curve is above zero, it means that walking is the recognised action; otherwise it is jogging. Overall, the classification success rate for the walking and the jogging actions for HEII-S2 dataset are 92% and 98% respectively using $\xi = 1$ and 99% and 100% using $\xi = 10$ and for HEII-S4 are 85% and 98% respectively using $\xi = 1$ and 90% and 100% using $\xi = 10$.

In Figure 6.7 we present results for HumanEvaII S2 data using HMS-MA(1-5) method. We can see the classification success rate (red line) and the corresponding error (blue line) for different values of variable ξ. The best results are for $\xi = 10$ i.e. classification success rate 99.6% and average error $73.5mm$.

Therefore, using a high value of ξ, (e.g. $\xi = 10$) improves the results of the action classification Also comparing the Figures 6.6a and b and Figures 6.6c and d we can see that for higher value of the variable $\xi = 10$ the difference between functions, $F_{10}^i(1)$ and $F_{10}^i(2)$ are represented by a smoother curve. Since the actions are properly recognised for such high percentage of the sequence, pose

estimation results are expected to be similar to the ones derived in section 5.5.5, where action labels were known.



(a) HEII-S2 $\xi = 1$ (b) HEII-S2 $\xi = 10$

(c) HEII-S4 $\xi = 1$ (d) HEII-S4 $\xi = 10$

Figure 6.6: HMS-MA method for action classification. Difference of functions $F_\xi^i(1)$ and $F_\xi^i(2)$ for a) HEII-S2 $\xi = 1$, b) HEII-S2 $\xi = 10$, c) HEII-S4 $\xi = 1$ and d) HEII-S4 $\xi = 10$.

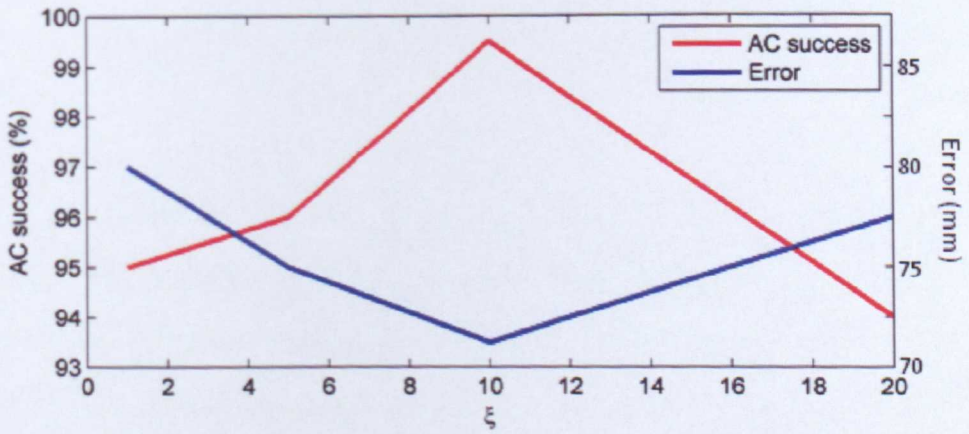


Figure 6.7: AC and error results using HMS-MA using different values of ξ for HEIIS2.

In Figure 6.8 we present results for G3D data subject 8 using HMS-MA(1-5) method. We can see the classification success rate (red line) and the corresponding error (blue line) for different values of variable ξ. The best results are for $\xi = 2$ i.e. 99.6% for classification success rate and $13.7mm$ for the error.

	punch right	punch left	kick right	kick left	defend
punch right	98.6	0	0	1	0.4
punch left,	1	96	0	0	3
kick right	0	0	100	0	0
kick left	0	0	0	100	0
defend	2.2	0	0	0	97.8

Table 6.1: Confusion matrix for subjects 6 to 10 using $\xi = 2$.

The optimal ξ is smaller than this on the previews results as in G3D data actions are shorter and change more frequently than in the HumanEva dataset.

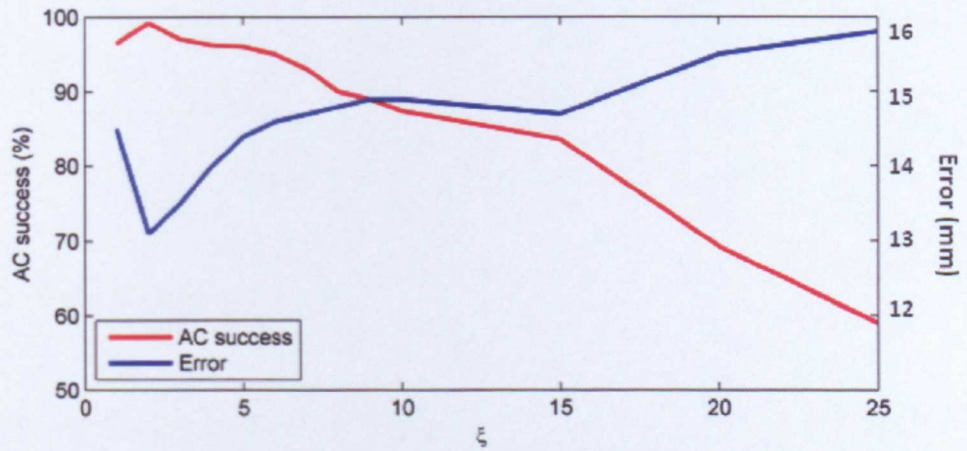


Figure 6.8: AC and error results using HMS-MA using different values of ξ for G3D data subject 8.

