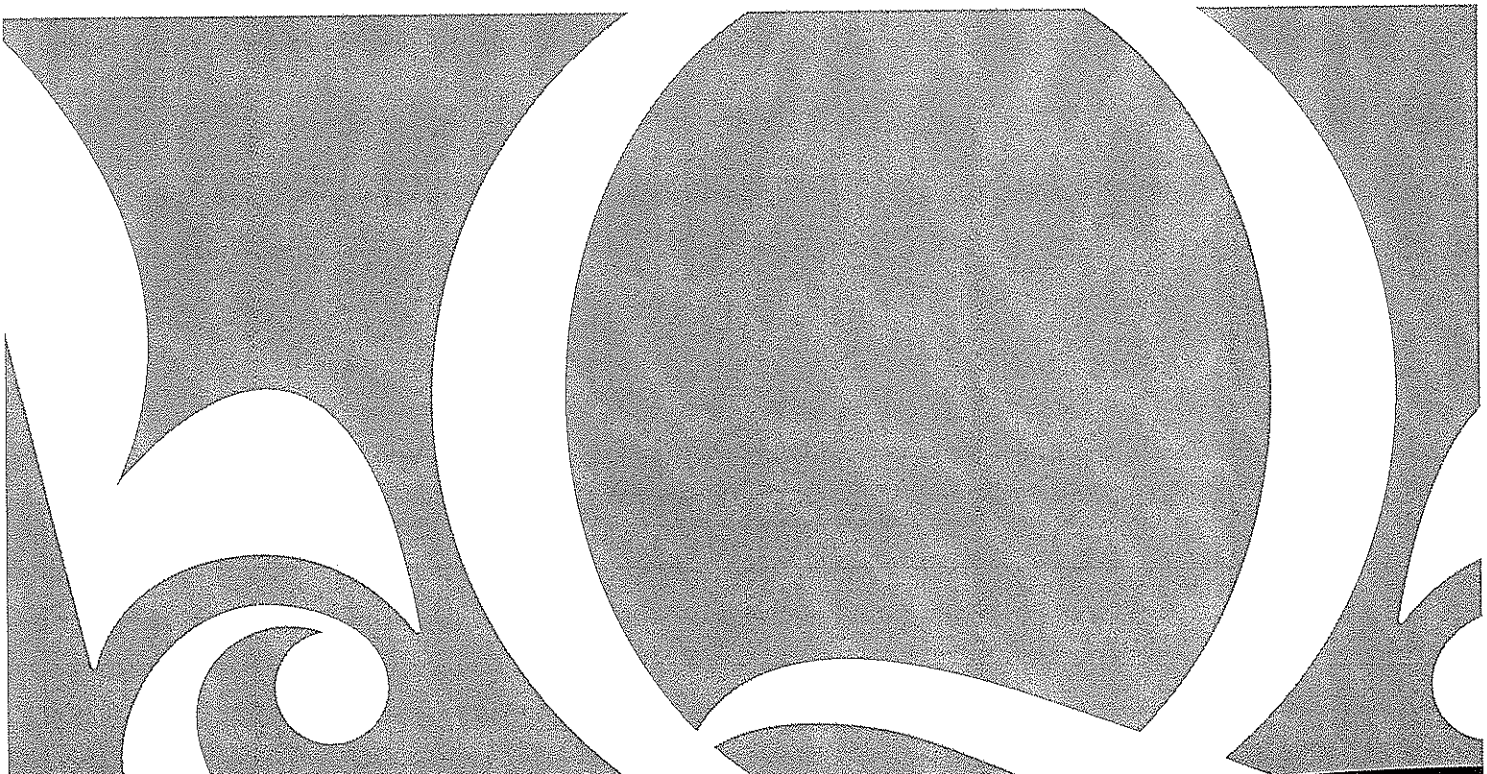


QUEEN MARY
AND WESTFIELD COLLEGE
UNIVERSITY OF LONDON

Department of Computer Science

Research Report No. RR-01-06 ISSN 1470-5559 December 2001

Learning the Visual Dynamics of Human body Motions

Eng-Jon Ong

Learning the Visual Dynamics of Human Body Motions

Eng-Jon Ong

A Thesis submitted to the University of London

for the degree of Doctor of Philosophy

Department of Computer Science
Queen Mary, University of London
2001

Abstract

To create a computer vision program that provides a visual interpretation of the motion of human bodies is a challenging problem. Effective solutions to the problem can potentially benefit areas such as motion capture, visually mediated human computer interaction and security applications concerned with understanding the intentions of human actions. However, interpretation of the motion patterns of any dynamical object inevitably requires an understanding of its underlying dynamics. Moreover, the complexity of the problem greatly increases when we consider the highly articulated object of a human body.

This thesis describes a framework for computationally learning the visual dynamics of human motions. Firstly, an analysis of appropriate representations for modelling the human body configurations is made. Issues arising from the representation's characteristics and complexity are identified and addressed. Important consideration is given to the reliability of such a representation when used in a visually driven system. Together, these enable an appropriate representation to be quantified. The motion of a human body is treated as an ordered sequence of instances of this representation.

Secondly, learning the dynamics of body motions is treated as a problem of computationally modelling a set of such ordered sequences. To this end, we propose a method whereby such a set is represented by a number of different prototypical example vectors. These example vectors can then be linearly combined to represent novel and valid body poses. Moreover, constraints on the possible combinations of examples are determined through learning. The mechanism developed was integrated into a dynamic tracking framework used for visually tracking ones' body motions.

Declaration

I declare that this thesis has been composed by myself, that it describes my own work, that it has not been accepted in any previous application for a degree, that all verbatim extracts are distinguished by quotation marks, and that all sources of information have been specifically acknowledged.

Additionally, some parts of the work presented in this thesis have been published in the following articles:

[1] E. Ong and S. Gong. Quantifying ambiguities in inferring vector-based 3-D models. In *The Eleventh British Machine Vision Conference*, Manchester, UK, September 2000.

[2] E. Ong and S. Gong. Tracking hybrid 2D-3D human models through multiple views. In *IEEE International Workshop on Modelling People*, Corfu, Greece, September 1999.

[3] E. Ong and S. Gong. A Dynamic 3D Human Model from Multiple Views. In *The Tenth British Machine Vision Conference*, Manchester, UK, September 1999.

Acknowledgements

I would like to thank the many people who have helped me walk the path towards this dissertation during my years here at Queen Mary.

First and foremost, I would like to thank my supervisor Prof. Shaogang Gong for his endless advice, discussions, encouragement and support over the last three years of my PhD research. I would also like to thank him for providing me with opportunities to attend various excellent conferences and workshops.

Additionally, I am greatly indebted to those who have spent countless hours and gone to great efforts to proof read and provide invaluable comments on this thesis. These people include: my supervisor, Prof. Dennis Parkinson, Prof. Heather Liddell, Dr. Richard Howarth, Dr. Peter McOwan, Andrew Anderson, Keith Anderson, Ting-Hsun Chang and Yongmin Li.

Furthermore, I have had the fortune in working with the other members of the Vision Group - Ting-Hsun Chang, Jamie Sherrah, Yongmin Li, Paul Verity, Jeffrey Ng, Andrew Anderson and Keith Anderson. As a result, I would like to thank them all for providing a friendly and supportive working environment.

Finally, but just as important, I would like to thank my parents and brother for providing limitless support during the last 3 years, regardless of the fact that they were on the other side of the planet in Malaysia.

Contents

List of Figures

Chapter 1

Introduction

1.1 Interpreting Human Body Motion Patterns

The human body is capable of undertaking a range of different tasks with ease due to its articulated structure. An important use of the human body's versatility is for communication. In this instance, the motions of different body parts or *biological motion* are often combined together to convey one's intentions [23]. Motion perception serves as an important component of the human visual system. This is supported by observations that specific areas of the human cortex are devoted to the detection of different motion types [19]. Motion is often used as a cue for focusing one's attention. Conversely, motionless objects are usually given less scrutiny than those in motion, implying that our visual system is well adapted to time-varying or *temporal* information [12]. For the human body, psychological studies by Johansson's moving light displays (MLD) [23] discovered the ability of humans to interpret various activities of other humans by observing spots attached to one's body parts. In particular, the gender of a person, or even the identity of a friend can be deduced by observing the motion of such spots [9].

Crucially, the first step in the interpretation of human body motion patterns is the recovery of motion information. Here, computational models for the human bodies (e.g. shape or 3-D skeletons) are used to explicitly predict and recover the motion performed by a person. In other words, motion is defined as variations

in the parameters of a computational model (e.g. shape vertices). The process of recovering and predicting such model parameters can be defined as *tracking*. Furthermore, when the model is used in conjunction with visual images to perform tracking, this term can be conveniently extended to *visual tracking*.

Computationally, the human body is a complex and dynamic assemblage. In attempting to track the body pose, one needs to account for the visual and structural features which are time varying and susceptible to noise. The aim of this research is to develop a robust and generic approach for the visual tracking of the human body motion patterns. To this end, the more specific problem of tracking the human body 3-D pose information or *body configuration*, via a vision-based medium needs to be tackled. Furthermore, one needs to account for the dynamic nature of motion patterns that manifests itself through a continuously changing sequence of human body configurations.

1.2 Approach

In order to visually track one's body configuration, we have adopted an approach that differs from typical approaches detailed in Chapter 2 in three key areas. The first is the form of information for representing the body configuration, which involves the unification of the visual appearance and body structure information into a hybrid vector. The second area revolves around the type of model chosen for computationally capturing the variations inherent in a human body's visual and structural information. This takes the form of an example-based framework, whereby prototypical examples are combined together to generate the information of a human body configuration. Finally, in order to determine the various parameters (i.e. prototypical examples and combination constraints), a learning based approach was adopted.

1.2.1 Unifying Different Modalities in a Single Representation

The human body, like any articulated object, contains a number of parts that are moveable. At any one time, a human body will consist of these parts all in a particular configuration. One can identify such a configuration of human body parts with the notion *human body configuration*. However, computationally, such a term is rather imprecise since it does not provide a clear and consistent description as to what a "body part configuration" really consists of. Therefore, a computational *representation* for the human body configuration is needed.

There are many choices for representing the information of the human body configuration. One can designate an individual type of representation as a *modality*. These modalities can take the form of visual appearances (e.g. contours or relative body parts positions). Alternatively, they can take the form of a 3-D virtual model consisting of 3-D vertices and constrained by an underlying 3-D skeleton.

Traditionally, these modalities were used individually [28]. An example would be to use only contours for tracking the body configurations of subjects [45, 3, 4, 55]. Alternatively, when 3-D models are used, computer graphics rendering methods are used to synthesise the visual appearance of the human body [16, 17, 32, 27, 70]. This in turn can be used for comparisons against an input image for correctness. However, we will see in this thesis that individually, these modalities have their own disadvantages. Visual appearances suffer from ambiguities in uniquely describing the body configuration due to the lack of depth information and self- occlusions. Three dimensional models require the synthesis of a realistic human body shape information, a necessarily complex task. However, the individual modalities have their advantages as well. 3-D based models have the advantage of being able to unambiguously represent human body config-

urations while 2-D based modalities already capture realistic visual appearances of the human body. Therefore, in order to exploit both such advantages, different modalities can be combined into a unified hybrid-vector form.

In an attempt to use the hybrid vector form to represent a human body's configuration, a number of issues must be addressed. Inevitably, ambiguities in the visual appearance components of the hybrid vector exist. It is therefore important to know how such visual ambiguities will affect a hybrid vector representation. To this end, this thesis will provide a study into the different aspects of how visual ambiguities will affect a representation (hybrid vector) that also contains 3-D structural information. An additional issue needs to be addressed. Having selected the representation, we have a form that can capture the information of different body parts at a time instance. However, it does not provide us with any knowledge about the dynamic characteristics of the body parts, that is, it does not tell us anything about the kinematics of the human body. To address this issue, we again adopted a different approach from the norm [32, 90, 89, 38], in the form of *example-based kinematics.*

1.2.2 Example-Based Kinematics

A *kinematics-based* approach is essentially a higher-level description of a model that captures the motion and other changes an object can undergo. With a hybrid vector, the kinematics model essentially accounts for the variations and constraints of the visual appearance and structural parameters of a human body. The kinematics approach however does not explicitly account for the causes of such variations in the human body information. Instead, such dynamics are implicitly captured, so that the end result (i.e. the positions and movements of parts) is the same as if we did have an accurate physical model that simulates the interactions between the muscular and skeletal structure.

Traditionally, kinematics-based approaches were used in 3-D based representa-

tions [32, 90, 89, 38]. In such cases, equations that determined the different values of the various 3-D components were defined *a priori*. This allows one to clearly define the dynamics of the 3-D structure. The kinematics parameters are then defined by the pre-determined constants in the equations. Such constants may, for example be used to constrain the possible movements of different body parts. Unfortunately, it is not clear as to how one can similarly model the dynamics of 2-D visual appearance components (e.g. contour components).

In this work, a set of prototypical hybrid vector examples, which captures a significant range of motion and variations of the human body, is used. Originally proposed for the recognition of rigid 3-D objects [76], this approach was extended to model the kinematics of faces [87, 78] and shapes of people [3, 4]. Since such an approach only requires the availability of a representation in a single vector form, it is suitable for the adopted hybrid vector. In place of the explicitly defined equations, the kinematics were instead defined as linear combinations of the prototypical examples.

In this example-based approach, the prototypical examples and the constraints on the possible combinations define the kinematics parameters. To make use of such a model, one then needs to determine these parameters. To this end, three key issues need to be dealt with; the number and contents of necessary and sufficient examples, defining the example combination constraints and determining its parameters. We will see in Chapter 6 how the constraints are defined. We find that it is not clear, analytically or intuitively, what the values of the kinematics parameters are. However, it is possible to automatically adapt the kinematics parameters to an acquired *training set* of hybrid vectors representing valid human body configurations. Such a process of adaptation can alternatively be thought of as automating the discovery of the necessary parameter values.

1.2.3 Learning the Kinematics Parameters

As there are many opinions on the meaning of the term "learning" [57], it would be advantageous to provide a more specific notion of "learning" used in this thesis. Our definition of learning will draw from works on neural networks [73]:

Learning is a process by which the parameters of a computational model are adapted to fit a given set of known examples.

Here, the computational model is the example-based kinematics model described above. The parameters are the kinematics parameters, such as joint angles or vertex 3-D co-ordinates. The set of known examples is a collection of hybrid vectors representing different human body configurations.

The task of learning achieves more than mere memorisation of the training set. We not only require the learnt model to reproduce consistent approximations with the training set, but it must also generate *novel* and valid examples. That is, the learning process is required to perform both memory recall and inference.

In this work, one needs to consider the spatial characteristics of the representation used. It is possible that a hybrid vector, essentially a high dimensional point, represents a body configuration. A complete training set can be thought of as a cloud of high dimensional points. Intuitively, a model that covers all possible body configurations would be the equivalent of an "enclosing shell" around such a cloud of points. However, in the domain of modelling the human body *motion* patterns, such a model alone is insufficient. One also needs to consider its temporal nature. The dynamics of the motion patterns are in essence the possible trajectories within the enclosing shell.

1.3 Applications

The ability to visually retrieve representations for interpreting human body motions can open up many promising applications. An important domain that can

directly benefit from this is intelligent surveillance systems. Knowledge on human body motion patterns can also provide additional cues for contents of video sequences. This can be used to aid in the process of information retrieval, where image sequences can be indexed according to the motion patterns of subjects. Additionally, the representation will undoubtedly contain some information on the posture of a human body. This in turn could benefit applications such as human motion capture, novel user interfaces and telecommunications. The rest of this section will provide details on the application domains mentioned above along with descriptions of how they will benefit from this research.

1.3.1 Intelligent Surveillance Systems

The body configuration usually provides information about one's intentions. For example, two people meeting each other may both have one of their arms raised, undergoing a waving gesture. Existing systems typically use statistical models of image templates for recognising simple interactions between two individuals [63, 88]. Having a more detailed but compact representation of the body configuration can allow more detailed interactions to be modelled.

1.3.2 Human Motion Capture

At present, there are many different methods for performing motion capture. Generally, there are two main approaches to the problem. The first involves attaching electronic sensors to the joints of a subject. Examples of such system include the Flock of Birds system from Ascension or the Isotrak II and Fastrak systems from Polhemus. These sensors help by relaying back joint positions and angles. The other approach involves an optical based system where special markers are used in place of sensors. A calibrated system of multiple cameras is then used to visually track the 3-D position of these markers. Recent systems marketed by Vicon Motion Systems have provided platforms for real-time optical motion

capture. However, the cost of the system is rather high, and the required space for using them large. Both methods are, to some extent invasive, therefore requiring the subject to wear special equipment, which restricts one's possible movements. Being able to visually track the body configuration *without* the need of special attachments can lift this restriction.

1.3.3 Novel User Interfaces

Virtual reality applications can benefit directly from a user-friendly method which can recover the body configuration information of a person without requiring one to wear specific attachments. The system can be trained to recognise and interpret the different configurations of the user and take appropriate actions.

1.3.4 Low Bandwidth Telecommunications

In telecommunication applications, such as video conferencing, it is often necessary to send the entire image of a person. This can require a high network bandwidth, making its widespread use difficult. However, if the body configuration of the person can be recovered, it can be used to drive a pre-constructed virtual model of the person. Here, only the virtual model parameters need to be transmitted, which are significantly smaller in size than that of visual images. Additionally, this can be a complement to existing systems for reconstructing virtual face models [14, 87, 24] to provide a more realistic reconstruction of the subject.

1.4 Contributions

The novel contributions of this thesis are as follows:

- A method for analysing the degree of ambiguities of visual information has been developed. The configuration of a human body was depicted by a representation that combined information on the body's visual appearance and structure into a hybrid vector. The ambiguities of the hybrid vectors'

visual appearance components were quantified. This provided a relative measure of the degree of ambiguity in each hybrid vector instance. The usage of such a measure was then demonstrated in the selection of more reliable camera viewpoints.

- An example-based Linear Combinations framework was proposed for learning the kinematics of the human body. The kinematics of the human body was captured through a collection of prototypical hybrid vector examples. These prototypical examples essentially define a feature space. The examples were learnt through statistical analysis of a training set of hybrid vectors. Information on instances of novel body configurations was possible through the linear combinations of the prototypical examples. Furthermore, kinematics constraints were imposed in the form of piecewise clusters in the feature space that restricted the possible linear combinations. The parameters of the constraints were learnt using an entropic framework that was aimed at discovering the model with the simplest structure.

- The use of transition matrices for learning the dynamics of human body motion in the feature space was proposed. It was discovered that the hybrid vectors exhibited discontinuous dynamics. These discontinuities were then treated as transitions between subspaces in the feature space. A model for the subspaces was provided by the kinematics constraints. These transitions were then captured with a transition matrix that was learnt from known sequences of continuous human body movements.