Similarly, in Table 6.1 we see the confusion matrix, which presents the classification results for each activity using the HMS-MA(1-5) method applied in G3D dataset for subjects 6 to 10, using $\xi = 2$ in equation 6.5. Also, in Tables 6.2 and 6.3 we can see the action classification success rate of every subject for each activity using $\xi = 1$ and $\xi = 2$, respectively. Using $\xi = 2$ the action classification success rate is higher (98.4%) than using $\xi = 1$ (94.2%), for all cases. Overall, we can see that HMS-MA is able to detect the correct action with a total success rate of 98.4% in the G3D dataset.

	punch right	punch left	kick right	kick left	defend	total
sub6	92	47	88	100	100	85.4
sub7	79	100	100	100	90	93.8
sub8	86	100	100	100	96	96.4
sub9	100	100	100	100	100	100
sub10	100	82	100	100	97	95.8
total	91.4	85.8	97.6	100	96.6	94.2

Table 6.2: Percentage success of every subject for each activity using $\xi = 1$.

	punch right	punch left	kick right	kick left	defend	total
sub6	100	80	100	100	100	96
sub7	93	100	100	100	100	98.6
sub8	100	100	100	100	100	100
sub9	100	100	100	100	100	100
sub10	100	100	100	100	93	98.6
total	98.6	96	100	100	98.6	98.6

Table 6.3: Percentage success of every subject for each activity using $\xi = 2$.

6.5.5 Pose Tracking Results

In this section we present pose tracking results produced by the HMS-MA. In Figure 6.9, the results for HE-II S2 walking and jogging actions per frame are shown using HMS-MA(1-5) and $\xi = 1$. The grey areas present the frames that the action classification process failed. The pose tracking error in the grey areas ($103mm$) is higher than the average error , as pose tracking depends highly on action classification. The average error along the whole multi-activity scenario is fairly constant (walking action: $70mm$, jogging: $77mm$). Overall, HMS-MA(1-5) using the optimal memory value, i.e. $\xi = 10$ performs similarly to APF [89] and H-APF [78] however the complexity of HMS-MA is significantly lower, as seen in Table 6.4 and Figure 6.10. HMS-MA performs similarly to HMS, with the additional advantage that action segmentation is automated.

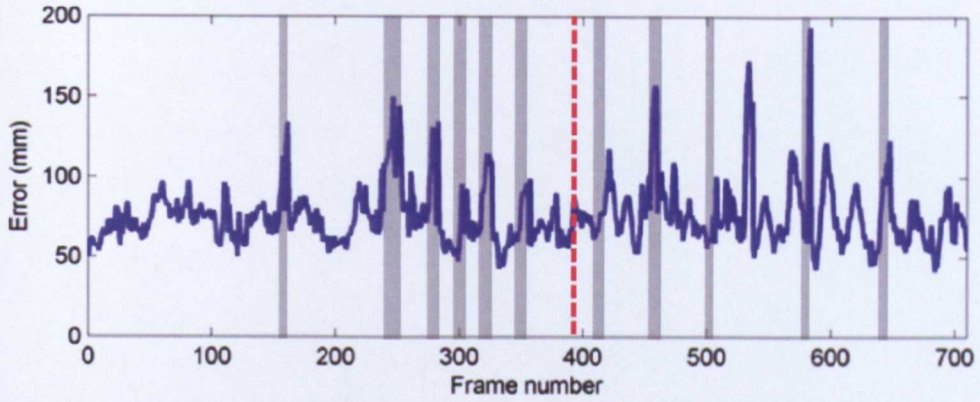


Figure 6.9: HMS-MA results for HEII-S2 walking and jogging actions. The grey areas present the frames that the action classification process failed.

	HEIIS2walk	HEIIS2jog	HEIIS4walk	HEIIS4jog	Comp.
APF	76	85	60	93	1000
H-APF*	75.2	75.2	81.8	81.8	500
HMS	63.1	80.9	62.5	102.4	130
HMS-MA	70	77	63	100	140

Table 6.4: Average error in *mm* and complexity (number of evaluations) for different configurations of APF and HMS (*the H-APF results are the average of whole sequence).

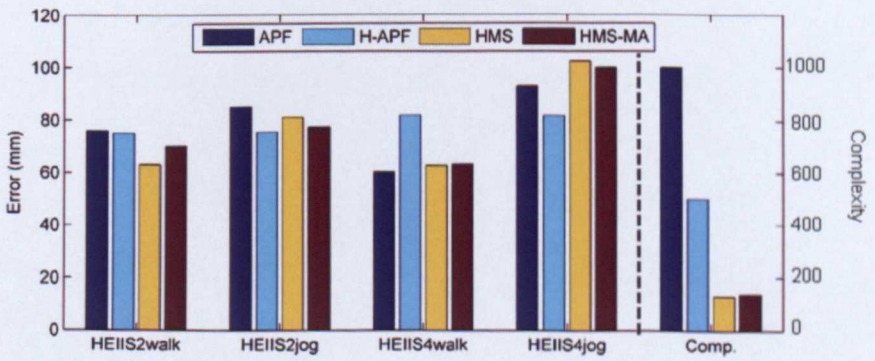


Figure 6.10: Average error in *mm* and complexity (number of evaluations) for different configurations of APF and HMS.

Compared to HMS, HMS-MA requires an extra computational cost for online action recognition, which is the cost of running HMS(1) for $K-1$ activities. In the walking-jogging scenario the overall computational complexity (mean number of observations per frame) increases by only 8% per frame in comparison with the HMS method (Section 5.5). Overall, the added complexity is a linear

function of the number of actions that are tested. HMS-MA pose tracking is not expected to outperform HMS, because in the experiments of section 5.5 the action label was considered known. Since recognition of the jogging action is highly reliable (98%), it is not surprising that the performance of HMS-MA is similar to HMS for the jogging part.

In the Figures 6.11 and 6.12 visual results for HEII-S2 using HMS-MA(1-5) are presented using $\xi = 2$.

In Tables 6.5, 6.6 and Figure 6.13, pose tracking results using the HMS-MA(1-5) method for every subject (6 to 10) and every action are summarised, using $\xi = 1$ and $\xi = 2$ respectively. The total average difference from [86] for all the subjects is $16mm$ for $\xi = 1$ and $12.6mm$ for $\xi = 2$. Similarly, to the previous results, higher action classification success rate using $\xi = 2$ (section 6.5.4) leads to more accurate pose tracking results.

To place our performance in context, we consider that in the evaluation of [86], which produces the "ground truth" measurements, a "true positive joint" is considered when the Euclidean distance of the estimated joint from the real one is within $100mm$. In addition, according to the analysis in [46], the depth resolution and the standard deviation of depth error of Kinect is $25mm$ and $14mm$ respectively, when the object is at $3m$ distance from the sensor, which is the case for the subjects in the G3D dataset. Therefore, we can claim that the accuracy of HMS-MA is comparable to [86], since the difference of performance is within the statistical error of depth measurements.