- The use of a stochastic framework based on guided sampling was developed for visually tracking the motion of human bodies. The uncertain nature of the motion patterns were addressed in the use of multiple hypothesis (or samples) for predicting possible body configurations. The evolution process

of each hypothesis was guided by the learnt spatio-temporal models. Specifically, different hypotheses were generated from an example-based kinematics model. The dynamic model aided the prediction of the future state of each hypothesis (i.e. future body configurations).

1.5 Overview of the Thesis

The remaining chapters of this thesis are structured as follows. The next chapter will consider relevant and related research on various approaches for tackling the tasks defined in the previous section. The rest of the thesis will describe the research undertaken, along with the necessary experimental results, analysis and conclusions.

The definition of the representation used for modelling 3-D human body configurations is described in Chapter 3. However, in order to recover this representation from visual images, a hybrid representation that fuses observable visual cues with 3-D components of the representation is also described.

In using visual cues for inferring the representation components, the issue of ambiguities inevitably arises. One common cause of these ambiguities is the loss of depth information when projecting 3D objects onto 2D planes. This projection often reduces the robustness in visual tracking. To address this issue, we introduce a method for quantifying the ambiguities of different types of visual information in Chapter 4. Additionally, we introduce a hybrid representation for learning visual information ambiguities.

Chapter 5 introduces a method for learning the hybrid representation defined in Chapter 3. The result of this learning process includes a model for generating novel instances of the hybrid representation. Furthermore, a set of constraints is also learnt to allow only valid instances to be generated. This will be given in Chapter 6.

The model defined in Chapter 5 and Chapter 6 only provides knowledge on spatial attributes of the hybrid representation. The dynamics capturing the temporal nature of the representation are yet to be taken into account. In Chapter 7, the methods for learning and modelling the dynamics of the representation are described. A novel use of a stochastic CONDENSATION [52] framework based for utilising the learnt spatio-temporal models to track the hybrid representation using visual images from a single camera is described in Chapter 8.

Finally, a summary of the work undertaken is provided, conclusions are drawn, the limitations of the existing approach are given along with possible enhancements in Chapter 9.

Chapter 2

Background Review

2.1 Introduction

In this chapter, a review on related work is given. The contents of the rest of this chapter are divided into three sections, each representing one of the tasks that need to be undertaken. Each section will begin with the description of the issues that arises when attempting to solve its respective task. A review of existing work on tackling the task will then be given, along with the issues already resolved.

Firstly, the task of selecting a representation for learning the human body motion patterns is considered in Section 2.2. Next, existing methods on modelling these representations are presented in Section 2.3. Methods on visually tracking a representation are then reviewed in Section 2.4. Finally, a conclusion is provided in Section 2.5 where various issues related to learning the dynamics of human motion were identified. An overview of the subsequent chapters that addresses each of these issues is also given.

2.2 Human Body Representation

There are generally three considerations in selecting a representation. The first involves the contents of the representation and what it will capture. In this research, representations that are capable of capturing human motion are of particular interest. Second, the plausibility of visually recovering this representation with a

certain degree of robustness and accuracy has to be considered. That is, is there any available visual information in the input image that can allow us to reconstruct the representation contents? Are they consistent across different object instances? Are these visual information often corrupted by noise? Finally, but crucially, whether the representation is ambiguous. That is, can the representation uniquely capture each different object configuration? The issue of ambiguity also apply to visual information used to reconstruct this representation. 2-D visual information will unavoidably have certain degrees of ambiguities. This is due to the loss of depth in 2-D images of 3-D objects. However, to what degree will the visual ambiguities affect a representation? There are broadly three different kinds of representations and schemes: image based, 2-D model based and 3-D model based representations. Let us now consider in more detail each of these schemes.

2.2.1 Pixel Based Representations

One of the most straightforward methods for a representation is to use purely low-level features of an image, namely to use pixel information from the 2-D visual images of a subject. The body configuration of a subject is captured in terms of texture information present in the form of a unique ordering of pixel values (grey-scale or colour triplets). Polana and Nelson [69] described such a representation as a means of "getting the man without finding his body parts". This refers to the fact that in using the image of a subject, some of its pixels would inevitably contain information about different body parts. The direct use of the image pixel information was adopted by Darell and Pentland [80] for recognising gestures. The temporal aspects of the human motion are handled by grouping a continuous series of images into a set.

One of the motivations for adopting this low level form of representation comes from its usage in applications involving face detection [46, 74] as well as face recognition [72, 71]. However, it was noted by Oren *et al.*[64] that the use of

image based representation on human bodies differs considerably from its original usage on faces. Here, additional difficulties originate from the flexible shape of a human body. Additionally, these shapes are rarely, if ever, rectangular. Thus, a set of pixels in the image are rendered distracting, in that they do not contain any information useful for interpreting the body configuration of a subject. Moreover, different clothing worn by different individuals adds an additional variability to the colour and texture of the image and thus its pixels.

As a result, filtering methods are usually carried out to detect and remove such distracting pixels. Often, the resulting filtered image pixels will only contain information about the shape of the silhouette. Here, the colour and texture information resulting from the clothing is discarded. In a controlled environment, for example the Kids Room system[5] or the Virtual PAT system [31], a method [30] involving the use of infrared lights and cameras for illuminating and subsequently detecting the subject is adopted. However, in the more unpredictable setting of an outdoor environment, a commonly used processing method involves the subtraction of a background image followed by morphological operations for noise removal. An example of such a method being adopted can be seen in the W4 system [26]. Here, the background scene is modelled by observing an empty scene for a period. The subject in a novel image is detected by subtracting the background scene from it. A combination of thresholding and morphological operators was then used to remove spurious noise.

2.2.2 2D Model Based Representations

In filtering out the unnecessary information, only a small set of pixels will convey any meaningful information. For example, in the cases above where only pixels provide information about the shape of a subject, we find that the majority of the other pixels are often unnecessary. This implies that pixel based representations can be wasteful (i.e. the inactive pixels convey no meaningful information).

It would be advantageous to only retain and represent the active pixels' information and discard the rest. Whilst one can iteratively detect and remove such redundancies through learning (see Section 2.3), one can also make this removal explicit.

Shape Based Models

One such a model is known as shape based models with which the shape of a subject is represented as a set of vertices defining its outline. Usually, this takes the form where all the coordinates of the vertices are concatenated into a high dimensional vector. One of the earliest work that adopted this approach for tracking persons was presented by Baumberg and Hogg [3, 4]. The vertices were placed uniformly around the silhouette of the body of a subject, resulting in a representation called the Point Distributed Model (PDM). Subsequently, these vertices were more efficiently estimated using B-spline curves [33]. Ultimately, the shape is represented as a set of B-spline control points. Similar use of this representation was presented by Xu *et al.*[47] and Sulivan *et al.*[43] for tracking human motion. Recently, shape models were also used by Broggi *et al.* [6] for detecting pedestrians. A closely related use of the PDM model was adopted by Cootes [77] and Heap [81] for modelling and tracking the shape of hands. This form of representation was also used to track the shapes of the shoulders and head by MacCormick [34].

In using shape to track a subject, at any point in time, a vertex on the shape vector will correspond to a certain part of the body. However, as with the PDM approach by Heap[81], such a vertex-body correspondence is inconsistent across the different configurations of a 3-D object. This phenomenon arises from the fact that the vertices usually defined evenly along the curve. Therefore, their positions are dependent on the length of the silhouette of an object at a certain configurations. This in turn introduces yet another degree of variation and sometimes

discontinuity as the shape vertices "slide around" the shape of the object as its configuration changes, making the learning process for this form of representation more complex [8]. A solution to the problem is to establish a consistent correspondence between the vertices and object parts (or body parts in particular) across a range of different configurations. However, it was noted by Cootes [77] that establishing such correspondences is not always possible due to self occlusion. In effect, establishing a consistent vertex-object correspondence in a shape with N number of vertices would be equivalent to tracking N parts of the object. Instead, a more viable approach would be to track a small number of body parts positions.

Body Parts Positions

It has been known in psychology from Johansson's study of moving lights display (MLDs) that it is possible to extract meaningful interpretations about a subject simply by observing a set of points attached to one's body parts [22, 23]. An interpretation for the findings by Johansson suggests the possibility of the human brain interpreting the intentions of a subject solely by recognising one's body parts motion patterns. This suggests that, it is not necessary to recover the 3D structural information of the human body, or that of the body parts in order to model and interpret human body motions effectively. The following will provide a survey of the representations aimed for such an interpretation. The remaining interpretation follows the argument that the 3-D positions of body parts are initially recovered from the motions of the markers. Using these 3-D body parts positions, one can then recognise the actions of a subject. We review representations that follow such an argument in Section 2.2.3.

The work by Johansson was approached purely from a psychological point of view. As such, a specialised setup where bright spots were attached to the joints of an actor dressed in black moving against a dark background was used. However, computer vision systems have to deal with a more unrestricted environment where

specialised markers on body joints may not be available and desirable.

Recent advances in colour tracking [75] has allowed for fast and robust tracking of certain body parts. For example, Wren *et al.* explored the notion of tracking body parts as blob features in the Pfinder system [13]. A subject was located using background subtraction. A probabilistic model for a subject's different parts' colours was then estimated and used to extract the positions of the body parts in subsequent images. Such an approach was extended by Sherrah and Gong [42, 41] in using Bayesian networks to deduce the positions of the hands and head by fusing colour, motion and coarse intensity measurements along with contextual semantics.

2.2.3 3-D model Based Representations

2-D based methods can be ambiguous. Objects at different 3-D configurations can sometimes yield similar 2-D model instantiations. Additionally, self-occlusion can remove the availability of the visual information of different object parts at different configuratioins. Moreover, these representations are 2-D *projections*, therefore changes in viewpoints would contribute yet another factor of variations in the representation contents. To overcome these difficulties, some work has instead used three dimensional object representations.

3-D Body Parts Positions

Let us now review a representation model which adopts the second standpoint of the results by Johansson in the MLD system [22, 23]. That is, one's ability to interpret the visual motions of moving human body parts (e.g. joint positions) requires an initial recovery of some three dimensional structure.

One representation that contains such 3-D structural information is the extension of the original 2-D MLD to a full 3-D version by Jenkin[53]. In this model, the 3-D location of the body parts was located using stereo vision. Similarly,

Azarbayejani *et al.* extended the Pfinder system into the Spfinder system to recover three dimensional positions of the hands and heads by using stereo camera setups [1, 2].

However, the sole use of three dimensional points or blobs does not fully take the advantage of available knowledge of the articulation constraints of a human body [15]. In fact, one can directly incorporate such prior knowledge about body articulation constraints.

Exploiting Known Structure: Internal 3D Skeleton

The simplest method for exploiting our knowledge about the human body is to define a representation of the underlying 3D skeleton. As with the previous body parts representation, 3D vertices are used for representing the positions of various body parts (usually joints and body endpoints). These vertices are then connected with other vertices in a manner which resembles the way that the joints and endpoints of a human body is connected to each other. One can then think of a pair of vertices which are connected to each other as the bone of a body part. A further constraint can be imposed by requiring the length of the bone to be constant.

Optical motion capture systems typically adopt such a representation for modelling the human body configuration [59]. The body joint positions are indicated by specialised markers and its 3-D co-ordinates recovered using a multiple camera system. Work on the use of 3D skeletons in less restricted setups where no markers are used was carried out by Moeslund and Granum [84]. With this approach, the arm was represented by a 3-D skeleton as described above. A more complete model of the entire body's skeleton was applied and visually tracked by Leventon [56].

The sole use of 3-D skeletons is not common due to the fact that they only resemble stick figures, albeit in three dimensions. This makes the 3-D skeletons

difficult to verify against available visual information, owing to the vast dissimilarity between 2-D projections of 3-D stick figures and visual images of human figures. As a result, a method for tackling this problem involves the attachments of 3-D objects onto the bones of the skeletons.

Visually Realistic 3-D Models

In order to attach objects to each bone, one first associates each bone with a co-ordinate system. Thus, the entire body can be represented by a set of underlying co-ordinate systems. Furthermore, these co-ordinate systems are usually hierarchically arranged to have the same structure as the human body. That is, a 3-D representation of the human body would then consist of a set of co-ordinate systems for capturing the orientation and position of each limb.

While the co-ordinate system gives the information on the location and orientation of the limb, no information about the shape of the corresponding object part is given. This is where the difference in various existing 3D representations of objects lies. Each type of 3-D representation usually has its own set of objects attached to each coordinate system. These objects would in turn have their own set of parameters (e.g. length and size).

One may start by attaching primitive shapes onto each limb's coordinate system. For example, O'Rourke and Badler [40] chose to model the body parts with a set of overlapping sphere primitives. Alternatively, Hogg [17] chose to replace the overlapping sphere primitives with simpler elliptical cylinders, in order to reduce the number of parameters for each body part. The model was then used for tracking a walking person. Recent work by Sidenbladh *et al.* [25] and Brand [51] also used 3-D cylindrical models for representing body parts of human subjects. A similar representation was used by Regh [38] for tracking the 3-D model of hands. However, the shape of the cylinder may be too rigid and specific to model body parts.

A more flexible primitive in the form of tapered super-quadrics was proposed by Gavrila and Davis [16] for tracking 3-D human models. Work by Kakadiaris and Metaxas [27] extended the use of super-quadrics into providing a means of generating a more realistic synthesis of the human body. This was achieved by modelling the physics of the body parts which was then used for recovering the parameters of deform-able models for each limb from noisy data [20].

However, the task of synthesising a realistic image of the human body is a complex task. One needs to accurately account for the different geometrical deformations each body part can undergo. Subsequently, these deformations depend on the physical interaction between the muscles and bones of a body part. In order to reconstruct each body parts' visual appearance these interactions would have to be simulated. On the other hand, such "realistic images" of a human body are already available to computer vision systems. These take the form of the 2-D visual images acquired from a camera. Therefore, one would ask, if it is feasible to employ the simpler model of a 3-D skeleton that captures the underlying dynamics of the human body, while simultaneously exploiting the available visual information of the human body from the input images. One possible approach for this lies in combining the *visually observed* information with the underlying 3D skeleton information in order to form a unified *hybrid representation.*

2.2.4 Hybrid Representations

The hybrid representation derives its name from the fact that different types of information are combined together into a single vectorised form. As noted above, this would firstly allow one to have 3-D structural information (e.g. 3D skeleton) of a human body. Secondly, the task of synthesising a realistic image of the human body is bypassed by exploiting the available visual information (e.g. shape information) from the images of a human body. However, such a form of representation for human bodies is still relatively new. To date, only

Bowden *et al.* [67] has used such a method for representing and tracking the upper torso of the human body. A greater exploration into the various aspects and characteristics of such a representation will be given in Chapter 3. There we will see how the combinations of different forms of information into a unified hybrid vector provides a representation which will allow us to extract hidden 3D skeletons of human bodies using available visual contours and body parts positions from image sequences.

2.3 Learning the Representation

After having selected the appropriate representation, the next task is to mathematically model the representation. To this end, two aspects need to be considered: spatial and temporal models of the representation.

2.3.1 Spatially modelling a representation

Any representation would consist of a number of parameters (e.g. 3-D joint angles), essentially making it a high dimensional vector. The space in which these high dimensional vectors exist is defined as a parameter space. In order to capture all different possible configurations of this representation, a model would have to capture the regions occupied by the representation in the parameter space. In attempting to do this, the following issues need to be considered:

- Can the model cope with the complexity of the representation?
- Is it possible to impose constraints on the model such that unrealistic instances of the representation will not be constructed?

In most of the existing approaches where a 3-D model was used, the spatial model for the representation is the representation itself [17, 16, 66, 61, 18, 38, 25]. For the purpose of visually tracking or recognising the 3-D object, given the underlying 3-D model parameters, the visual appearance of the model can be

reconstructed using computer graphics techniques. This visual reconstruction can then be compared against the visual information of an input image to measure the accuracy of our model in representing the configuration of the real world object.

Instead of having an explicit 3-D model and using computer graphics to synthesise the model at different poses or configurations, example based methods provide an attractive alternative. Ullman and Basri [76] suggested to select a subset of the training examples and linearly combining them to generate novel representation instances. However, this was only done for rigid 3-D objects. It is not clear how many and what examples are needed for a more complex articulated object of flexible shapes. Furthermore, to guarantee that only valid examples are generated, constraints on the linear combinations of examples were explicitly defined through analytically derived equations. However, it will be shown that in Chapter 6, this becomes an intractable task when the object has more complex dynamics (e.g. a highly articulated object).

Alternatively, Baumberg and Hogg [3] used Principal Component Analysis (PCA) to statistically model the variations of the contours of walking people. Each contour is represented by a fixed number of 2-D vertices and is then considered as a high dimensional vector, constructed by concatenating all its 2-D vertices together. To perform PCA, a set of *representation eigenvectors*, which inherently captures the subspace spanned by the training examples, is obtained. This subspace is usually defined as the *eigenspace*. These eigenvectors can then be linearly combined to generate novel contours of walking people. Interestingly, we will show in Chapter 5 that PCA can be a way of obtaining the sufficient examples for linear combinations introduced by Ullman and Basri[76].

However, in the case of a more complex representation, PCA alone is insufficient to accurately capture all the characteristics of a representation. This is especially true if the representation spans a non-linear high dimensional space. This can cause certain combinations of the eigenvectors to yield invalid representation

examples. To address this, constraints can be placed to restrict the combinations, as proposed by Heap and Hogg[83] for learning 2-D hand shapes. This was then extended by Bowden *et al.* [67] for learning a hybrid representation of 2-D and 3-D cues for tracking 3-D human body skeletons. We will show in Chapter 6 that the method for obtaining these constraints is also a general method for learning the constraints for the linear combinations method by Ullman and Basri[76].

A recent approach described by Rosales and Sclaroff [70] uses a multi-layer perceptron neural network to learn the mapping of 2-D visual cues to the corresponding 3-D skeletons. The multi-layer perceptron architecture was chosen for its ability to capture non-linear variations in the training data.

2.3.2 Temporal Dynamics of Human Body Motion Patterns

Since the representation concerned was geared towards learning human motion patterns, it will inherently be dynamical in nature. These dynamics possibly include discontinuities along with other forms of nonlinearities. Therefore, it would be advantageous to have a model for the dynamics of the representation. This could later be used as constraints for the tracking process.

One of the simplest method for modelling the motions of a human body is to use a passive physics-based model. This usually takes the form of a dynamical model that constrains the variations of each parameter of the human model. For example, Deutscher *et al.* [32] modelled the dynamics of a 3-D human body representation as a critically damped second order Gaussian linear model. In other words, the displacements of the 3-D representation components are obtained by sampling from a second order Gaussian distribution. These displacements were then used in a stochastic framework for generating multiple hypotheses of the 3-D representations. However, such models do not directly exploit known and available patterns in the motions of a human subject.