In Figure 6.14 we see the error in mm for every frame, for subjects 6 to 10. The colour dashed lines specify the different action types, according to the ground truth, while the colour dots on the horizontal axis represent the estimated action for every frame.

Figure 6.11: Results using HMS-MA(1-5) at HEII-S2 subject. Left and right part of the estimated skeleton are shown in red and blue respectively.

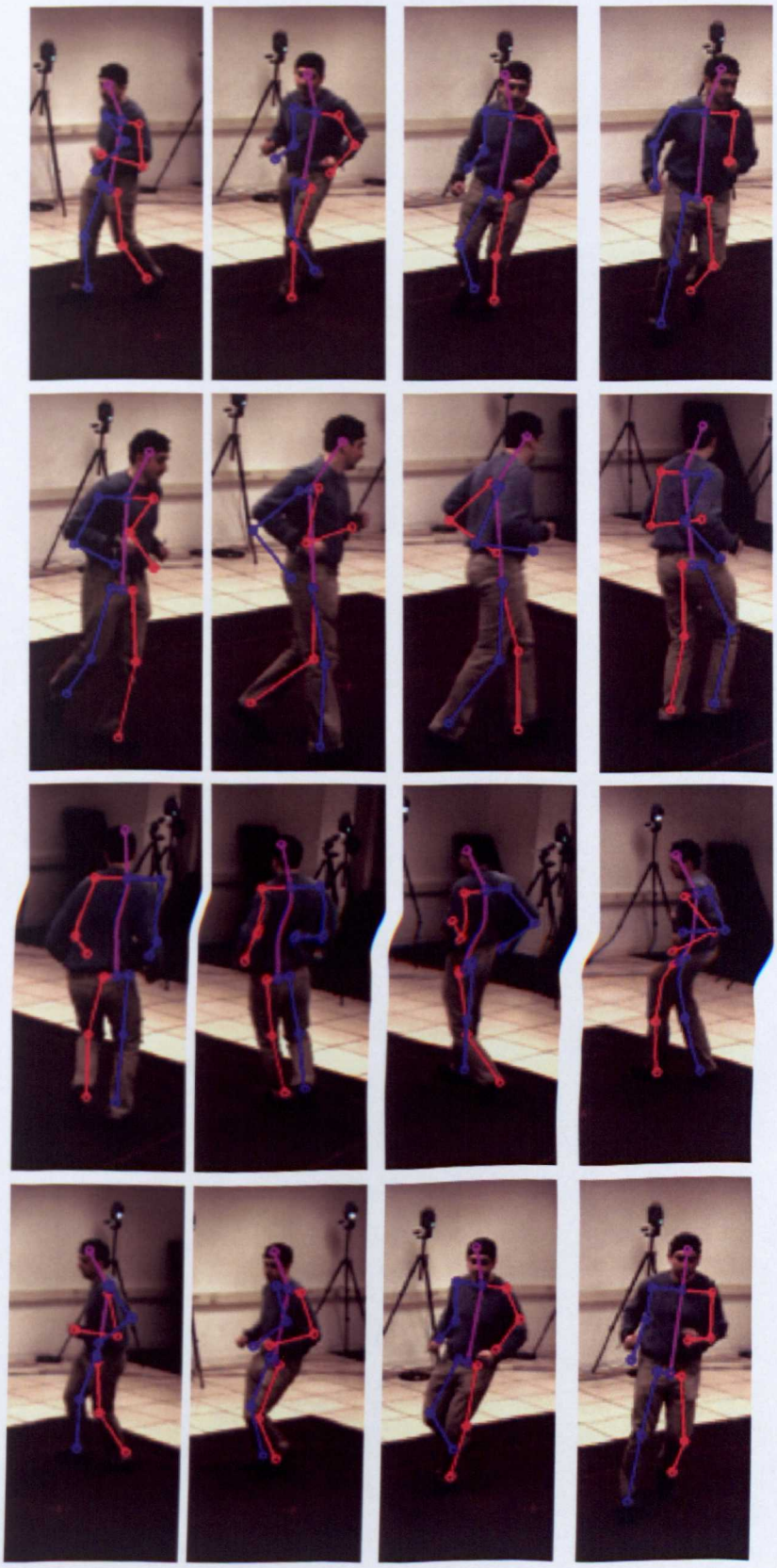


Figure 6.12: Results using HMS-MA(1-5) at HEII S2 subject. Left and right part of the estimated skeleton are shown in red and blue respectively.

	punch right	punch left	kick right	kick left	defend	total
sub6	29.7	43	71.2	42.6	27.7	22.9
sub7	19.5	14.3	19	16.6	22.1	14.5
sub8	24	18.4	22.6	22.8	15.6	14.8
sub9	16.5	15.1	25.8	21.4	10.8	13.2
sub10	14	18.6	32.2	28.4	31.4	12.6
total	20.5	21.5	35.3	26.1	21.9	16

Table 6.5: Average error results in mm per action using $\xi = 1$.

	punch right	punch left	kick right	kick left	defend	total
sub6	22.8	19.9	20.6	15.5	24.4	14.6
sub7	18.3	14.8	21.7	15.8	20.1	13.7
sub8	18	17.3	17.5	14.1	16.5	13.1
sub9	16.5	15.1	25.8	21.4	10.8	13.2
sub10	11.3	14.5	26.1	27.2	11.2	8.6
total	17.3	16.3	22.3	18.8	16.6	12.6

Table 6.6: Average error results in mm per action using $\xi = 2$.