An interesting alternative was described by Iwai *et al.* [89], where a collection of known representation trajectories is used to model its dynamics. With this approach, the representation consists of the 3-D parameters of a 3-D human model. The current configuration of the model is then used to index the appropriate preset representation trajectory. This is then used to predict the parameters of the 3D human model. One concern with this method lies when the model undertakes novel trajectories.

Example based methods are usually adopted for capturing the temporal dynamics of human visual appearances in image based representations. This in turn usually takes the form of temporal templates. Davis and Bobick [29] proposed such a method by encoding history information into the value of the pixel. This was done by firstly extracting motion regions of an image sequence resulting in a *motion-energy* image (MEI). Following this, a *motion-history* image (MHI) is constructed whereby the motion recency at a point is set at its corresponding pixel. This results in a pair of images which both captures the spatial information of a body motion while simultaneously its temporal signature in the MHI. Therefore, to encode a set of different body motions, a set of such MEI/MHI templates (one for each motion pattern) was generated and stored. However, a disadvantage in these methods arises from the shortcomings of the representation itself. As the templates are two dimensional motion images, they too are susceptible to self occlusions. When this happens, the motion information of the occluded part will be unavailable. Additionally, only the motion information captured by each template is view dependent. Should the subject be viewed at a different angle, new templates would need to be acquired.

Another method was proposed by Heap [81] whereby the dynamics of the tracked representation was modelled as a transition matrix. To obtain the transition matrix, a set of training examples was first partitioned into different sets using a fixed number of clusters. The transition matrix was then built by observing the

cluster transitions of individual examples in training sequences. However, this was only used to represent the dynamics of a single type of information, namely the dynamics of the shapes of a hand. We will explore further the use of this method for modelling the visual and underlying dynamics of a human body in Chapter 7.

2.4 Visually Tracking the Representation

With a model for both the spatial and temporal aspects of a representation, the next task is use the model for visual tracking. In doing so, a number of issues have to be dealt with. The first involves having to cope with possible non-deterministic and non-linear dynamics in tracking a human body. The non-linear aspect can be addressed to a certain extent by the model itself. However, in the case of non-deterministic representation over time, the model can only provide choices as to possible valid configurations of the representation.

The next issue involves the ambiguities in the representation and the model. For example in tracking 3D skeletons, there are some body configurations that are ambiguous, since different 3D skeletons can give similar 2D visual projections.

The visual tracking of an articulated object can be considered as a form of a model fitting task. Computationally, this is equivalent to a cost function minimisation problem. The cost function to be minimised is usually defined by a matching function between the representation states with the visual features in an image. Thus, an important element of visually tracking an object lies in the minimisation of this matching function.

In the case where the representation is a 3D virtual model of the object of interest, the matching function will measure the discrepancy between the projections of 3D model against different visual information extracted from the input image. Following this, there exist several different methods for minimising the matching function. This usually involves adjusting the 3D parameters of the vir-

tual model. For example, Regh [38] minimises the matching function numerically using the Gauss-Newton method. Additionally, motion constraints on the model parameters can be used to further guide the optimisation process as described by Yamamoto *et al.* [61, 62]. Alternatively, the minimisation problem can be solved by a high dimensional search method as described by Gavrila and Davis [16]. In attempting to address the problem of ambiguities, all these methods employ a multi-camera setup. In addition, they all require the cameras to be calibrated.

Another method for tracking articulated object uses learnt statistical models of the representation. An example of this is described by Baumberg *et al.*[3]. Here, 2D contour models of people walking are learnt statistically using PCA. The PCA model was then used to reconstruct novel instances of the 2D contour models for matching against image features of walking people.

Recently, stochastic frameworks were also used for visually tracking articulated objects. For example, Heap [81] used the CONDENSATION framework for tracking 2D hand shapes. Alternatively, a similar framework was used by Deutscher *et al.*[32] and Sidenbladh *et.al* [25] for tracking the 3D parameters of human models. However, such methods were only used to track one form of information, the shape of a hand (as with Heap) or the underlying parameters of the human subject (as with Sidenbladh and Deutscher). We exploit this approach in Chapter 8 to robustly track both the visual and underlying parameters of a human subject.

2.5 Conclusions

In this chapter, existing work on issues raised whilst attempting to learn human body motion patterns were reviewed. To sum up, three issues were identified, each of which will be given further consideration in the following chapters:

- Selecting a representation and analysing its suitability for modelling human body motion patterns (Chapters 3 and 4).

- Learning the spatial constraints of the human body via such a representation (Chapters 5, 6 and 7).
- Visually tracking the human body motion patterns (Chapter 8).

Human Body Configuration Representation

In Section 2.2.2 it was found that two dimensional representations (e.g., shape and body parts positions) provides measurable information. However, one shortcoming of such a form comes from the fact that visual projections of three dimensional objects are ambiguous. Conversely, three dimensional representations (e.g. 3-D human body models) do not suffer from such ambiguities. However, there is the disadvantage that such a form cannot be directly verified against available visual images that are only two dimensional. In order to address such a shortcoming, one needs to construct a virtual model that can synthesise a realistic visual projection of the human body. To achieve such a goal, one requires the modelling of both the underlying structure (3D skeleton) and its surrounding flesh. This may not be a trivial task due to the fact that the flesh surrounding the underlying skeleton is fairly deform-able in nature, adding many parameters to the model. Additionally, one notes that in a vision system, information on the visual projections of a human subject are already available. Therefore, one would suspect that a more superior representation would be the one that can exploit the advantages of *both* the visual and underlying structure aspects of a human body. We explore this in greater depth in the next chapter where both visual and underlying structural information are unified into a single representation. Additionally, we will also see how the ambiguities of two dimensional information affects the uncertainty of three dimensional information when they are both unified together in Chapter 4.

Learning the Representation

In Section 2.3, the issues regarding the learning of motion patterns of a human body using a selected representation were identified. These issues can be grouped into two major areas:

- Learning the spatial constraints of the human body
- Learning the dynamics of the body motions

As the human body is a highly articulated object, it posses a vast amount of possible configurations and a complicated constraint surface for the underlying structure parameters. Additionally, the resulting visual information too contributes to other forms of variations. Thus, analytically modelling such spatial constraints for the visual and underlying information of a human body may be very complex. As a result, a simpler alternative would be to *learn* such constraint information from available example data of a human body at various configurations. We show in Chapter 5 and Chapter 6 how one can go about achieving such a learning task using an example-based method called linear combinations of examples.

Information on the spatial constraints of a human body only provides knowledge on which body configurations are valid and which are not. However, it does not contain any form of information on how different parameters of a human body changes. In other words, it does not contain the temporal information on the motion patterns of human bodies. One finds that the motion patterns of human bodies are governed by a subject's intentions. Such a trait introduces a great uncertainty into any process that aims to predict future configurations of a human body. However, this does not happen in all cases. There are similarities in methods for performing certain tasks. That is, different subjects move their bodies in similar fashions to achieve and identical goal. Furthermore, constraints are also imposed by the muscular mechanisms that allow movements of the body.

In Chapter 7, we will see how the unpredictable nature of human motion patterns can be tackled using a passive dynamics model. On the other hand, the predictable nature is exploited by using models of previously observed human body motion patterns.

Visual Tracking using the Representation

In order to gather more information of human motion patterns, it would be advantageous to have a mechanism whereby one can efficiently gather useful observations. In a vision system, this requires the ability to visually track the representation of the human body. It was found in the process of identifying relevant issues for learning human motion patterns that a number of factors have to be considered:

- Constraints to allow the representation to represent only realistic human body configurations.
- Nonlinearity in the variations of the human body representation components.
- Ambiguities in using visual information.
- Robustness in identifying the actual body configuration, given visual images which may contain corrupting noise.

The first factor on tracking plausible body configurations is resolved using a learnt spatial-constraint model. The next factor of nonlinear dynamics is addressed using a passive dynamics model. The third factor which involves the ambiguities in visual information is tackled using a framework for identifying the degrees of ambiguities of visual information given in Chapter 4. We will see in Chapter 8 how these learnt spatial and temporal models along with the ambiguity framework can be simultaneously exploited in a probabilistic framework for visually tracking a human subject.

Chapter 3

Representing Human Body Configurations

To model the human body motion patterns, we need a suitable computational representation for the human body configurations. To this end, this chapter will define the information available for learning about possible motion patterns of a human body. We begin in Section 3.1 by describing a representation that will be used to capture the body configurations of a human. This representation will take the form of a 3-D skeleton and we discuss different methods for representing its parameters. Also, since this 3-D form is not visible in 2-D images, we use observable visual information of the human body to indirectly infer the underlying 3-D skeleton parameters. Section 3.2 will then introduce and define available visual information that can be separated into two types: positions of the body parts and their shape. As a result, we have three different *modalities* (3-D skeleton parameters, body part positions and shape). Additionally, one can categorise these three modalities into *hidden information* (3-D skeleton) and *observable information* (body part positions and shape).

Furthermore, we describe in Section 3.3 how the availability of observable variables can be used to infer possible values of the corresponding hidden variables. Computationally, we adopted a hybrid vector form introduced by Bowden[67], where observable information and hidden information are combined into a single

representation. We will also see how hybrid vectors allow one to exploit the correlations between the different modalities (hidden and observable) for inferring hidden information. Following this, a description of a training set of different hybrid vectors and its acquisition system and process is described in Section 3.4. We then discuss the characteristics of such a representation when applied to motion patterns of human bodies in Section 3.5 and conclude this chapter in Section 3.6.

3.1 Capturing the Underlying Body Configuration Information

In order to capture information on the human body configuration, one requires a representation that can capture the variations exhibited by the individual body parts. A straightforward method of tackling such a task is to adopt a model similar to the underlying structure of a human body. For this reason, a 3-D skeleton model is adopted. A 3-D skeleton is modelled as a set of hierarchically linked rigid bones (i.e. fixed length). For example, in the skeleton of a human body, the lower arm is hierarchically linked to the upper arm. Any changes to the orientation or position of the upper arm affects the lower arm as well. A bone on the 3-D skeleton can also be thought of as a body part of a human body. Therefore, the parameters of the bones capture the information of the individual body parts. Two such means of representing 3-D skeleton bone parameters will be described next. The first representation uses the end points of bones in the form of 3-D vertices. The second uses the angles of the bones relative to a fixed plane in 3-D space.

3-D Vertices

Formally, the number of joints in a skeleton is defined to be (N_T). A single parameter of the skeleton is then defined by its joint position in 3-D space (x, y, z). We then represent a 3-D skeleton's configuration as a vector of (N_T) joint positions

concatenated together, ($\mathbf{V_B} = x_1, y_1, z_1, x_2, y_2, z_2, ..., x_{N_T}, y_{N_T}, z_{N_T}$) An example of this representing the 3-D skeleton of the upper torso of a human body can be seen in Figure 3.1.

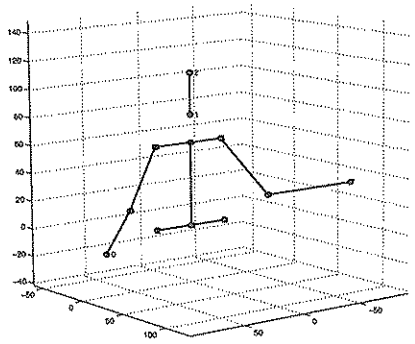


Figure 3.1: Skeleton 3-D vertices.

3-D Joint Angles

An alternative to using the 3-D positions of skeleton joints is to use the spherical angles of the joints with respect to the $X - Y$ plane. One advantage of using the joint angles lies in the availability of results from extensive studies in physical medicine [58]. Additionally, it does not require one to model the variations due to the size and lengths of the bones. Consequently, the representation will have simpler dynamics. The 3-D skeleton's configuration is then represented as a vector of (N_T) bone joint angles concatenated together, $\mathbf{v}_B = (\theta_1, \phi_1, \theta_2, \phi_2, ..., \theta_{N_T}, \phi_{N_T})$. An illustration of the bone joint angles can be seen in Figure 3.2.

3-D Skeletons Cannot be Observed Directly

Both of these representations suffer from the disadvantage of not being able to be measured directly from visual images. In other words, the 3-D skeleton parameters can only be inferred. Visual cues that can be directly and reliably measured from an input image are used to indirectly recover the 3-D skeleton information.

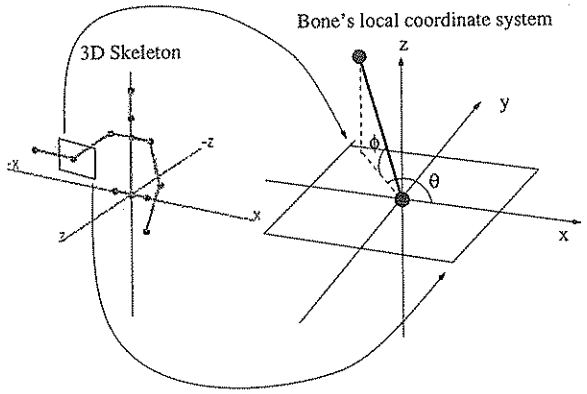


Figure 3.2: Skeleton joint angles.

3.2 Visual Observations of the Human Body

This section will describe two forms of 2-D visual information that can be used to retrieve the body configuration. In particular, the positions of various parts of the object were first used. With a 3-D skeleton model, one would know the connectivity between the vertices (i.e. bone endpoints) and the 3-D lengths (bone length) between each pair of connected vertices. Using them, one can recover the 3-D co-ordinates of the skeleton vertices in conjunction with the 2-D positions of different body part joints. One assumes that the 3-D length between two connected vertices does not change. Since the x and y co-ordinates are already known, one can use the distance formula between the two vertices to recover the depth (z) co-ordinate. However, this is based on the assumption that it is possible to reliably track all different joints of a subject. In reality, only a small subset of the required body parts can be tracked in a reliable manner, which include the hands and head of a subject. Using such a reduced set of body parts would result in ambiguous estimations of the 3-D skeleton parameters (see Chapter 4). By adding shape information for capturing the overall configuration of the subject, as shown in Chapter 4 such ambiguities will be decreased. Let us now give the formal definitions of the body parts positions and shape.

3.2.1 Spatial Information: Body Parts Positions

A number (N_P) of trackable object parts with a vector of 2-D positions is defined as, $\mathbf{v}_P = (x_1, y_1, ..., x_{N_P}, y_{N_P})$. For example, in a tracking human body, three parts can be tracked, the head and the two hands. Here, N_P is three. The first two components of $\mathbf{v}_P$ would be the position of the head. The second and third components the position of the left hand and finally the last two components represent the position of the right hand.

3.2.2 Shape Information: Body Silhouette

The contour of a person's silhouette is represented by a Point Distribution Model (PDM) commonly used for modelling and tracking 2-D shapes [77]. It consists of a number (N_S) of 2-D vertices, $\mathbf{v}_S = (x_1, y_1, ..., x_{N_S}, y_{N_S})$, distributed evenly across the entire contour. An example of a contour of the human body upper torso can be seen in Figure 3.3 illustrating how the contour vertices are evenly spaced.

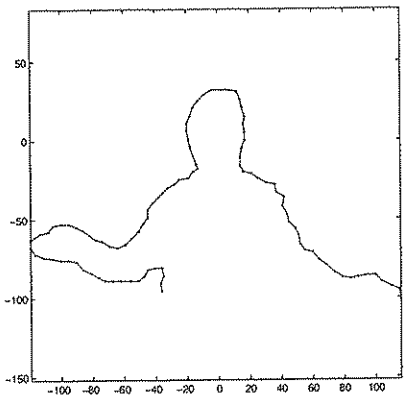


Figure 3.3: An observable 2-D contour. The contour is centred at the head's centre. The horizontal and vertical axes represent the x and y pixel co-ordinates respectively.

3.3 Unifying Visual and Hidden Information: A Hybrid Vector

Intuitively, one would expect there to be a "common-ground" between the visual appearance of the human body and its underlying 3-D skeleton structure. After all, changes in the underlying skeletal parameters would inevitably cause certain changes in the body's visual appearance. Such appearance changes are captured by its visual image. If one could learn the correlation between the dynamics of the underlying 3-D structure and its 2-D visual appearance, this would provide a possible mechanism that allows us to infer the underlying 3-D body skeleton from the observable visual cues.

3.3.1 Advantages of Observable-Hidden Variable Correlations

Let us now illustrate how the correlation between different modalities can lead to a modality that can be used to disambiguate the possible values for the other modalities.

Initially, consider the case where we have one hidden variable that can take on any real value, but no observable variables are available. Furthermore, prior knowledge about its occurrence in the form of a probability distribution is available. In Figure 3.4, we show a simulated probability distribution for a single hidden variable. Without any other information available, this distribution would be the only information that can be used for predicting the possible values of the hidden variable. However, given a new observable variable that has some form of correlation with the hidden variable, a constraint surface capturing the correlation between the hidden and observable variables can be constructed (see the Figure 3.5a). Note that after knowing a value for the observation variable, we now have a more precise probability distribution for the possible hidden variable values (see Figure 3.5b).

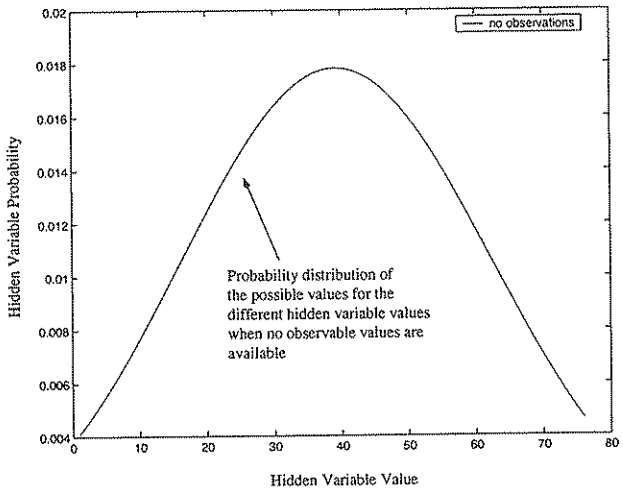


Figure 3.4: A probability distribution of a simulated hidden variable. The horizontal axis shows the value of the hidden variable. The vertical axis shows the probability of the hidden variable taking a value. Without any other measurements, at any point in time, this graph is all we have for inferring the value of the hidden variable.

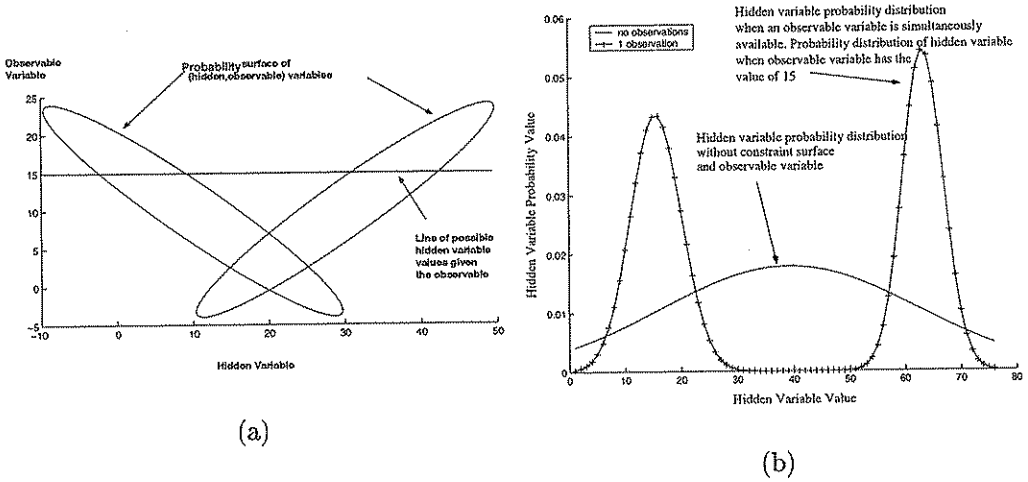
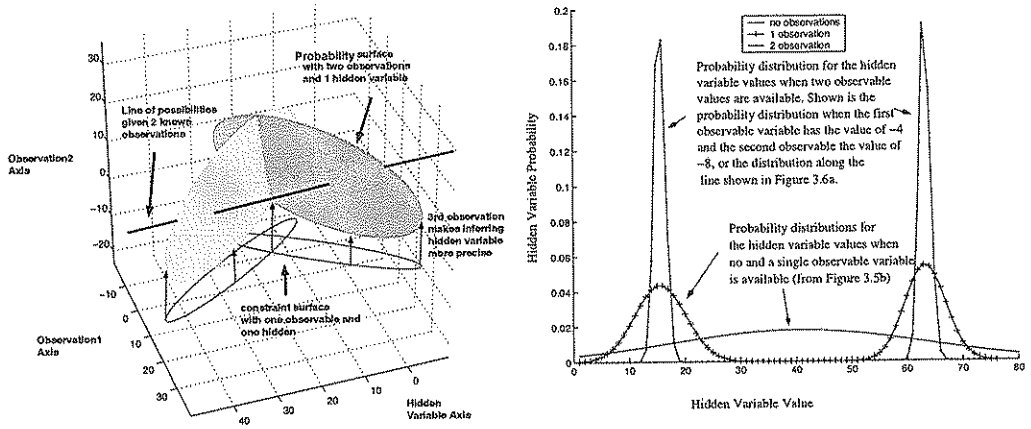


Figure 3.5: The probability of the simulated hidden variable becomes more precise when an additional correlated observation is available. We define the hidden-observable variable probability distribution to have the shape indicated with ellipses (iso-contours for probability of 0.3) in (a). The hidden variable here has the same characteristics to that in Figure 3.4, that is, if we were to marginalise across the different observable variables, we would get the probability distribution shown in Figure 3.4. We can build a more precise hidden variable probability distribution, as shown in (b), given a value for the observation variable (for example 15, as shown in (a)). Also shown is the hidden variable probability graph from Figure 3.4 in (b) for comparison with the case where no observable variables were available.



(a) (b)

Figure 3.6: The probability of possible hidden variable values is further made more precise when the number of types of observations is increased. Following Figure 3.5, another observable variable was added. The probability distribution is again indicated using the iso-contour ellipses seen in (a). Similarly with Figure 3.5, a constraint surface can be built by defining valid hidden-observation triplets to only those within the ellipses. When two observable variables are available, a probability distribution (shown in (b)) for the possible hidden variables can be built along the line shown in (a).

Furthermore, suppose we add in another observable variable, one can also unify it with the already formed hidden-observable variable. Similarly, a constraint surface capturing the correlation between the two observable and hidden variables can be formed (see Figure 3.6a). This would allow us to exploit any new correlation between the new observable variable and the hidden variable to provide an inference with greater precision, as can be seen in the probability distribution graph in Figure 3.6b. Here, a comparison is made between the three probability distributions; without observables, with one observable and two observables. We find that the high probability peaks are narrower and thus it is clearer what may be the possible values for the hidden variable. However, since the number of probability peaks are more than one, it is unclear as to which represents the correct value for the hidden variable, hence making the distribution *ambiguous*.

In a more complex context where the observable variables are made of the

components of the shape and body part position modalities, while the hidden variables are the 3-D skeleton components, one will find that similar ambiguities exists, as will be shown in Chapter 4 . For example, there may be many different 3-D skeleton models that are associated with a single shape and body part positions. In other words, there are some visual information which cannot provide us with a precise inference of the underlying 3-D skeleton (these visual data are *ambiguous* when used to infer hidden underlying parameters). One of the causes of such a phenomenon is the lack of depth information. Consequently, multiple body configurations with different underlying skeletons can result in the same contour and body parts positions. Further investigation of the ambiguity issue and means of addressing this problem is given in Chapter 4.

3.3.2 Hybrid Vector Definition

Both the 3-D skeletal information and its corresponding visual cues can be fused by combining them into a unified *hybrid-vector* representation. Given a vector ($\mathbf{v}_B$) representing an object's underlying 3-D skeleton, the object's 2-D information represented by its contour ($\mathbf{v}_S$) and when possible, positions of its different parts ($\mathbf{v}_P$), the skeleton-hybrid-vector representation can then be defined as the concatenation of all 3 vectors; $\mathbf{h}_S = (\mathbf{v}_S, \mathbf{v}_P, \mathbf{v}_B)$, as shown in Figure 3.7.

This brings about the issue of how to model this set of information-fused data based on learning. Tackling such a learning task requires one to understand and subsequently cope with the underlying dynamic mechanisms of the representation. More precisely, there would be a need to cope with the dynamics of the human body and its visual projections. The mechanisms for performing such a learning task will be detailed in Chapters 5 and 6. However, we will first study into the characteristics of this representation across a range of different human body configurations.

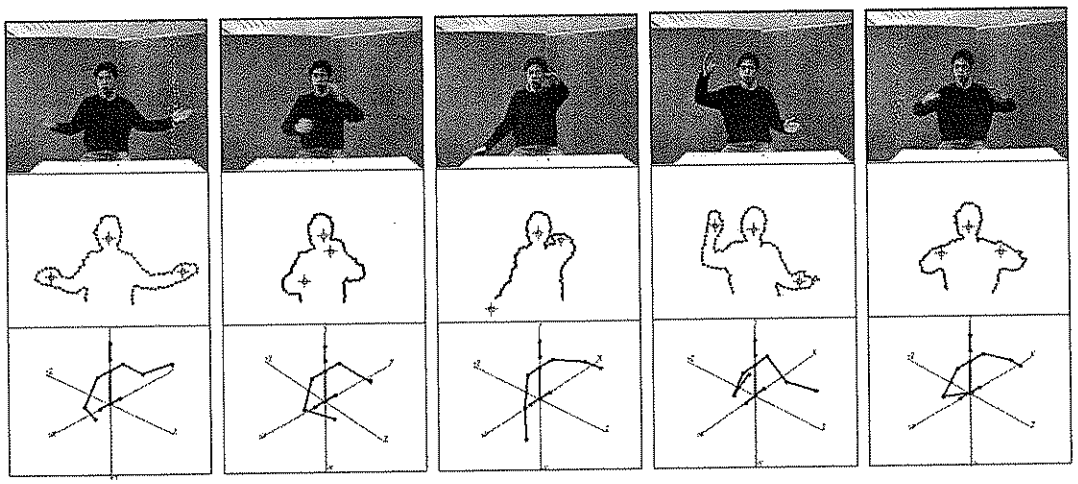


Figure 3.7: Different instances of the hybrid vector for the human body is illustrated here. The top row corresponds to the input images. The middle row corresponds to the contours, $\mathbf{v}_S$, and body parts positions, $\mathbf{v}_P$. The bottom row shows the corresponding skeleton, $\mathbf{v}_B$.

3.4 Acquiring the Hybrid Vector Training Set

For studying the characteristics of the human body configurations, the underlying body configuration of the subject was represented by a skeleton that consisted of 13 3-D vertices so that, the dimensionality of the skeleton vector is 36 as shown in Figure 3.1.

For the shape vector, a contour where 100 points distributed evenly across the silhouette of the subject was selected, resulting in a dimensionality contribution of 200 to the hybrid vector.

Finally, the positions of the left and right hands were chosen from the body parts. The reason for using the hands as opposed to other parts of the body is that, in most cases, we can have consistently extractable hands' positions by tracking the colour of skin in the image. We can benefit from the fact that there exists reliable and robust methods for tracking colour information [75]. The hands' position data contributes to a four dimensional sub-vector. In all, the resulting hybrid vector, consisting of shape, hand positions and skeleton vectors, has a dimensionality of 240.

In order to acquire a training set of hybrid vectors for a broad range of different body configurations, a Pentium 200 system configured with a Matrox Meteor frame-grabber was used. The training images were captured at a resolution of 320 by 240 pixels. Training image sequences at a fairly high frame-rate (20-25fps) by saving all the images initially to memory instead of directly to a disk store. Instead, the acquired images were saved to disk storage only after the entire motion sequence was performed. However, the capture frame-rate was also dependent somewhat on other factors, for example, the operating system and processing loads due to other users. Consequently, some frames in a motion sequence may have been skipped unintentionally during the acquisition process. We will see in Chapter 7 how such missing frames can affect our methods and experiments.

Using such a system, a total of 20 different body motion colour sequences were recorded. Additionally, a short colour sequence of the background scene without the subject being present was also acquired for each motion sequence. This resulted in a total of 1021 image frames of a subject with different body configurations. From each frame, the hybrid vector components were extracted with the following procedures:

3.4.1 Extracting the Skeleton 3-D Vertices

In order to determine the 3-D vertices of the skeleton, the length of the bones of each body part in the pixel metric was initially determined. This was done by requiring the subject to assume a body configuration where all ones body parts are oriented to be parallel to the image plane (e.g. the starfish pose [13]). This was performed for each motion sequence. Following this, for each of the motion sequence frames, the 2-D positions of the joints were located manually. Using the manually located joints' 2-D positions, and the length of the bones which links them, the remaining depth value of the joint was retrieved using the Euclidean distance formula.

3.4.2 Extracting the Visual Information (Contour and Body Parts)

In order to extract both the contour and body parts positions, the background pixels were initially removed using background subtraction. To achieve this, a simple background model was first built by determining the maximum and minimum (red,green,blue) values for each background image pixel. Thus, for each background pixel ($\mathbf{b}_{x,y}$) at position (x, y) would contain a range of red, green and blue values, whose maximum and minimum values are defined as $red^{max}_{x,y}, red^{min}_{x,y}, blue^{max}_{x,y}, blue^{min}_{x,y}, green^{max}_{x,y}$ and $green^{min}_{x,y}$ respectively.

Next, the non-background pixels were roughly located by determining if a pixel on the image has red, green or blue components that lie outside the corresponding background pixel's colour range. Formally, an image pixel ($\mathbf{img}_{x,y}$) with colour components $r_{x,y}, g_{x,y}, b_{x,y}$, for the red, green and blue values respectively, is a foreground pixel if it violates any of the following constraints:

$$red^{max}_{x,y} \leq r_{x,y} \leq red^{min}_{x,y} \tag{3.1}$$

$$green^{max}_{x,y} \leq g_{x,y} \leq green^{min}_{x,y} \tag{3.2}$$

$$blue^{max}_{x,y} \leq b_{x,y} \leq blue^{min}_{x,y} \tag{3.3}$$

Following this, a dilation operation [60] is performed, whereby, if the surrounding 3x3 neighbourhood of a foreground pixel are not all foreground pixels, it is removed.

Extracting the Body Parts Positions

To obtain the hands and head positions, a colour classification method using Gaussian mixture models for skin colour was used [75]. This allowed us to identify the foreground pixels that were skin coloured. Following this, K-means clustering [10] was performed on the skin colour foreground pixels to determine the centres of three skin coloured blobs.

Extracting the Contour

To obtain the contour, a binary image containing pixels lying on the subject's silhouette edge was first extracted. This was achieved by subtracting a dilated foreground pixel image from the original foreground image [60]. Next, the inner boundary tracing algorithm [60] was used to extract an ordered list of pixels that traced out a curve originating from the lower left extreme to the lower right extreme of the subject's silhouette. To acquire a set of contours with a consistent number of components, only a predefined number (N) of 2D points were chosen from the extracted ordered list of contour pixels. Additionally, these N points were spreaded out across the contour by evenly sampling from the ordered list. Specifically, every pixel at the list position that was a multiple of ($N/100$) would be chosen. For the purpose of our experiments, the value of N was heuristically set to 100.

Such an approach allows a set of contours with a consistent number of components to be extracted easily and automatically. However, one disadvantageous lies in the lack of correspondence between the components of different contours. It is shown in Chapter 7 that such a lack of correspondence introduces non-linear and discontinuous characteristics into any representation that incorporates this contour information.

3.4.3 Combining the Different Acquired Components

To determine which of the two of the three colour blob centres belong to the hands, the 3-D skeleton's hand vertices were used. The two centres which are closest to the 3-D skeleton's hand vertices (only the (x, y) co-ordinates of the 3-D hand vertices are used) are selected as the hand positions, whilst the remaining centre is set as the head position.

Finally, all the components are made relative to the head position. To achieve

this for the 3-D skeleton, the head's position was subtracted from the 2-D coordinates (x, y) of all its vertices. Similarly, the head position is also subtracted from the contour points and hand positions.

3.5 Characteristics of Human Body Kinematics using Hybrid Vectors

3.5.1 Nonlinearity: Movements of an Articulated Object

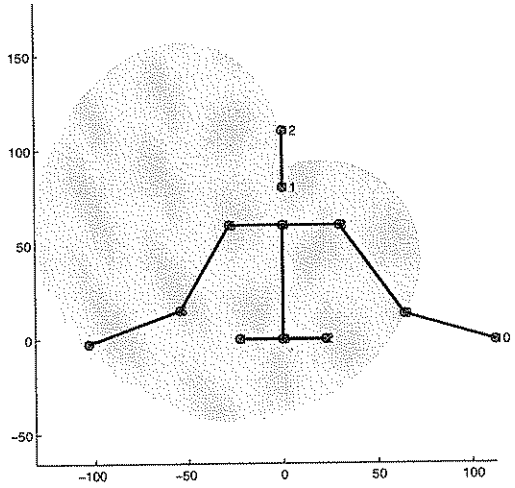


Figure 3.8: Constraint area of the left hand vertex. Illustrated is the shaded area that is bound by the locus of the most extended positions of the left hand from the body.

We find that the positional parameters for possible human body movements are often bounded by non-linear surfaces, reflecting bones and muscles that restrict the movements a human can make. While a degree of non-linearity can be avoided by using the skeleton joint angles instead of the 3-D positions of the joints, other forms of non-linearity will inevitably manifest itself in the visual information. As an example, Figure 3.8 shows the area covered by the 2-D projection of the left hand vertex of the 3-D skeleton. The shaded section illustrates the constraint area for the possible 2-D positions of the left hand vertex of the skeleton shown, with all other points attached to this body part also constrained to a similar shape. As a result, we would expect the visual components of the hybrid vector to have some

degree of non-linearity associated with it. However, it is difficult to see such an aspect by just observing the individual components of the visual modalities. An alternative approach is to treat this as a high dimensional visualisation problem.

3.5.2 Visualising the Characteristics

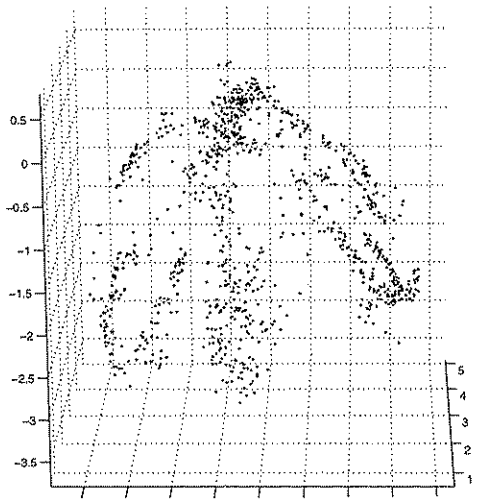


Figure 3.9: Nonlinear subspaces spanned by a range of human body configurations. Shown here is the visualisation of the hybrid vectors using Principal Component Analysis. The projections of the hybrid vectors onto the first three largest eigenvectors are shown.

Due to the high dimensionality of the hybrid vectors, it is not helpful to visualise all its contents simultaneously. However, one could still gain insights into the salient characteristics of this representation by employing methods that can capture such features. One simple but effective method for this is Principal Component Analysis (PCA). One of the characteristics of PCA is its ability to extract out the salient information by recovering example hybrid vectors that capture the largest variation across the different information types (i.e., skeleton, body part positions and shape). The structural characteristics of the representation can be visualised by projecting available training hybrid vectors onto this PCA space.

Nonlinear Subspaces

We can see the results of such projections in Figure 3.9 which makes apparent the non-linearities due to both the underlying 3-D skeleton vertex movement constraints and subsequently shape components. As noted previously, the hybrid vectors altogether occupy non-linear regions. Further analysis and methods into dealing with such a non-linear distribution will be provided in Chapter 6.

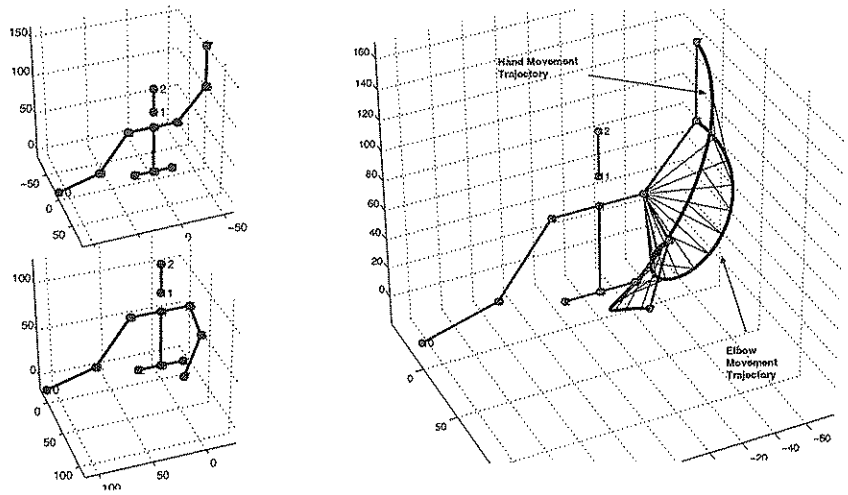
Nonlinear Trajectories

In addition to the fact that the structure of the valid hybrid vectors space is highly non-linear, there is yet another aspect of nonlinearity to deal with. This takes the form of nonlinearities in the temporal patterns of typical human body motions.