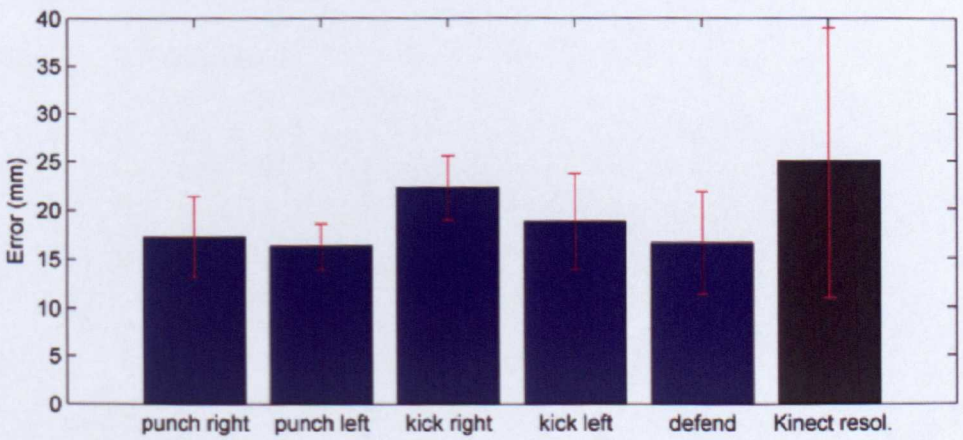
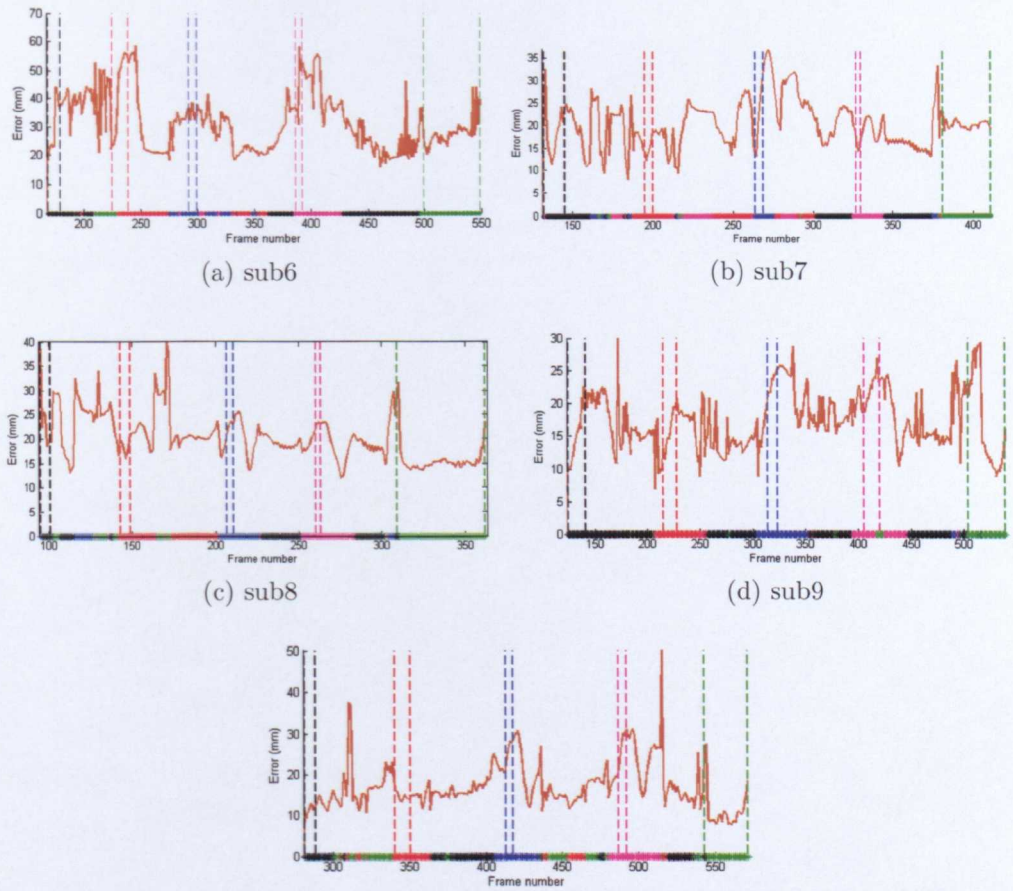


Figure 6.13: Average difference from [86] for G3D dataset in mm per action.



(a) sub6

(b) sub7

(c) sub8

(d) sub9

(e) sub10

Figure 6.14: Results using HMS-MA(1-5) for subject 6-10 and $\xi = 2$. The colour dashed lines specify the different action types, according to the ground truth. The colour dots on the horizontal axis represent the estimated action for every frame. The following colour code is used: black-punch right, red-punch left, blue-kick right, magenda-kick left, green-defend.

In Figure 6.15 we see the error per frame for the HMS-MA method applied in G3D subject 9, using HMS-MA(1) and HMS-MA(1-5). We can see that the error when we use all the levels of the hierarchy is lower that when we use only the first level. Therefore while HMS-MA(1) is used to recognise the action of every frame, HMS-MA(1-5) is more accurate to estimate pose. A visual example of this is given in Figure 6.16. In the left image a pose is estimated using HMS-MA(1) and in the right using HMS-MA(1-5). The HMS-MA(1-5) estimation is more accurate than the one provided by HMS-MA(1), since HMS-MA(1-5) is able to adapt to stylistic difference between samples in the training and the testing datasets.

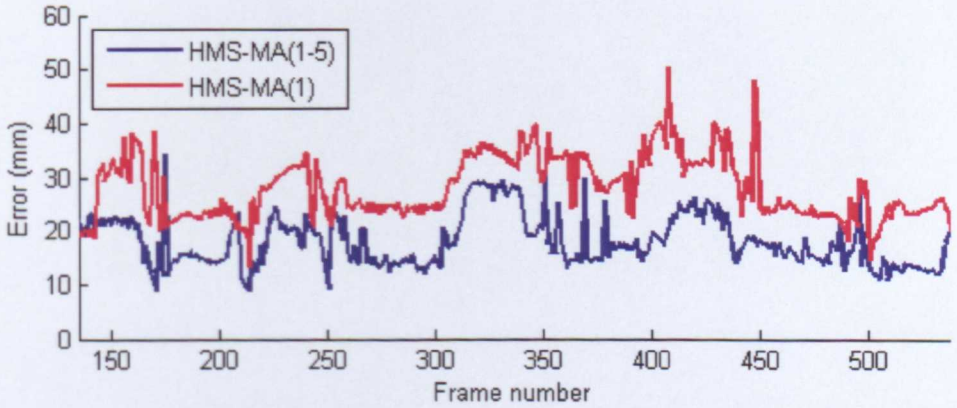


Figure 6.15: Results for G3D subject 9 using HMS-MA(1) (red) and HMS-MA(1-5) (blue) methods.