As an example, the Figure 3.10b shows the movement of a 3-D skeleton between two body configurations shown in Figure 3.10a. It is clear that the 3-D trajectory of the 3-D vertices concerned is highly non-linear. While the 3-D vertices can be replaced by a more linearly varying joint angle pair, the visual projection of the body parts will still yield non-linear trajectories (see Figure 3.11). We describe in Chapter 7 how such nonlinear trajectories can be learnt and the resulting models further exploited in Chapter 8 to aid the tracking of human motion patterns.

Discontinuities in Visual Representation

In addition to the non-linear trajectories caused by non-linear human body motions, we find that discontinuities in the dynamics of the visual representation introduce yet another form of nonlinearity. As a result, we find that the dynamics of the entire hybrid vectors is discontinuous as well. This issue will be the main topic of discussion in Chapter 7.



(a) (b)

Figure 3.10: Nonlinearities of a human body illustrated by the changes a 3-D skeleton undergoes shown in (b) between the two configurations shown in (a).

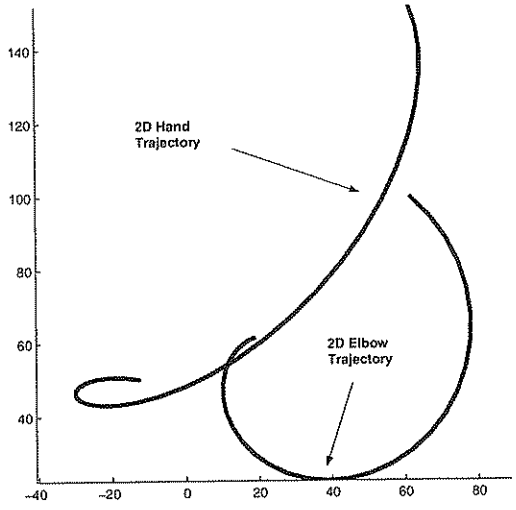


Figure 3.11: The trajectories of visual projections of the 3-D skeleton. Shown here are the (x,y) co-ordinates of the hand and elbow joint for the motion shown in Figure 3.10. It can be seen that the resulting 2-D trajectory is non-linear as well.

3.6 Conclusions

In this chapter, we have shown how 3-D skeleton models combined with shape and hand positions can be used to represent the configuration of a human body. Each bone in the skeleton model essentially accounts for a single body part. However, the parameters of the 3-D skeleton model are not directly observable from 2-D visual images. Therefore, 2-D information from the visual appearance of a human body is used to indirectly infer the 3-D skeletal parameters. To this end, hand positions and body shape 2-D information were used to represent the visual appearance of the human body. This results in the availability of three different types of information or modalities (3-D skeletal parameters, hand positions and shape information).

Since all three types of information were derived from the human body, there exists a correlation between the 3-D skeleton and human body visual appearance. In order to exploit such correlation, the three different modalities were firstly combined into a single representation known as a hybrid vector. This was achieved by concatenating the contents of the individual modalities together into a single high dimensional vector feature.

A constraint surface can be formed over the space occupied by the valid hybrid vectors. Essentially, such a surface provides a mechanism for identifying only the valid combinations of observable (2-D visual appearance) and hidden 3-D skeleton parameters. Where there are correlations between different information types, such a constraint surface spans across both the hidden and observable information components. Thus, in situations where only the observable components are available, one can use the observations in conjunction with the constraint surface to infer the hidden parameters.

In order to build such a constraint surface, one firstly needs an understanding of the characteristics of the valid hybrid vectors. To this end, PCA was chosen

to visualise the structure of the hybrid vectors. Since a hybrid vector contains data on the visual appearance and underlying structure of the human body, it is interesting to note how the characteristics of these modalities contribute to the overall structure of the hybrid vectors. In particular, constraints on the possible body configurations due to the muscular and bone mechanisms were found to have caused non-linearities in both the visual and structural components of a human body. Consequently, the valid hybrid vectors were also found to occupy a non-linear subspace. Such non-linearities manifest themselves in the dynamics of the hybrid vectors within this non-linear subspace. Furthermore, it was also found that the shape components adopted contained inherent discontinuities. The issues concerning the construction of the constraint surface will be further dealt with in a learning framework in Chapters 5 to 8.

Finally, one observes that, the 2-D information that results from 3-D objects will inevitably contain ambiguities. Different configurations of a 3-D object can result in the same 2-D projection. Such ambiguities can cause problems to the consistency of the inference process. Therefore, it is important to know the degree of ambiguities present in the data. The next chapter will deal with understanding more about the ambiguities inherent in the 2-D visual appearance of a human body when used to infer its corresponding underlying 3-D skeleton.

Chapter 4

Visual Ambiguities of 3-D Objects

4.1 Introduction

Ambiguities are a constant cause of many problems in computer vision. This is especially true when tracking 3-D articulated objects, which is one of our crucial tasks: the recovery of the underlying 3-D articulated object parameters from measurable 2-D features in images. Examples include the 3-D measurements for both location and orientation of different 3-D object parts. In general, 3-D model parameters cannot be obtained directly from input images since they are of different forms. That is, images are pixels while 3-D parameters are for example joint angles. However, it is known that a 3-D object does generate certain visual features which *can* be extracted directly from an input image (e.g. its shape information given by edges). The task is to recover the underlying 3-D object parameters using measurable visual features.

However, the lack of depth information in visual images can cause serious problems. One such problem is that of self occlusion, where parts of an object are obscured by other parts, causing important visual features to be lost. Another problem lies in the inadequacy of 2-D projections to uniquely represent 3-D objects at certain poses. Here, the underlying 3-D model at different poses generates very similar visual features.

4.1.1 Previous Work

Previously, Regh [39] predicted the presence of occlusions of parts of a hand using layers of image templates fixed onto a 3-D kinematics model. This work is related to work on tracking and motion coding [44, 79]. Each template here consists of an image of a particular hand segment. These image templates are then oriented in accordance to its corresponding part in the kinematics model. Additionally, since the kinematics model is 3-D, it also allows one to retrieve the depth ordering of the image templates. From this, one can detect the overlap between different templates. The greater the overlap, the larger the occlusion of a part. One limitation in such a method lies in the use of image templates. We find that the templates can only be rotated parallel to the image plane. Thus, should a part undergo rotations in depth, the image template would have to be reacquired. This was acceptable when a hand that does not change its orientation is being tracked from the side view, since at such a viewpoint, transformations that human fingers can undergo will be mostly parallel to the image plane. However this is not the case in the context of a human body where the body parts are capable of far more flexible movements, including large amounts of rotations in depth.

An alternative to image templates comes from adopting deformable curves to track the shape of the object of interest. In particular, MacCormick and Blake [34] used a deformable B-spline curve to track the shape a subject's head and shoulders. To handle the issue of multiple subjects and the possibility of subjects occluding other subjects, an integer "pseudo-depth" label was assigned to each subject. For example, the smaller the value of the label assigned to a subject, the closer this subject is to the camera. This label can then be used to predict the degree of overlap between different subjects. Consequently, this allows one to predict those segments of contours of a subject that will be visible. The correctness of the label value assignments are verified using a likelihood calculation. To use

such an approach in the context of tracking human body parts, a subject's head contour can be replaced by contours of other body parts. One disadvantage in such an approach lies with the fact that the amount of label combinations would greatly increase as the number of subjects to be tracked is increased.

Finally, another approach was adopted by Kakadiaris and Metaxas [27] where a 3-D human model was used to predict the degree of occlusions of different body parts. Here, from a particular viewpoint, a *visibility index* is assigned to each body part. Thus, the visibility index represents the degree of visibility of a body part at a specific viewpoint. The process where the visibility index value of a body part is calculated includes three steps:

- Computing the visible area of a body part with the possibility of occlusions by other body parts: The entire 3-D model is firstly projected onto the camera image plane with hidden surface removal. From this, the visible area of a body part (V_{Occ}) in the image plane was calculated.

- Determining the projection area of the *entire* unoccluded body part: Only the body part of interest is projected onto the camera's image plane. Following this, the area of this projection (V_{All}) is calculated.

- Calculate the occlusion ratio, $R_{Vis} = V_{All}/V_{Occ}$.

Therefore, the degree of occlusion a body part is under is indicated by the value of the ratio (R_{Vis}). For example, should the body part be highly occluded, the ratio would be small. However, such a method requires the existence of a 3-D model of the human body. In this context, the 3-D skeleton is inadequate, since the projection of the bones would only yield lines.

In this chapter, we provide an alternative method for quantifying the ambiguities when using visual information to infer unobservable information (3-D skeleton parameters).

The next section will define formally the notion of visual ambiguity in the context of the 3-D skeleton hybrid vectors representation. Section 4.3 provides a method for quantitatively measuring ambiguities and a general algorithm is introduced for extracting ambiguity measurements from existing training data.

In Section 4.4, a more specific algorithm for extracting ambiguities of estimated 3-D skeleton parameters using the visual features described in the previous chapter is given. Experimental results are provided in Section 4.5. In Section 4.6, we then see how visual information from different viewpoints can have varying ambiguities before concluding in Section 4.7.

4.2 Definition of Ambiguity

Observable and Hidden Information

First, a set of (A) different types of visual features $(\mathbf{v}_1, ..., \mathbf{v}_A)$ is defined as *measurable* data because it can be directly extracted from images. The vector, $\mathbf{v}_i$, with u_i number of components, contains information about the visual feature it represents: $\mathbf{v}_i = \{v_{i,1}, ..., v_{i,u_i}\}$. For example if $\mathbf{v}_i$ represents a point distribution model (PDM) of a contour, its components would consist of the (x, y) coordinates of its points. All the visual vectors are concatenated into a *measurement-data* vector, $\mathbf{w} = \{v_{1,1}, ..., v_{1,u_1}, ..., v_{A,1}, ..., v_{A,u_A}\}$.

Second, a *hidden-data* vector is defined as $(\mathbf{m})$ for storing the (B) underlying 3-D model parameters: $\mathbf{m} = \{m_1, ..., m_B\}$. The 3-D model parameters could take the form of joint angles for the 3-D skeleton. We will see in Section 4.4 where the 3-D model parameters are defined by the spherical angles of the skeleton bones to the $x - y$ plane. An illustration of this can be seen in Figure 4.6.

Finally, the hybrid vector $(\mathbf{y})$ is defined by the concatenation of the measurements-data along with its corresponding hidden-data: $\mathbf{y} = (\mathbf{w}, \mathbf{m})$.

For example, suppose we are interested in only part of a body, the left arm,

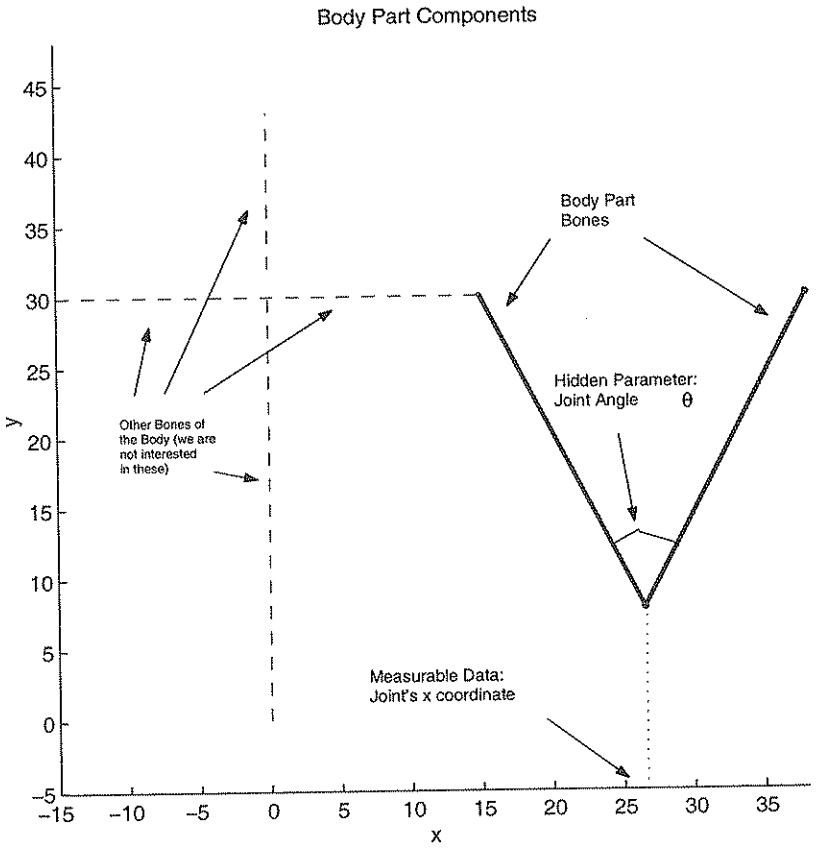


Figure 4.1: Components of a body part.

as illustrated in Figure 4.1. Furthermore, the only visible information is the x-coordinate (x) of a joint. The hidden information is the angle (θ) between the two bones which makes up the body part. The ambiguity vector is then defined as (x, θ).

Measurable-Hidden Information Constraint Model

A constraint model (volume or surface) can be constructed to capture valid instances of the measurable data (visual features) and its corresponding underlying 3-D model components.

Recovering the missing 3-D model parameters, given only 2-D visual features, can be achieved by finding the point on the constraint model whose visual fea-

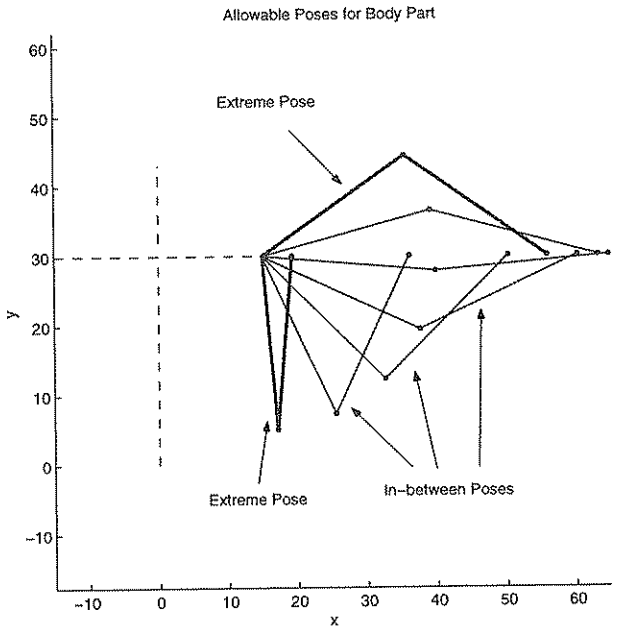


Figure 4.2: Valid body part poses which are allowed.

ture components are closest to the given input data. This yields a vector which represents the visual features closest to those recovered from the input image and which also contains the corresponding "hidden" 3-D model parameters.

Expanding on the previous example, the body part configurations are restricted to those shown in Figure 4.2. Given these restrictions a plot of the valid ambiguity vectors can be made (see Figure 4.3). The measurable information (joint x coordinate) takes the horizontal axis of the graph, while the hidden parameter (joint angle θ) takes the vertical axis of the graph. It can be seen that the hybrid vectors for valid left arm configurations falls onto a curve.

To make use of such the curve described above as a constraint model, firstly suppose we were given a novel measurement (x_{nov}). The goal is now to infer the hidden parameter (θ_{nov}) using this constraint curve. In order to do that, the point on the constraint surface whose measurable data (x co-ordinate) is closest or equal to that of x_{nov} is located. From there, the corresponding hidden parameter (θ_{nov})

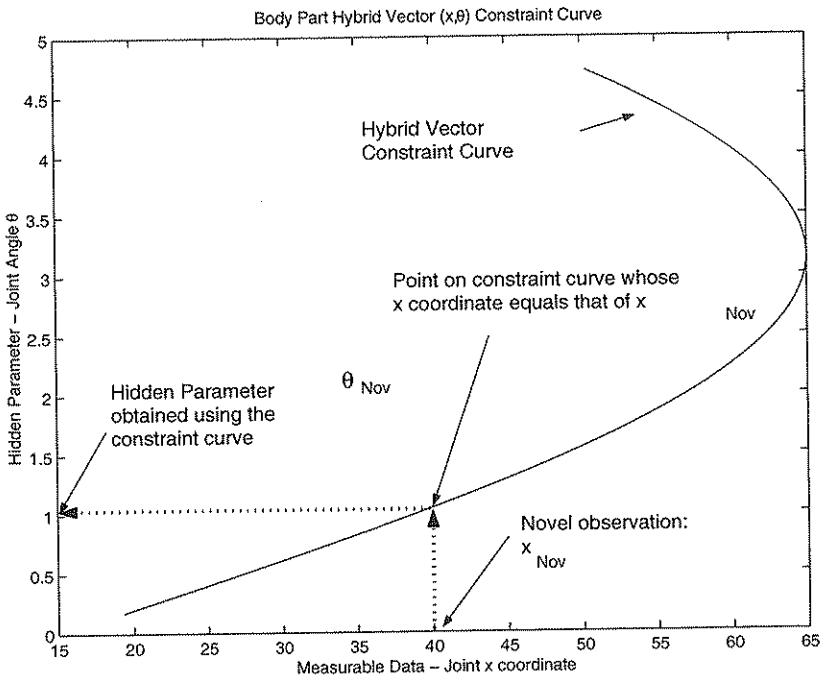


Figure 4.3: Body part hybrid vector constraint curve used to infer the hidden parameter.

can be located and thus inferred (see Figure 4.3). However, one notes that towards the end of the constraint graph, the constraint curve turns back on itself. Such a phenomenon occurs due to ambiguities present in using measurable information to infer hidden parameters.

Ambiguities in Measurable Features

Ambiguous and self-occluded visual features can cause multiple points on the constraint surface and have measurement-data equally similar to those extracted from the input image, but each with significantly different corresponding hidden-data components [21]. As a result, it is not possible to decide which 3-D model parameters can be selected for the given visual features. In other words, a hybrid vector has ambiguous measurable components when there exists many hybrid vectors with similar measurable components but dis-similar inferred components.

To illustrate this problem, we return to the previous example. As was noted,

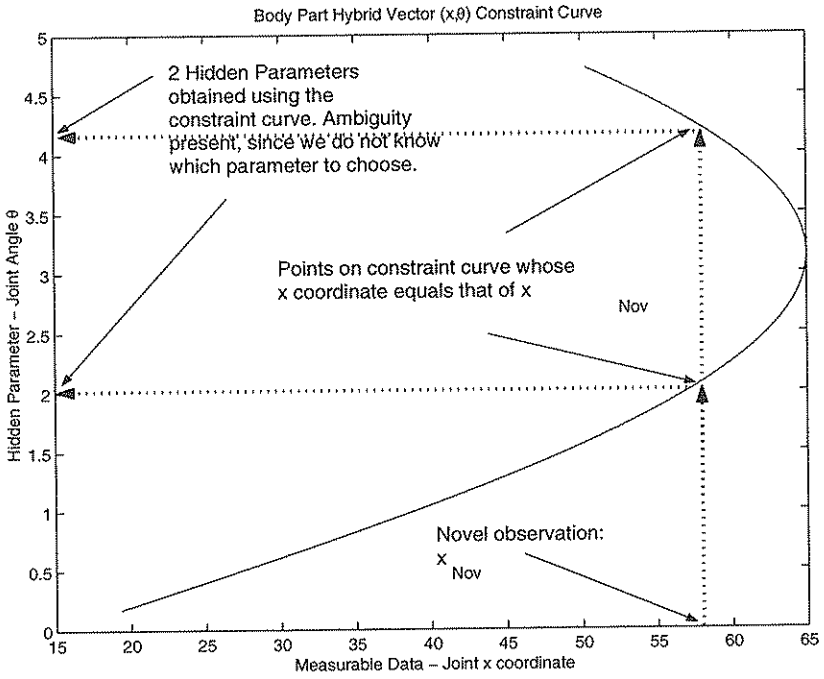


Figure 4.4: Ambiguities in using the constraint curve to infer the hidden parameter.

the end of the constraint graph showed the constraint curve turning back on itself. Therefore, if we were to receive observations (x_{nov}) in that region, we would have two points on the constraint surface with similar measurements (see Figure 4.4). In this case, it is not clear which is the correct hidden parameter (θ_{nov}). Should we have three potentially correct parameters, we would have a more difficult time choosing. Therefore, one can calculate the degree of ambiguity of the measurable data by counting the number of such potentially valid hidden parameter.

The process for recovering such a measurement-ambiguity-degree can be fairly straightforward. For example, one draws a vertical line and count the number of intersections with the constraint curve. However, in general, the constraint surface will in most cases not be a curve. Instead, it will take the form of a complex nonlinear hyper-surface. We will discuss more about such a constraint surface in Chapter 5 and Chapter 6. An equivalent process of "drawing a vertical line and counting intersections" would take the form of integrating across the

hidden parameters for each of the valid measurement data configurations. For complex hyper-surfaces or hyper-volumes, this is often intractable. However, one can approximate such a process when an existing set of training data containing both measurable data and its corresponding hidden parameters is available. In the next section, a framework is presented for performing such an approximation, i.e., extracting the ambiguity values of measurable data.

4.3 Quantifying Ambiguity: A Framework

This section describes a method for extracting the ambiguity values of each hidden component in the representation using its corresponding measurement components. As illustrated in the diagram given in Figure 4.5, the method consists of two components: a *measurements similarity function* and a *hidden components ambiguity function.*

4.3.1 The Measurement Similarity Function

A measurement similarity function is defined to compare visual features. As defined in Section 4.2, a measurement vector consists of a number (A) of visual feature vectors $(\mathbf{v}_1, ..., \mathbf{v}_A)$. In order to compare two sets of visual features, a set of functions $(f_1, ..., f_A)$ for measuring the similarities between two instances of a set of (A) visual features is introduced. The i^{th} visual feature similarity function $(f_i(\mathbf{v}_i^1, \mathbf{v}_i^2))$ is a mapping $f : \mathbf{R}^{u_i} \times \mathbf{R}^{u_i} \rightarrow \{0, 1\}$, where u_i is the number of components in $\mathbf{v}_i$. This mapping is responsible for comparing two visual feature instances $(\mathbf{v}_i^1, \mathbf{v}_i^2)$. Each similarity function depends on the visual features being compared, returning the value 1 when the visual feature instances $(\mathbf{v}_i^1, \mathbf{v}_i^2)$ being compared are deemed similar and the value 0 otherwise. An example can be seen in Section 4.4.1 for comparing the similarities of an articulated object's silhouette contours.

These individual visual feature similarity functions together define a *measurement-*

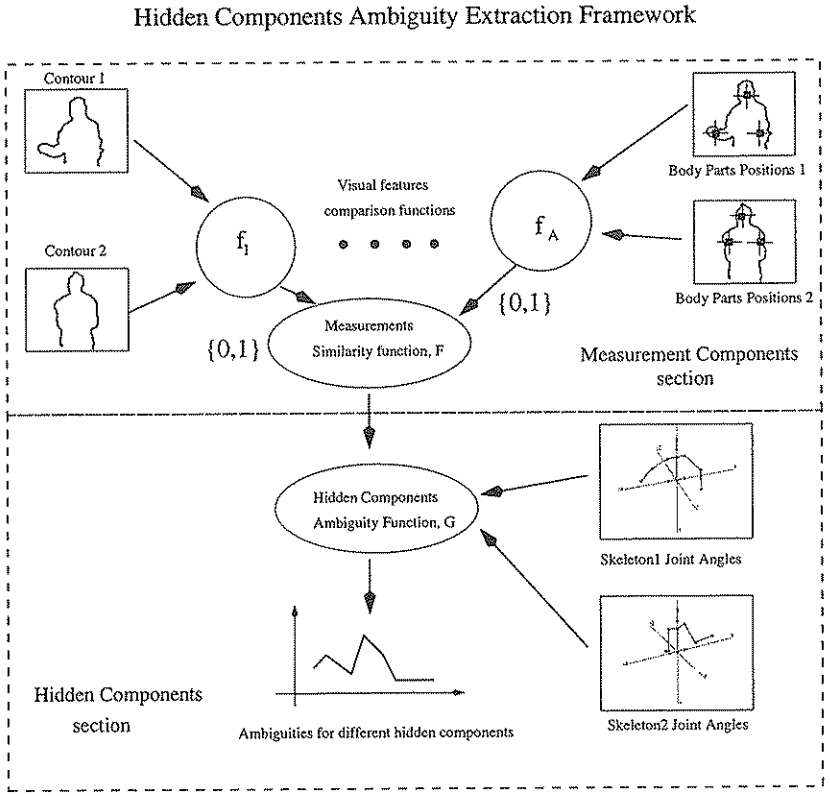


Figure 4.5: An overview diagram of the ambiguity extraction method described in Section 4.3.

similarity function,

$$\mathcal{F}(\mathbf{w}^1, \mathbf{w}^2) = \prod_{i=1}^{A} f(v_i^1, v_i^2) \tag{4.1}$$

where the a^{th} measurement vector is defined as $\mathbf{w^a} = (\mathbf{v}_i^a, ..., \mathbf{v}_i^a)$ and $a \in \{1, 2\}$. $\mathcal{F}$ returns 1 if all visual features in two instances of the measurement vector are similar enough.

4.3.2 The Hidden Components Ambiguity Function

A hidden component ambiguity function is defined for measuring the ambiguities of the hidden components given the measurements. In the case where the measurements are similar, as determined by, $\mathcal{F}$, a check for significant differences between its corresponding hidden components $(\mathbf{m}^1, \mathbf{m}^2)$ is made. To do this, a

function $\mathcal{G}(\mathbf{m}^1, \mathbf{m}^2)$ where $\mathcal{G} : \mathbf{R}^B \times \mathbf{R}^B \rightarrow \mathbf{R}^B$ is introduced. This function provides a vector that indicates to what degree each of the hidden components differ from one another, given the corresponding measurements $(\mathbf{w}^1, \mathbf{w}^2)$. Function $\mathcal{G}$, depends on what the hidden components represent. An example of this function is given in Section 4.4.2 for comparing two instances of 3-D skeleton joint angles and determining to what degree they differ and thus to what degree they are ambiguous.

Now, an algorithm is introduced for extracting the hidden components' ambiguity values from examples in a set of N training hybrid-vectors $(\{\mathbf{y}^1, ..., \mathbf{y}^N\})$.

Initialisation Step

- Create N number of B-dimensional vectors $(\{\mathbf{c}_1, ..., \mathbf{c}_N\})$ for storing the ambiguity values for the hidden-data components for each training example.
- Initialise all the components of $\mathbf{c}_{k,j}$ to 0, where $k \in \{0, ..., N\}$ and $j \in \{1, ..., B\}$.

Ambiguities Extraction Loop

- For each training example $\mathbf{y}^a = (\mathbf{w}^a, \mathbf{m}^a)$, where $a \in \{1, ..., N\}$,
  - For each of the other training examples, $\mathbf{y}^b = (\mathbf{w}^b, \mathbf{m}^b)$, where $b \in \{1, ..., a-1, a+1, ..., N\}$,
    - if($\mathcal{F}(\mathbf{w}^a, \mathbf{w}^b) == 1$),
      - Evaluate ambiguity values ($\mathbf{y}$) between hidden components, $\mathbf{m}^a$ and $\mathbf{m}^b$,

$$\mathbf{y} = \mathcal{G}(\mathbf{m}^a, \mathbf{m}^b)$$

      - Update the ambiguity values for example a,

$$\mathbf{c}_{a,j} = y_j, \text{ if } y_j > \mathbf{c}_{a,j}, \text{ where } j \in \{1, ..., B\}$$

This extraction process results in a set of vectors containing the ambiguity measures for the hidden components in each example. Having associated all the training data with their appropriate ambiguity measures, we next describe a method for modelling such ambiguities and labelling novel visual measurements.

4.4 Extracting the 3-D Skeleton Ambiguities

In order to extract the ambiguities of a 3-D skeleton, the visual information described in Chapter 3 is used. Following the terminology introduced in Section 4.2, we have a set of visual feature vectors, $\mathbf{v}_1, \mathbf{v}_2$ (i.e. $A = 2$). These vectors represent the visual measurements that can be directly extracted from the image.

The PDM of the object's silhouette contour is defined to be a vector $(\mathbf{v}_1)$ containing the coordinates of a number (u_1) of evenly distributed 2-D points; $\mathbf{v}_1 = (x_1, y_1, ..., x_{u_1}, y_{u_1})$. Next, we define the vector containing the positions of the left hand (x_l, y_l), right hand (x_r, y_r) and the head (x_h, y_h) as, $\mathbf{v}_2 = (x_l, y_l, x_r, y_r, x_h, y_h)$.

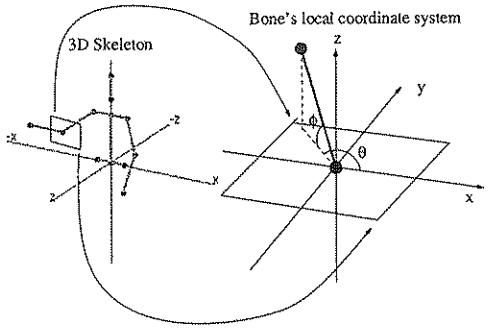


Figure 4.6: An illustration of the 3-D skeleton joint angles, θ and ϕ in the local (x, y, z) coordinate system of a joint.

The hidden components vector is defined as $(\mathbf{m})$ and contains $2u_2$ number of joint angles for a 3-D skeleton with u_2 number of joints; $\mathbf{m} = (\theta_1, \phi_1, ..., \theta_{u_2}, \phi_{u_2})$. Each joint contains two angles, θ and ϕ, which represents the angles of the joint off its local x and z axes respectively (see Fig 4.6).

In order to use the framework for learning the ambiguities described in the

previous section, we introduce a similarity function for the measurable data (2-D components) of the hybrid vector (Section 4.4.1) and a ambiguity function for the hidden components, i.e., the underlying 3-D skeleton's joint angles, in Section 4.4.2.

4.4.1 Similarity Functions for 2-D Measurements

Let us now see how two PDMs ($\mathbf{v}_1^1$ and $\mathbf{v}_1^2$) representing the body contours can be compared for similarity. Suppose $\mathbf{v}_1^1$ and $\mathbf{v}_1^2$ are both vectors with $2u_1$ number of components (i.e. u_1 number of 2-D points). Additionally, we define the n^{th} components of the k^{th} PDM vector (e.g. k is either 1 or 0 here) is defined as, $v_{1,n}^k$. In order to decide if $\mathbf{v}_1^1$ and $\mathbf{v}_1^2$ are similar or not, the following function was introduced:

$$f_1(\mathbf{v}_1^1, \mathbf{v}_1^2) = \prod_{I}^{u_1} d_1(v_{1,2i}^1, v_{1,2i+1}^1, v_{1,2i}^2, v_{1,2i+1}^2) \tag{4.2}$$

$$d_2(e,f,g,h) = \begin{cases} 1 & \text{if } \sqrt{(e-g)^2+(f-h)^2} \leq r \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

Based on equation (4.3), two PDMs are similar only if all the 2-D points of a PDM are within a vicinity (r) of the corresponding 2-D points on the other PDM on the image. In other words, r represents the variations in the PDMs' components due to errors and noise in the contour acquisition process.

A disadvantage of this method lies in the similarity comparison being local. That is, should all that is required for two contours to be considered dissimilar is for one component in the two different contours to have a distance greater than r (e.g. when one contour component is highly corrupted by noise). This is regardless of the fact that all the other points on both the contours may be very close.

For the purpose of our experiments, the value of r was set heuristically. This was achieved by observing the 2-D points on PDMs extracted from a sequence containing a subject at a static pose. Here, the position variance in the 2D points

on the contour due to image noise (since the body pose was kept static) was observed. This value was found to be about 5 pixels.

Next, we define two sets of body part positions to be similar if each body part's location is within a predefined vicinity, or "near enough" to a body part in the other set. Formally, if we are given two sets of body parts positions, $\mathbf{v}_2^1 = (x_1^1, y_1^1, x_2^1, y_2^1)$ and $\mathbf{v}_2^2 = (x_1^2, y_1^2, x_2^2, y_2^2)$, we define the similarity function for the sets of body parts positions to be:

$$f_2(\mathbf{v}_2^1, \mathbf{v}_2^2) = \prod_{i=1}^{2} d_2(x_i^1, y_i^1, x_i^2, y_i^2) \tag{4.4}$$

$$d_2(e, f, g, h) = \begin{cases} 1 & \text{if } \sqrt{(e-g)^2 + (f-h)^2} \leq r \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

where r represents the vicinity distance.

The size of the vicinity can be calculated for example by measuring the variance of the noise of the estimated body parts position. For our experiments however, a heuristic estimation of 5 pixels was chosen for the hand positions comparison function parameter, $r = 5 * 2 = 10$. Similarly a heuristic value of 5 pixel tolerance was chosen for the contour comparison parameter, giving $t = u_i * 5$. Both of these were achieved by visually observing the amount of movements of extracted contour points and a hand positions when the body configuration was kept static.

4.4.2 Ambiguity function for the skeleton joint angles

Given two joint angles of an articulated object's 3-D skeleton, they are defined to be similar if both are within a preset range (γ) of each other. This preset range determines the coarseness of the 3-D skeleton's joint angles estimation. Formally, the similarity function for comparing two corresponding 3-D skeleton joint angles sets, $\mathbf{x}^1$ and $\mathbf{x}^2$, is given as

$$\mathcal{G}(\mathbf{m}_1, \mathbf{m}_2) = (d_3(m_1^1, m_1^2), ..., d_3(m_B^1, m_B^2)) \tag{4.6}$$

$$d_3(\theta_1, \theta_2) = \begin{cases} |\theta_1 - \theta_2| & \text{if } \theta_1 + \gamma > \theta_2 > \theta_1 - \gamma \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

where, $\mathbf{m}^1 = (m_1^1, ..., m_B^1)$, $\mathbf{m}^2 = (m_1^2, ..., m_B^2)$ and B is the number of joint angles of the two skeletons compared. An *ambiguity vector* is also defined as $(\mathbf{v}_A = \mathcal{G}(\mathbf{m}_1, \mathbf{m}_2))$.

Additionally, an overall measure of an example's ambiguity values (A_v) can be obtained by summing together the values of the ambiguities:

$$A_v = \sum_{i=1}^{2B} \mathbf{v}_{A,i} \tag{4.8}$$

4.5 Experiments

For the experiments, a set of hybrid vectors containing the contour, body part positions and underlying 3-D skeleton joint angles was obtained. In particular, the contour is a PDM curve, with 100 points distributed evenly across the silhouette of the subject. The body part positions consists of the hand positions as described in Section 4.4.1. Finally, the underlying 3-D skeleton consisting of 13 vertices was firstly obtained. The joint angles of the skeleton were then extracted, resulting in 9 pairs of joint angles. A total of 1021 examples of different poses were obtained.

4.5.1 Body Parts Positions versus Contours

Using the set of training data described above, two experiments on the inherent ambiguities of the individual measurable-data (contour and body parts positions) were first carried out. In the first experiment, the contour information was removed, resulting in a set of hybrid vectors consisting of only the body parts positions and the corresponding 3-D skeleton. The ambiguities of the individual parts were then extracted. However, in order to understand the overall ambiguity of each example, the ambiguities of the individual parts were added together. This provides a single ambiguity measurement for each example.

A graph of the sorted examples' ambiguity values is shown in Figure 4.7. The graph with O marks represents the ambiguity of examples, which consist of the

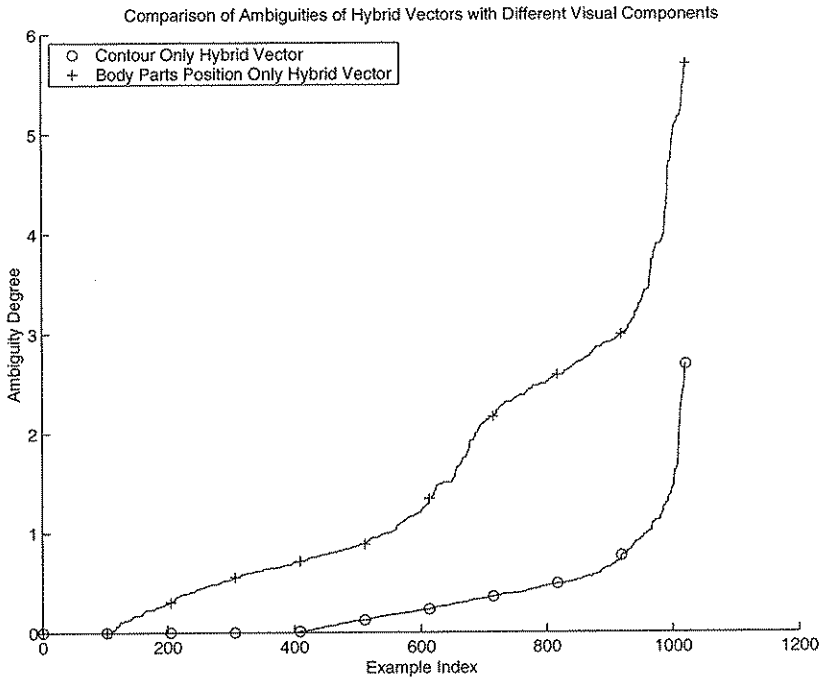


Figure 4.7: The results comparing the different ambiguity degrees for hybrid vectors which contain only the contour information and those which only contain the body parts position information as the measurable information.

contour and underlying 3-D skeleton. The other graph with + marks shows the ambiguity values of examples which instead, contain the body parts position and underlying 3-D skeleton.