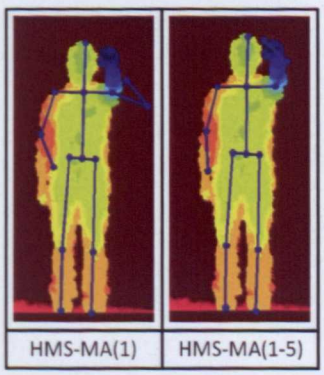


Figure 6.16: Depth map and the pose estimation using HMS-MA(1) (left image) and HMS-MA(1-5) (right image).

In the Figures 6.17 we see visual results for G3D subject9 using HMS-

MA(1-5).

Figure 6.17: Results using HMS-MA(1-5) at G3D dataset subject9.

6.6 Discussion

In this chapter we presented a novel human pose tracking methodology for multi-activity scenarios called HMS-MA. The HTLE method is used to generate a set of hierarchies of manifolds. Each hierarchy represents a single activity. The HMS-MA method is applied to these hierarchies in two stages. First, the action of every frame is recognised, and then the pose is estimated, based on the result of the first step. The HMS-MA is applied in publicly available datasets, and results demonstrate the ability of the method to deal with multi-activity scenario problems in pose tracking and online action recognition.

TLE has been used before for action recognition [52], by comparing a

whole sequence with the manifold in low-dimensional space, in an offline manner. On the other hand, HMS-MA is able to produce frame-based action recognition results and use them for online pose tracking problems.

The system is equipped with short memory to improve online action recognition results. The size of the memory is represented by a single variable, whose value depends on the frequency and the speed of actions.

The HMS-MA method extends the HMS-HTLE methods in order to deal with multi-activity scenarios. The extra complexity of the action recognition step that is required in HMS-MA is relatively low compared to the original complexity of HMS-HTLE. In addition, the accuracy of HMS-MA is similar to state-of-the-art methods [24, 77, 29, 27], but with significantly lower complexity.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis we have proposed novel generative $3D$ human pose tracking methods for single and multi-activity scenarios. The pose tracking problem is very challenging because of the complexity and the high-dimensionality of the human posture. We proposed solutions that may achieve accurate results with low-computational cost, compared to other generative methods.

In order to constrain the search of the optimal pose, we use dimensionality reduction methods in order to learn low-dimensional models from training datasets. In particular, we selected TLE as the base dimensionality reduction approach, as it is able to suppresses stylistic variation and produce compact manifolds which may be considered almost $1D$ in most cases and therefore are suitable for fast exploration.

However, such results are constrained by the training dataset and may not accurately match the potential style of the observed sequence. In order to move beyond this constraint and generate poses that correspond to unseen stylis-

tic variations of actions, we represent human poses using multiple levels, e.g. two levels in MPLC and five levels in HMS-HTLE. Searching through those structures is driven by the observation function, so to minimise the computational cost of the method.

In addition, we intentionally avoid using particle filtering because of its high computational cost. Instead, fast deterministic gradient-based optimisation methods are chosen. Generally in gradient-based optimisation, results depend on how close the global optimal solution is to the initial pose, because searching may be trapped to a local optimal. Our methods search through multiple levels to reduce this effect and result to better accuracy.

Finally, we deal with multiple-action scenarios by combining pose tracking with online action recognition. Specifically, a short memory action recognition method is used to assigned an action label on each frame. Such a memory mechanism allows generating a smoother and more accurate representation of actions.

In Chapter 4, the MPLC pose tracking method was presented. Firstly, in the MP stage, the observation pose is compared with the model hypothesis constrained by a TLE low-dimensional manifold to avoid divergence of pose tracking. Secondly, the LC stage deals with the problem of stylistic variations of human activity by refining each limb individually. The LC method is triggered to search for the optimal position of only the body parts that have erroneously determined during the MP method.

In Chapter 5, the HMS-HTLE pose tracking method was presented. First, HTLE, a novel hierarchical dimensionality reduction method, was introduced. HTLE generates a hierarchy of manifolds from a single action training dataset, based on the hierarchy of human body. HMS searches efficiently through the HTLE hierarchy, driven by the observation function. Since searching is mainly performed in compact TLE manifolds, a low number of evaluations is sufficient

for each level. HMS-HTLE is able to combine sub-poses from different manifolds to represent unseen poses. The result is a human pose in which the individual body parts are generated independently, which may have not seen in the training dataset. The HMS-HTLE method improves the results of the human pose tracking problem when compared with the MPLC method, as it searches through more levels than MPLC.

The previous pose tracking methods may only be applied in single-action scenarios. In Chapter 6, the multi-activity $3D$ pose tracking method HMS-MA, was presented. The HTLE method is used to generate a set of hierarchies of manifolds and each hierarchy represents a single activity from the training dataset. When inferring unseen sequences, firstly the action of every frame is recognised. A short memory mechanism is used to provide reliable online action recognition results. Then, the hierarchy of manifolds that corresponds to the recognised action of the specific frame is searched for estimating the pose, as in chapter 5. HMS-MA is able to produce accurate pose tracking results in multi-activity scenarios without significantly increasing the computational cost, in comparison to HMS-HTLE.

7.2 Future Work

A challenge in pose tracking problems is to minimize the computational cost without affecting the accuracy of the method. The pose tracking methods which were presented in this thesis have low complexity, compared to other generative methods. However, the high-computational cost of calculating the observation function, prevents their use in real-time applications. Therefore, one future direction could be on optimising the algorithm of comparing the candidate pose to the input observation and on implementing the algorithm on a real-time platform (e.g. C/C++ using dedicated hardware).

One of the issues that were not investigated in this thesis was the smoothness of the sequence of estimated poses. Results from many pose tracking methods suffer from jitter, which is undesirable for some applications (e.g. virtual replay), as it causes a final outcome that looks unnatural. Therefore, future work could investigate techniques, such as dynamic models and operators, to smooth the sequence of estimated poses, which look natural.

Bibliography

[1] Ankur Agarwal and Bill Triggs. 3D human pose from silhouettes by relevance vector regression. *Computer Vision and Pattern Recognition. Proceedings of the IEEE Computer Society Conference on*, 2:882–888, 2004.

[2] Ankur Agarwal and Bill Triggs. Learning to track 3D human motion from silhouettes. *Proceedings of the twenty-first international conference on Machine learning*, 69:2, 2004.

[3] Ankur Agarwal and Bill Triggs. A local basis representation for estimating human pose from cluttered images. *Proceedings of the 7th Asian conference on Computer Vision ACCV*, 3851:50–59, 2006.

[4] Ankur Agarwal and Bill Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58, 2006.

[5] J.K. Aggarwal and Q Cai. Human motion analysis: A review. *Nonrigid and Articulated Motion Workshop. Proceedings., IEEE*, pages 90–102, 1997.

[6] Helmut Alt, Christian Knauer, and Carola Wenk. Matching polygonal curves with respect to the Fréchet distance. *Symposium on Theoretical Aspects of Computer Science STACS*, pages 63–74, 2001.

[7] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. *Computer Vision and Pattern Recognition. CVPR. IEEE Conference on*, 1(2):1014–1021, 2009.

[8] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-GaussianBayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.

[9] Alexandru O Balan, Leonid Sigal, and Michael J Black. A quantitative evaluation of video-based 3D person tracking. *Visual Surveillance and Performance Evaluation of Tracking and Surveillance. 2nd Joint IEEE International Workshop on*, pages 349–356, 2005.

[10] A.O. Balan and M.J. Black. An adaptive appearance model approach for model-based articulated object tracking. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 758–765. IEEE, 2006.

[11] Jan Bandouch, Florian Engstler, and Michael Beetz. Evaluation of hierarchical sampling strategies in 3d human pose estimation. In *Proceedings of the 19th British Machine Vision Conference (BMVC)*, 2008.

[12] Adam Baumberg. *Learning deformable models for tracking human motion.* PhD thesis, University of Leeds, 1996.

[13] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14:585–591, 2001.

[14] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–

1396, 2003.

[15] Victoria Bloom, Dimitrios Makris, and Vasileios Argyriou. G3D : A Gaming Action Dataset and Real Time Action Recognition Evaluation Framework. *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 7–12, 2012.

[16] Christoph Bregler, Jitendra Malik, and Katherine Pullen. Twist Based Acquisition and Tracking of Animal and Human Kinematics. *International Journal of Computer Vision*, 56(3):179–194, February 2004.

[17] Brown University. HumanEVA dataset, Available online: http://vision.cs.brown.edu/humaneva/ (accessed on 10 February 2013).

[18] Brown University. Image & MOCAP Synchronized Dataset(v.1.0), Available online: http://cs.brown.edu/~ls/Software/index.html (accessed on 10 February 2013). 2004.

[19] Fabrice Caillette, Aphrodite Galata, and Toby Tody Howard. Real-time 3-D human body tracking using learnt models of behaviour. *Computer Vision and Image Understanding*, 109(2):112–125, February 2008.

[20] Fabrice Caillette and Toby Howard. Real-time markerless human body tracking with multi-view 3-d voxel reconstruction. In *British Machine Vision Conference*, volume 2, pages 597–606. Citeseer, 2004.

[21] TJ Cham and JM Rehg. A multiple hypothesis approach to figure tracking. *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 2(Cvpr 99):239–244, 1999.

[22] K.M. Cheung, Simon Baker, and Takeo Kanade. Shape-from-silhouette across time part i: Theory and algorithms. *International Journal of Com-*

puter Vision, 62(3):221–247, 2005.

[23] K.M.G. Cheung. Visual hull construction, alignment and refinement for human kinematic modeling, motion tracking and rendering. *PhD diss., Carnegie Mellon University*, (October), 2003.

[24] John Darby, B. Li, and N. Costen. Backing off: Hierarchical decomposition of activity for 3d novel pose recovery. *British Machine Vision Conference*, 186:187–191, 2009.

[25] John Darby, Baihua Li, and Nicholas Costen. Tracking human pose with multiple activity models. *Pattern Recognition*, 43(9):3042–3058, September 2010.

[26] Quentin Delamarre and Olivier Faugeras. 3D articulated models and multiview tracking with physical forces. *Computer Vision and Image Understanding*, 2001.

[27] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. *Computer Vision and Pattern Recognition*, 2:126–133, 2000.

[28] J. Deutscher and B. North. Tracking through singularities and discontinuities by random sampling. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 2:1144–1149, 1999.

[29] J. Deutscher and Ian Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, February 2005.

[30] Ahmed Elgammal and Chan-Su Lee. Nonlinear manifold learning for dynamic shape and dynamic appearance. *Computer Vision and Image Un-*

derstanding, 106(1):31–46, April 2007.

[31] Ahmed Elgammal and C.S. Lee. Inferring 3D body pose from silhouettes using activity manifold learning. In *Computer Vision and Pattern Recognition. CVPR. Proceedings of the IEEE Computer Society Conference on*, volume 2, pages II–681. IEEE, 2004.

[32] Olivier Faugeras and Quang-Tuan Luang. The Geometry of Multiple Images: the laws that govern the formation of multiple images of a scene and some of their applications. *MIT press*, 2004.

[33] N Fenton and M Neil. Risk Assessment and Decision Analysis with Bayesian Networks. *CRC Press*, 2012.

[34] D. Gavrila and LS Davis. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. *International workshop on automatic face-and gesture-recognition*, pages 272–277, 1995.

[35] D. Gavrila and LS Davis. Tracking of humans in action: A 3-D model-based approach. In *ARPA Image Understanding Workshop*, pages 737–746. Citeseer, 1996.

[36] L Goncalves and E Di Bernardo. Monocular tracking of the human arm in 3D. *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 764–770, 1995.

[37] Kristen Grauman, Gregory Shakhnarovich, and Trevor Darrell. Inferring 3D structure with a statistical image-based shape model. *Proceedings Ninth IEEE International Conference on Computer Vision*, (Iccv):641–647 vol.1, 2003.