As can be seen, it is clear that overall, the ambiguity values of the examples with the body part positions as the measurable information has a larger ambiguity value than those with the contour as the measurable information. Thus, one can conclude that the body part positions is inherently a more ambiguous measurable information when used to infer the 3-D skeleton.

One reason for such a result lies in the fact that only hands have been used to generate the body parts positions. Given the immense flexibility of the human body arms, there are many arm poses that can result in the same set of hand positions. Conversely, the contour has a larger number of points, which follow the outline of the body.

On the other hand, from a computational point of view, available methods for visually tracking body part positions like the hands are more robust than methods for tracking contours. Existing methods for tracking body parts positions usually rely on colour information [75]. This requires the colour of the body part to posses some form of similarity across different subjects. It is for this reason that usually only hands and heads are tracked. Other parts of the body are usually covered by clothing, which can have different colours across different subjects. Therefore, if one were to track those parts, individual colour models will need to be built for each subject. However, being able to retrieve such a colour model that is associated with the body part is not necessarily a trivial task. It requires being able to locate the body part in the first place to obtain its colour information. This brings about a circular argument for tracking body parts that are covered by clothing; one needs the colour models to obtain the body parts position, however, one needs the body parts positions to build the colour models.

Most contours can only be reliably extracted in a controlled background (e.g., a blue screen environment or a static background). Background clutter and moving background objects can often cause immense difficulties. Alternatively, as will be shown in Chapter 8, one can attempt to synthesise a contour, and check its correctness with respect to the input image. However, this too is susceptible to background clutter.

One therefore concludes that using visual information from body parts positions is inherently more ambiguous, but computationally more robust. Conversely, the contour has inherently less ambiguities, but is computationally a less reliable form of visual information for inferring its associated 3-D skeleton. In order to exploit the existing advantages of both types of information (contour and body part positions), a hybrid vector that contains both forms of information is used.

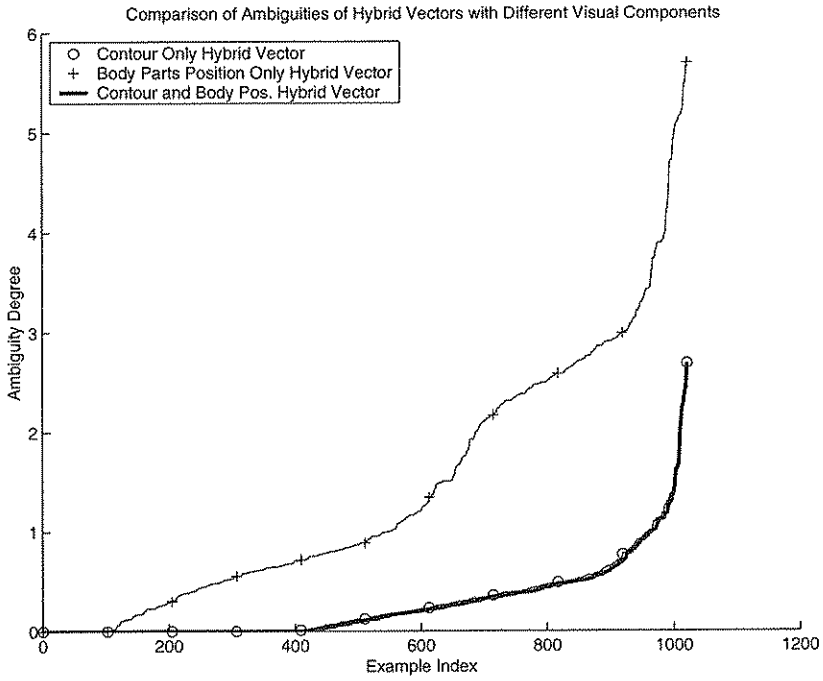


Figure 4.8: The results comparing the different ambiguity degrees for hybrid vectors which contain only the contour information, only the body parts position information and that which contains both contour and body parts positions as the measurable information.

4.5.2 Hybrid Vectors Give the Best of Both Worlds

From the previous two experiments described above, it was shown that the contour was less ambiguous but also computationally less reliable visual information. The body part positions had the opposite characteristics, being more ambiguous but also more computationally reliable. Therefore, it would be advantageous if one could exploit the less ambiguous nature of the contour and yet also have the computational reliability of the body parts position. To this end, the hybrid vector that uses *both* the body parts positions and body contour as the visual information is used. This form is consistent with that of the original data set, as was described at the start of this section. Therefore, the entire hybrid vector consists of the body part positions, body contour and the associated 3-D skeleton.

Similar to the previous two experiments, the body parts ambiguity values

of each example was first extracted and added together to provide an overall ambiguity measure. One can then visualise the extent of the ambiguities present in the hybrid vector examples by plotting the ambiguity values. In Figure 4.8, we compare the ambiguity values of the new hybrid vector to the other two hybrid vectors which consisted of individual visual information (i.e. contour only or body parts position only). As can be seen, the ambiguity values of the new hybrid vector are at least as low as that of the contour-only hybrid vector.

However, there remain many examples that have high ambiguity values. Such phenomena illustrate the insufficiency in using a single image from a single view for inferring the underlying 3-D skeleton at certain poses. In such a situation, a solution would be to provide visual information from a different viewpoint, such that the new view helps resolve the ambiguity.

4.6 Selecting Visual Information from Different Views

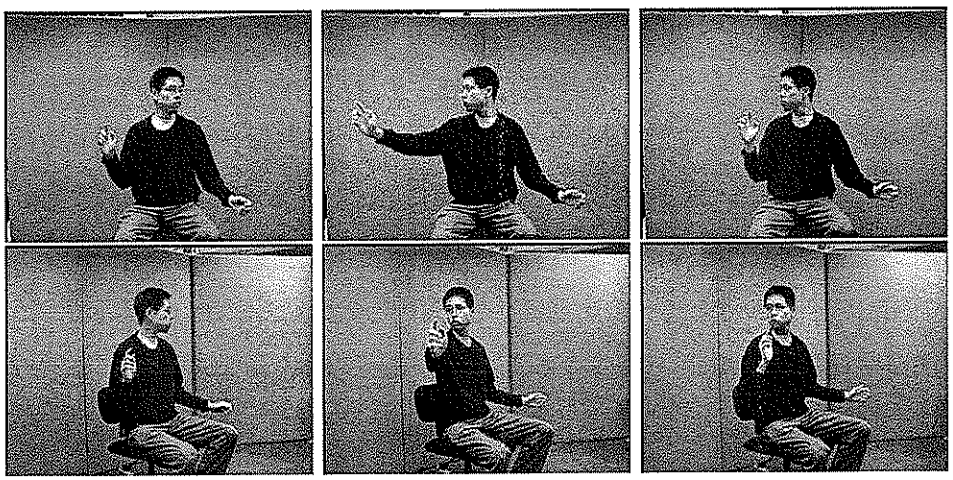


Figure 4.9: A multi view sequence. A subject's visual information was captured from different view points (front and three quarter view). The subject was told to extend and retract his arm over the entire sequence. The frames 1, 21 and 40 are shown to illustrate the different poses across the sequence.

To illustrate the ambiguities inherent in visual information obtained from dif-

ferent camera views, a multiple viewpoint example sequence of a subject undergoing body pose changes is shown in Figure 4.9. Two views were chosen, the front view and the 3/4 view. Images of both views were acquired simultaneously. Subsequently, the contour and body parts positions were extracted using similar methods for acquiring the training hybrid vectors as used in the experiments described in Section 4.5.1 and Section 4.5.2. The underlying 3-D skeleton was extracted by hand from the frontal view. It has to be noted that the 3/4 view hybrid vectors share the same underlying 3-D skeleton as the frontal view. This is due to the fact that they both represent the same body pose, although the visual information of such a body pose is obtained from different camera viewpoints. The hybrid vector sets for the sequences from both views was obtained by concatenating the contour, body part positions and 3-D skeleton vectors into a single hybrid vector.

The overall ambiguities of the sequence were analysed by firstly extracting the degree of ambiguities of the hidden parameters of each individual example. Again, an overall ambiguity measurement was obtained by summing all the hidden parameters' ambiguities together. This allows one to then compare the differences in ambiguities caused by the visual information obtained from different viewpoints (see Figure 4.10). One can see that in the majority of cases, one view (frontal) has more reliable visual information in comparison to the other viewpoint (3/4 view).

4.7 Conclusions

In this chapter, a method has been provided for quantifying the ambiguities of a human body's visual information. This was achieved by dividing the information that represents a body configuration into two categories: the measurable information and the hidden information. The measurable information consists of the

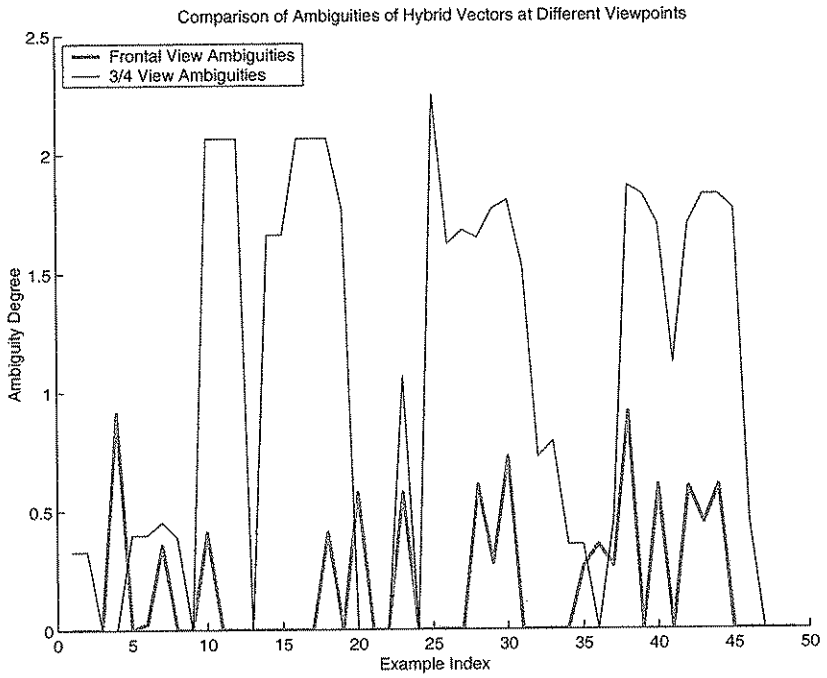


Figure 4.10: A comparison of the difference in the overall ambiguities in the visual information obtained between the front and three quarter view. It clearly shows that in the majority of the cases throughout this example sequence, the visual information from the front view is less ambiguous than that provided from the 3/4 viewpoint.

contour and body part positions of the human body. The hidden information consists of the underlying 3-D skeleton's joint angles. The degree of ambiguity of the hidden information (joint angles) can be extracted by detecting instances where different examples have similar measurable information but *different* hidden information. The degree of the ambiguities is then the magnitude of the variation in the hidden information.

With such a method, it was found that certain visual information is inherently ambiguous. In particular, it was found that the body part positions were more ambiguous than the contour information. Furthermore, combining both of these information sources into a hybrid vector resulted in a representation that was no more ambiguous than the least ambiguous visual modality.

Additionally, the limitation of the visual information of body configurations at

certain points was also discovered. Evidence of such a phenomenon can be seen at certain body configurations, where all the visual information gives a high degree of ambiguity. It can be seen that visual information acquired from a different viewpoint about such body configurations can be less ambiguous.

The ambiguity measurements only provide us with a means for estimating the ambiguities of different hidden parameters. Such a measurement only indicates how unreliable the inferred hidden parameters will be. However, the actual values of these parameters are not estimated. In order to do this, one needs to deal with both the spatial and temporal dynamics of the human body's underlying kinematics model. In the next chapter, we will describe how different characteristics of the human body's spatial kinematics can be revealed and learnt using an example based framework.

Chapter 5

Learning Human Body Configurations

5.1 Introduction

A human body like any articulated object is capable of many possible configurations. In order to computationally capture these body configurations, a representation of the human body was introduced in Chapter 3. This representation is constructed as a hybrid vector containing information on both the visual appearance and underlying 3-D structure of the human body. The visual appearance takes the form of shape data and the positions of the hands. Meanwhile, a 3-D skeleton model represents the human body's underlying structure. Adopting the hybrid representation, the differences in body configurations will manifest as variations in both the visual appearance and structural information. The nature and complexity of such variations result from the muscular and joint constraints which restricts the possible body configurations for the individual body parts. Consequently, only a limited set of body configurations will be valid. With an objective of modelling such a set of valid body configurations, a greater understanding into the complexity of its variations would be advantageous. This chapter presents a method that can be used to not only analyse the complexity of different human body configurations but also build a computational model for it.

In general, the complexity of the human body can be expressed as the amount

of "information" that is required to capture all of its valid body configurations. One then needs a more specific definition for this "body pose information". To this end, the method of linear combination of examples provides a suitable framework. Intuitively, the linear combination framework revolves around the linear combination of prototypical examples, or *prototypes* for generating novel examples of a similar form. Thus, one can treat the prototypes, as the information needed to capture different body configurations.

5.1.1 Linear Combination of Examples

The method of linear combination of examples was initially exploited by Ullman and Basri [76] for the recognition of 3-D objects using linear combinations of 2-D images of the object. This approach was also explored by Poggio and Vetter [86] in the case where only one prototype image was available. It was shown that the visual changes caused by transformations on an object could be captured by a small set of images depicting the object at different viewpoints. These images can be regarded as prototypes. Therefore, any novel images of the object at different viewpoints can be reconstructed by a linear combination of these prototype images. Computationally, this involves a sum of weighted prototype images. The constraints for these weights or coefficients of the linear combinations were analytically derived the object's 3-D transformation equations.

However, such an approach has the limitation that only rigid 3-D transformations were accounted for. This is insufficient when dealing with articulated or deformable objects. Examples of such objects are faces or human bodies. To tackle such an issue, correspondence between the components of two examples has to be established initially [54, 86, 87, 85]. Such a task involves recovering the mapping of each component on an example image to a corresponding component on the other example. For example, such a mapping would involve establishing pixelwise correspondence between two example images. However, in the case of

certain model-based examples (e.g. 3-D hand model [82]), such component correspondences would already have been established. This is because each component consistently represents the same point on the object (e.g. each vertex represents the same point on a 3-D hand model, regardless of the hand pose). However, there are other model based representations where correspondences between different example components have not been established (e.g. Point Distribution Model shape [3, 81]). In such cases, a method for establishing correspondences between the components of two different examples is required. We will see in Chapter 7 how a dynamical model can be used to indirectly learn such required correspondences.

In the remaining sections of this chapter, a description and definition for the LC method is given in Section 5.2. A learning method for extracting the required examples will be described in Section 5.3. An interesting consequence of such a learning process in achieving information fusion will be described as well. Having defined the characteristics of the examples, they are then used to investigate the degree of complexity of the human body motions in Section 5.4. Additionally, an insight into the salient human body kinematics and visual ambiguities captured by different examples is provided. Finally, a summary is given and conclusions are drawn in Section 5.5

5.2 Linear Combinations: A General Definition

Linear combinations of examples allows one to recover novel vector-based representations by linearly combining prototypes of other vector instances (see Figure 5.1). The significance of this method is its ability to "encode" a generative representation into coefficients of linear combinations of such prototypes.

5.2.1 Definition

Formally, a linear combination of examples is defined as follows. Given an arbitrary N-dimensional representation (e.g., a system with N variables for tracking), suppose there exists a set of (E) example instances, $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_E\}$, of this representation, a novel instance $(\mathbf{n})$ can be reconstructed by the linear combination:

$$\mathbf{n} = \sum_{i=1}^{E} a_i \mathbf{e}_i \tag{5.1}$$

where the set of coefficients for the linear combination is $\{a_1, a_2, ..., a_E\}$.

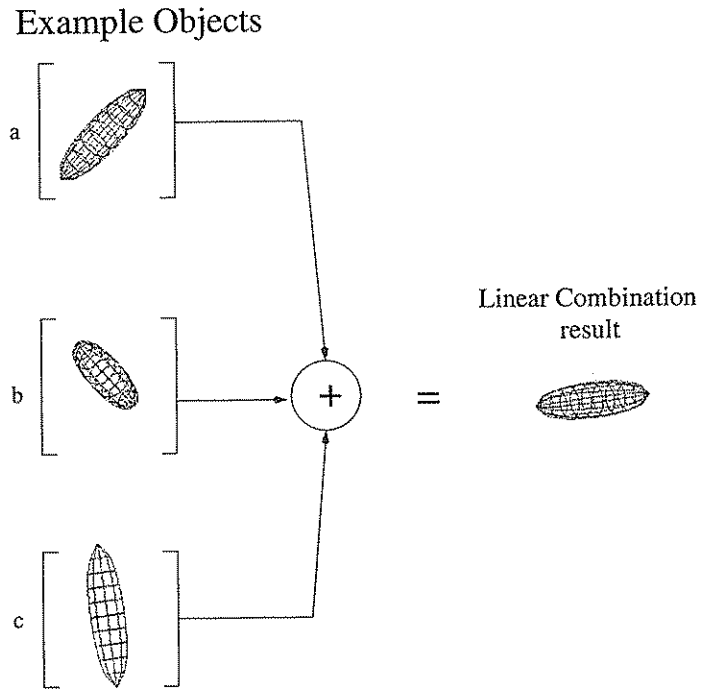


Figure 5.1: An illustration of the linear combinations concept. The objects initially weighted (e.g. a, b, c are the weigths in the figure) before combined together through the addition operation.

An immediate issue that comes to mind is the question of how the required examples are to be extracted. To address this, the next section will define a computational method that will be used to *learn* the prototypes from an available training set.

5.3 Learning the Prototypes by Information Fusion

Two computational issues in learning the prototypes are considered: determining the number of prototypes and the contents of these prototypes.