[38] Keith Grochow, S.L. Martin, A. Hertzmann, and Z. Popović. Style-based

inverse kinematics. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 522–531. ACM, 2004.

[39] M Hofmann and D M Gavrila. Multi-view 3D human pose estimation combining single-frame recovery, temporal integration and model adaptation. *IEEE Conference on Computer Vision and Pattern Recognition (2009)*, pages 2214–2221, 2009.

[40] Shaobo Hou, Aphrodite Galata, Fabrice Caillette, Neil Thacker, and Paul Bromiley. Real-time body tracking using a gaussian process latent variable model. *Computer Vision. ICCV. IEEE 11th International Conference on*, pages 1–8, 2007.

[41] Nicholas R Howe. Silhouette lookup for monocular 3D pose tracking. *Image and Vision Computing*, 25(3):331–341, 2007.

[42] S Ioffe and D Forsyth. Human tracking with mixtures of trees. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 1(C):690–695, 2001.

[43] Michael Isard and Andrew Blake. C ONDENSATION - Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[44] Maja J Jenkins, Odest Chadwicke and Mataric. A spatio-temporal extension to isomap nonlinear dimension reduction. *Proceedings of the twenty-first international conference on Machine learning*, page 56, 2004.

[45] I A Kakadiaris and D Metaxas. Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. *IEEE Conference on Computer Vision and Pattern Recognition (2008)*, 22(12):81–87, 1996.

[46] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors (Basel, Switzerland)*, 12(2):1437–54, January 2012.

[47] KN Kutulakos. A theory of shape by space carving. *International Journal of Computer Vision*, 2000.

[48] N.D. Lawrence. Gaussian process latent variable models for visualization of high dimensional data. *Advances in neural information processing systems*, 16:329–336, 2004.

[49] N.D. Lawrence and J. Quiñonero Candela. Local distance preservation in the GP-LVM through back constraints. In *Proceedings of the 23rd international conference on Machine learning*, pages 513–520. ACM, 2006.

[50] Neil D. Lawrence and Andrew J. Moore. Hierarchical Gaussian process latent variable models. *Proceedings of the 24th international conference on Machine learning - ICML '07*, pages 481–488, 2007.

[51] Michal Lewandowski and D Makris. Automatic configuration of spectral dimensionality reduction methods for 3D human pose estimation. *Vision Workshops (ICCV*, pages 1–8, 2009.

[52] Michal Lewandowski, Dimitrios Makris, and JC Nebel. View and style-independent action manifolds for human activity recognition. *Computer Vision-ECCV 2010*, pages 547–560, 2010.

[53] Michal Lewandowski, Dimitrios Makris, and Jean-Christophe Nebel. Automatic configuration of spectral dimensionality reduction methods. *Pattern Recognition Letters*, 31(12):1720–1727, September 2010.

[54] Michal Lewandowski, Jesus Martinez-del Rincon, Dimitrios Makris, and

Jean-Christophe Nebel. Temporal Extension of Laplacian Eigenmaps for Unsupervised Dimensionality Reduction of Time Series. *2010 20th International Conference on Pattern Recognition*, pages 161–164, August 2010.

[55] Ming Li. *Towards Real-Time Novel View Synthesis Using Visual Hulls.* Doctoral dissertation, Universität des Saarlandes, 2005.

[56] Zhengdong Lu, Miguel Carreira-Perpinan, and Cristian Sminchisescu. People tracking with the laplacian eigenmaps latent variable model. *Advances in neural information processing systems*, 20:1705–1712, 2007.

[57] D Marr and H K Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London Series B Containing papers of a Biological character Royal Society Great Britain*, 200(1140):269–294, 1978.

[58] Jesús Martinez Del Rincon, Jean-Christophe Nebel, and Dimitrios Makris. Graph-based Particle Filter for Human Tracking with Stylistic Variations. *British Machine Vision Conference*, pages 1–11, 2011.

[59] Georgios Mastorakis and Dimitrios Makris. Fall detection system using Kinects infrared sensor. *Journal of RealTime Image Processing*, 2012.

[60] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J Gortler, and Leonard McMillan. Image-based visual hulls. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374, 2000.

[61] Microsoft Kinect. Available online: http://www.xbox.com/en-us/kinect/ (accessed on 10 February 2013), 2010.

[62] Ivana Mikic, Mohan Trivedi, Edward Hunter, and Pamela Cosman. Human

body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53(3):199–223, 2003.

[63] T Moeslund. A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding*, 81(3):231–268, March 2001.

[64] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, November 2006.

[65] Greg Mori and Jitendra Malik. Estimating Human Body Configurations using Shape Context Matching. *Computer Vision ECCV 2002*, 3:150–180, 2002.

[66] Greg Mori and Jitendra Malik. Recovering 3D human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1052–1062, 2006.

[67] Greg Mori, Xiaofeng Ren, Alexei A Efros, and Jitendra Malik. Recovering human body configurations: Combining segmentation and recognition. *Computer Vision and Pattern Recognition. CVPR. Proceedings of the 2004 IEEE Computer Society Conference on*, 2:II–326, 2004.

[68] Alexandros Moutzouris, Jesus Martinez-del Rincon, Michal Lewandowski, Jean-Christophe Nebel, and Dimitrios Makris. Human pose tracking in low dimensional space enhanced by limb correction. *18th IEEE International Conference on Image Processing (ICIP)*, pages 2301–2304, September 2011.

[69] Alexandros Moutzouris, Jesus Martinez-del Rincon, Jean-Christophe Nebel, and Dimitrios Makris. Human Pose Tracking by Hierarchical Manifold Searching. *International Conference on Pattern Recognition (ICPR)*, pages 866–869, 2012.

[70] Eng-Jon Ong, Antonio S. Micilotta, Richard Bowden, and Adrian Hilton. Viewpoint invariant exemplar-based 3D human tracking. *Computer Vision and Image Understanding*, 104(2-3):178–189, November 2006.

[71] Dirk Ormoneit, Hedvig Sidenbladh, Michael J. Black, and Trevor Hastie. Learning and tracking cyclic human motion. *Advances in Neural Information Processing Systems*, pages 894–900, 2001.

[72] T Poggio and F Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[73] Ronald Poppe. Evaluating example-based pose estimation: Experiments on the humaneva sets. *Centre for Telematics and Information Technology University of Twente*, 2007.

[74] Ronald Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1-2):4–18, October 2007.

[75] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, June 2010.

[76] Ronald Poppe and M Poel. Comparison of silhouette shape descriptors for example-based human pose recovery. *7th International Conference on Automatic Face and Gesture Recognition FGR06*, pages 541–546, 2006.

[77] Leonid Raskin and Michael Rudzsky. Using Hierarchical Models for 3D Human Body-Part Tracking. *Proceedings of the British Machine*, pages 11–20, 2009.

[78] Leonid Raskin, Michael Rudzsky, and Ehud *Rivlin. 3D Human Body-Part* Tracking and Action Classification Using a Hierarchical Body Model. *Procedings of the British Machine Vision Conference 2009*, pages 12.1–12.11,

2009.

[79] Leonid Raskin, Michael Rudzsky, and Ehud Rivlin. Dimensionality reduction using a Gaussian Process Annealed Particle Filter for tracking and classification of articulated body motions. *Computer Vision and Image Understanding*, 115(4):503–519, April 2011.

[80] Xiaofeng Ren and AC Berg. Recovering human body configurations using pairwise constraints between parts. *Computer Vision, 2005. ICCV 2005.*, 2005.

[81] Gregory Rogez, Jonathan Rihan, Srikumar Ramalingam, Carlos Orrite, and Philip Torr. Randomized trees for human pose detection. *IEEE Conference on Computer Vision and Pattern Recognition (2008)*, 398(Cvpr 2008):1–8, 2008.

[82] R Rosales and S Sclaroff. Learning body pose via specialized maps. *NIPS*, 2002.

[83] S T Roweis and L K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science (New York, N.Y.)*, 290(5500):2323–6, December 2000.

[84] Loren Arthur Schwarz, Artashes Mkhitaryan, Diana Mateus, and Nassir Navab. Human skeleton tracking from depth data using geodesic distances and optical flow. *Image and Vision Computing*, 30(3):217–226, March 2012.

[85] G Shakhnarovich, P Viola, and T Darrell. Fast pose estimation with parameter-sensitive hashing. *Proceedings Ninth IEEE International Conference on Computer Vision*, 2:750–757 vol.2, 2003.

[86] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finoc-

chio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297 1304, June 2011.

[87] Hedvig Sidenbladh, M. Black, and Leonid Sigal. Implicit probabilistic models of human motion for synthesis and tracking. *Computer Vision ECCV*, pages 784–800, 2002.

[88] Hedvig Sidenbladh, Michael J Black, and David J Fleet. Stochastic tracking of 3D human figures using 2D image motion. *Computer Vision ECCV*, pages 702–718, 2000.

[89] Leonid Sigal, Alexandru O. Balan, and Michael J. Black. HumanEva : Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion. *International Journal of Computer Vision*, 87:4–27, August 2010.

[90] Leonid Sigal and MJ Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. *Brown Univertsity, Techniacl Report CS-06-08*, 2006.

[91] Leonid Sigal, Michael Isard, BH Sigelman, and MJ Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. *Advances in neural information processing systems*, 16, 2003.

[92] Cristian Sminchisescu. 3D Human Motion Analysis in Monocular Video Techniques and Challenges. *2006 IEEE International Conference on Video and Signal Based Surveillance*, 36:76–76, 2006.

[93] Cristian Sminchisescu and Allan Jepson. Generative modeling for continuous non-linearly embedded visual inference. *Proceedings of the twenty-first*

international conference on Machine learning, 2004.

[94] Cristian Sminchisescu, Atul Kanaujia, and Dimitris Metaxas. Learning joint top-down and bottom-up processes for 3d visual inference. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2:1743–1752, 2006.

[95] Cristian Sminchisescu and Bill Triggs. Estimating Articulated Human Motion with Covariance Scaled Sampling. *The International Journal of Robotics Research*, 22(6):371–391, 2003.

[96] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[97] R. Szeliski. Rapid octree construction from image sequences. *CVGIP Image Understanding*, 58:23–23, 1993.

[98] J B Tenenbaum, V de Silva, and J C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science (New York, N.Y.)*, 290(5500):2319–23, December 2000.

[99] TP Tian, R Li, and Stan Sclaroff. Tracking human body pose on a learned smooth space. *Technical Report*, 2005.

[100] Michael E Tipping. The relevance vector machine. *Advances in Neural Information Processing Systems*, 12:652–658, 2000.

[101] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.

[102] Raquel Urtasun, D.J. Fleet, and Pascal Fua. Monocular 3D tracking of

the golf swing. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 932 938. IEEE, 2005.

[103] Raquel Urtasun, D.J. Fleet, and Pascal Fua. 3D people tracking with Gaussian process dynamical models. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 238 245. Ieee, 2006.

[104] Raquel Urtasun, D.J. Fleet, Aaron Hertzmann, and Pascal Fua. Priors for people tracking from small training sets. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 403 410. IEEE, 2005.

[105] Raquel Urtasun and Pascal Fua. 3d human body tracking using deterministic temporal motion models. *Computer Vision-ECCV 2004*, 2004.

[106] R Van Der Merwe and E Wan. Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models. *Measurement*, 6(3):701–704, 2003.

[107] UK Vicon Motion System Ltd., Oxford. Vicon Optical Motion Catpure system *Available online: http://www.vicon.com/* (accessed on 10 February 2013).

[108] Jack Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models. *Advances in neural information*, 18:1441–1448, 2006.

[109] Jiang Wang. Mining actionlet ensemble for action recognition with depth cameras. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297, June 2012.

[110] Junqiu Wang and Yasushi Yagi. Adaptive mean-shift tracking with auxiliary particles. *IEEE transactions on systems man and cybernetics Part B Cybernetics a publication of the IEEE Systems Man and Cybernetics Society*, 39(6):1578–1589, 2009.

[111] Qiang Wang, Guangyou Xu, and Haizhou Ai. Learning object intrinsic structure for robust visual tracking. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 2:II–227–II–233, 2003.

[112] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys (CSUR)*, 38(4), December 2006.