5.3.1 Example Learning and Information Fusion

Our hybrid vector consists of a number of different types of information. Correlation between and within the different information can be calculated using a correlation matrix of available examples of such vectors. Formally, given a set of hybrid vectors, $\mathbf{X} = \mathbf{x}_1, ..., \mathbf{x}_N$, its correlation matrix ($\mathbf{S}$) can be recovered by

$$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{x}_m)(\mathbf{x}_i - \mathbf{x}_m)^T \tag{5.2}$$

where $\mathbf{x}_m$ is the hybrid vectors global mean given by:

$$\mathbf{x}_m = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{5.3}$$

Having recovered the correlation matrix, one can now represent it instead using a set of examples, $\mathbf{e}_1, ..., \mathbf{e}_K$. To this end, Principal Component Analysis (PCA) provides us with just the appropriate means for recovering such examples. Formally, each example ($\mathbf{e}_i$) is required to satisfy the following:

$$\mathbf{S}\mathbf{e}_i = \lambda_i \mathbf{e}_i \tag{5.4}$$

That is, the examples take the form of eigenvectors of the correlation matrix. Additionally, each example ($\mathbf{e}_i$) is also associated with an eigenvalue (λ_i) denoting the significance of the correlation between the different components along this example's direction. We also sort the examples in descending significance. That is, the first example ($\mathbf{e}_1$) is associated with the largest eigenvalue, the second example with the second largest eigenvalue and so on.

The contents of the examples are therefore given by the eigenvectors. Since the examples that are the eigenvectors altogether represent the correlation matrix, they can also be thought of as modelling the correlation between the information. Therefore, it can be thought of as revealing the common traits between different modalities.

5.3.2 Effects of Noise on Determining Example Sufficiency

One can then select the required number of prototypes by observing the number of non-zero eigenvalues. However, this assumes that the available hybrid vector components are not corrupted by any noise, which in most real world cases, tends not to be true. As a result, such noise will cause certain redundant eigenvectors to take on non-zero eigenvalues. It is then common to introduce a heuristic to "prune" insignificant eigenvectors by cutting off eigenvectors which contribute less than a predefined amount to the total eigenspace. That is,

$$\lambda_i = \begin{cases} \lambda_i & : \ \sum_{j=1}^{i} \lambda_j > L_e \\ 0 & : \ \sum_{j=1}^{i} \lambda_j <= L_e \end{cases} \tag{5.5}$$

However, this requires one to define the "percentage of significant eigenspace" value (L_e). A more general alternative to this is to use the Bayesian methods for performing example-set-model selection.

5.3.3 Bayesian PCA: The most probable examples

Here, we use a method introduced by Kass and Raftery [68]. We can compute the number of required prototypes by evaluating the probability of modelling the data ($\mathbf{X}$) with k prototypes:

$$p(\mathbf{X}|k) \approx \left(\prod_{j=1}^{k} \lambda_j \right)^{-N/2} v^{-N(d-k)/2} N^{-(m+k)/2} \tag{5.6}$$

where the the number of training data is given by N, the dimensionality of the data is d, $m = d(d-1)/2 - (d-k)(d-k-1)$ and the average variances of

the discarded prototypes (v) is given by

$$v = \frac{\sum_{j=k+1}^{d} \lambda_j}{d-k} \tag{5.7}$$

Using this, the dimensionality (k) which yields the highest probability value will be chosen as the value for the number of prototypes. Having provided a method for computationally extracting both the number of required prototypes and their contents, we now move on to applying these methods to analysing the complexity of human body motions.

5.4 Salient Human Body Kinematics

Here, we will see the outcome of the process for extracting the prototypes of the upper torso of a human body. The data set used was that of the hybrid vector set described in Chapter 3. Each hybrid vector consists of three different modalities; the 3-D skeleton vertices, the contour of the body silhouette and the positions of the hands.

5.4.1 Number of Sufficient Prototypes

Thus, in order to recover the sufficient prototypes for modelling the kinematics of the upper torso, the eigenvalues (λ_i) of the hybrid vector examples covariance matrix was recovered using Equation 5.4. The eigenvalues and its cumulative graph can be seen in Figure 5.2. The presence of noise in the skeleton hybrid vectors has allowed some eigenvalues to be non-zero, when they should in fact be zero. In order to determine the appropriate number of prototypes, the probability of using different numbers of examples to capture the data set was evaluated using the Bayesian PCA method described in Section 5.3.3. Based on Equation 5.6, the graph showing the probabilities is obtained and shown in Figure 5.2c. From this, it was found that the required number of prototypes for capturing the variations of both the visual and hidden modalities of the human body's upper torso is 67.

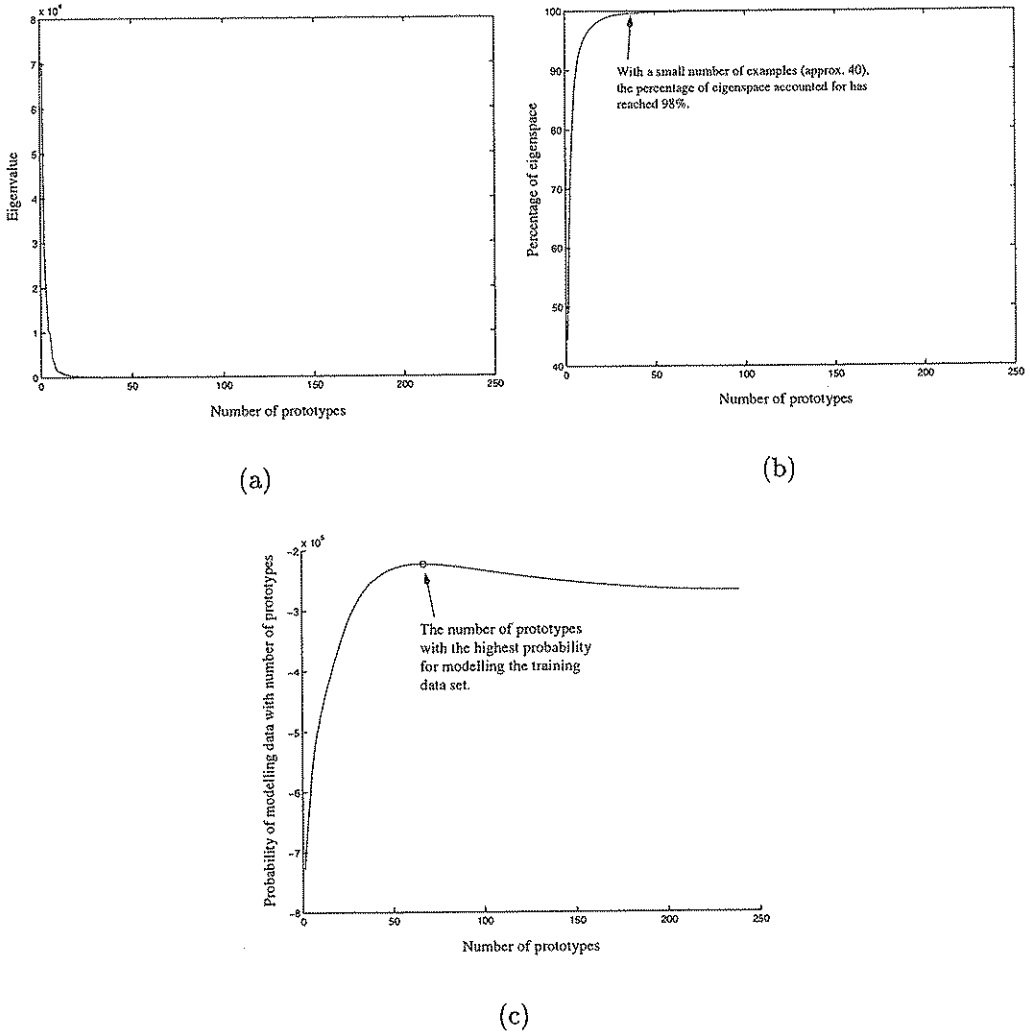


Figure 5.2: The eigenvalues of the extracted examples. The eigenvalues is shown in (a) while its corresponding contribution in percentage to the capturing of the hybrid vectors variations is shown in (b). Additionally, the log probability of modelling the skeleton hybrid vectors using different number of prototypes is shown in (c). The highest probability is highlighted with a circle.

5.4.2 Analysis of Variations Captured by the Prototypes

Having decided the sufficient number of prototypes, we now move on to inspect the properties of these prototypes themselves. The mean hybrid vector can be seen in Figure 5.3. As each example has the same dimensionality of the hybrid vector, it can be split into the appropriate modality vectors, each of which can be separately displayed. Therefore, for each example, we can view the first 200 dimensions as a shape vector, the next 36 dimensions as the 3D vertices of the skeleton and the last 4 dimensions as the positions of the hands.

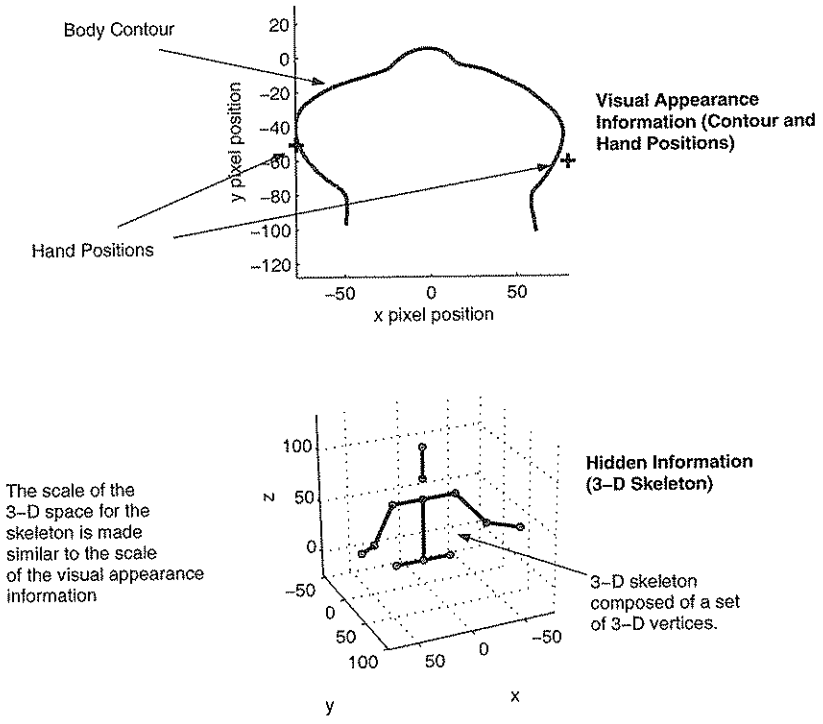


Figure 5.3: The skeleton hybrid vectors mean vector. Shown here are the visual appearance information on the top consisting of the contour and hand positions (crosses). The 3-D skeleton is shown at the bottom as a collection of connected 3-D vertices in the 3-D space. The scale of the 3-D skeleton space is made similar to the visual information such that all the components of the hybrid vector have the same variance scale.

We can therefore visualise each of this component. However, in displaying the components of the prototypes, we find that they do not directly convey any meaningful information (see Figure 5.4). One of the reasons is that the components of

the prototypes represent the *variations* in the hybrid vectors. As such, it would perhaps be more meaningful to visualise their properties in terms of deviations from the mean skeleton hybrid vector. In doing so, we find that the first few examples captures the large variations of all the three different modalities, as can be seen by its extent in the enclosed regions in Figure 5.5. As we progress further into the latter examples, we find that they contribute less and less to capturing any variations in the dynamics of the visual observations (shape and body parts positions) variations as well as the skeleton variations (see Figure 5.6). Towards the end of the required number of prototypes, we start to see negligible variations in different modalities. This can be observed in Figure 5.7 which shows prototypes 60 to examples 67.

5.5 Conclusion

In this chapter, investigation was given to the task of modelling the valid human body configurations. It was also required that the resulting model can be used to reconstruct any valid body configurations at will. Tackling such a task requires one to deal with the complexities of a body pose. In using a hybrid vector based representation for modelling the body pose, such a task would require one to be able to handle the variations present in the visual appearance and the structural information of a human body. To this end, the linear combinations of examples framework was used. There, prototypes are used to capture the above mentioned variability in the visual and structural information of a human body. Reconstruction of an example representing a novel body pose can be carried out by linearly combining the prototypes.

It was found that the required prototypes could be recovered by using the statistical method of PCA. There, eigenvectors representing salient variations across the different components of the hybrid vector are used as the prototypes. Ad-

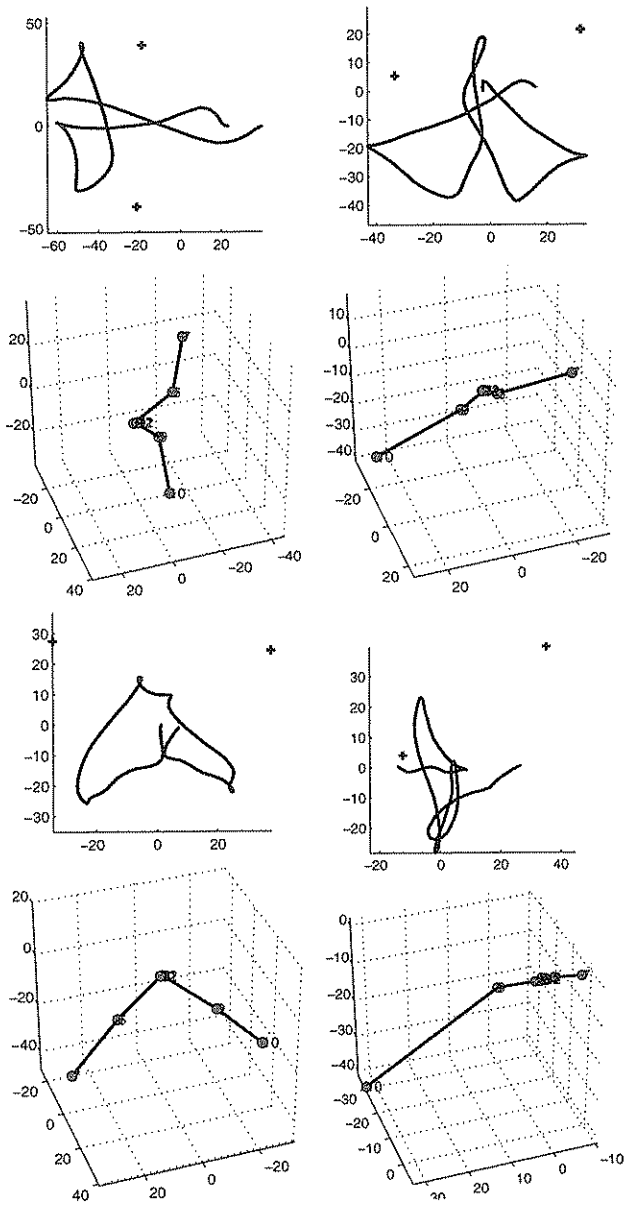


Figure 5.4: Visualisation of the extracted example components. Each example is divided into two parts, the top part which shows the contour components and body parts positions. The body parts positions are shown as crosses. Meanwhile, the lower part shows the skeleton components.

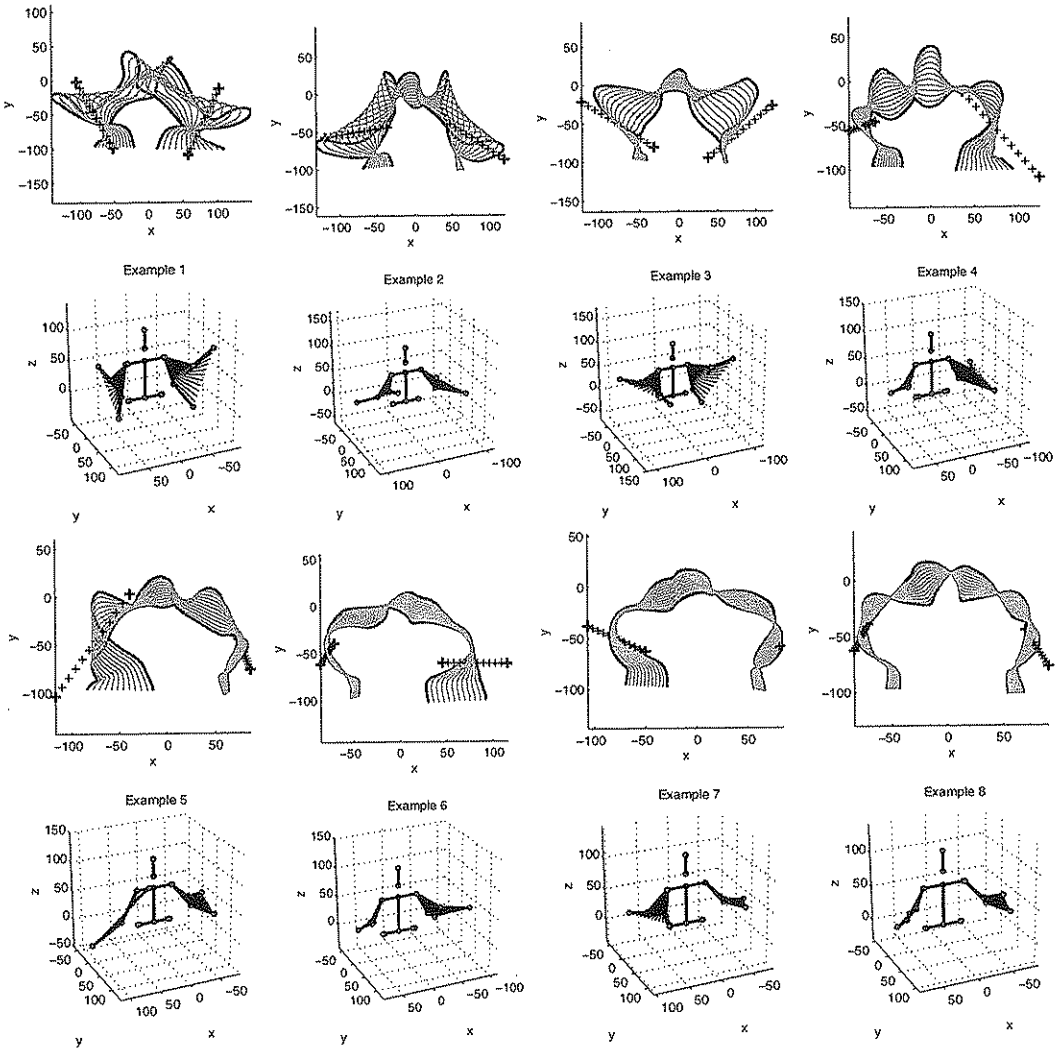


Figure 5.5: Visualisation of the first 8 extracted example components as deviations from the mean vector. For the contours, the extreme deviations are shown as thick dark silhouettes. The lighter contours represent those in between the extremes. Along with the contours, the right and left hand positions are indicated by crosses. The bottom part shows the different 3-D skeletons the example captures.

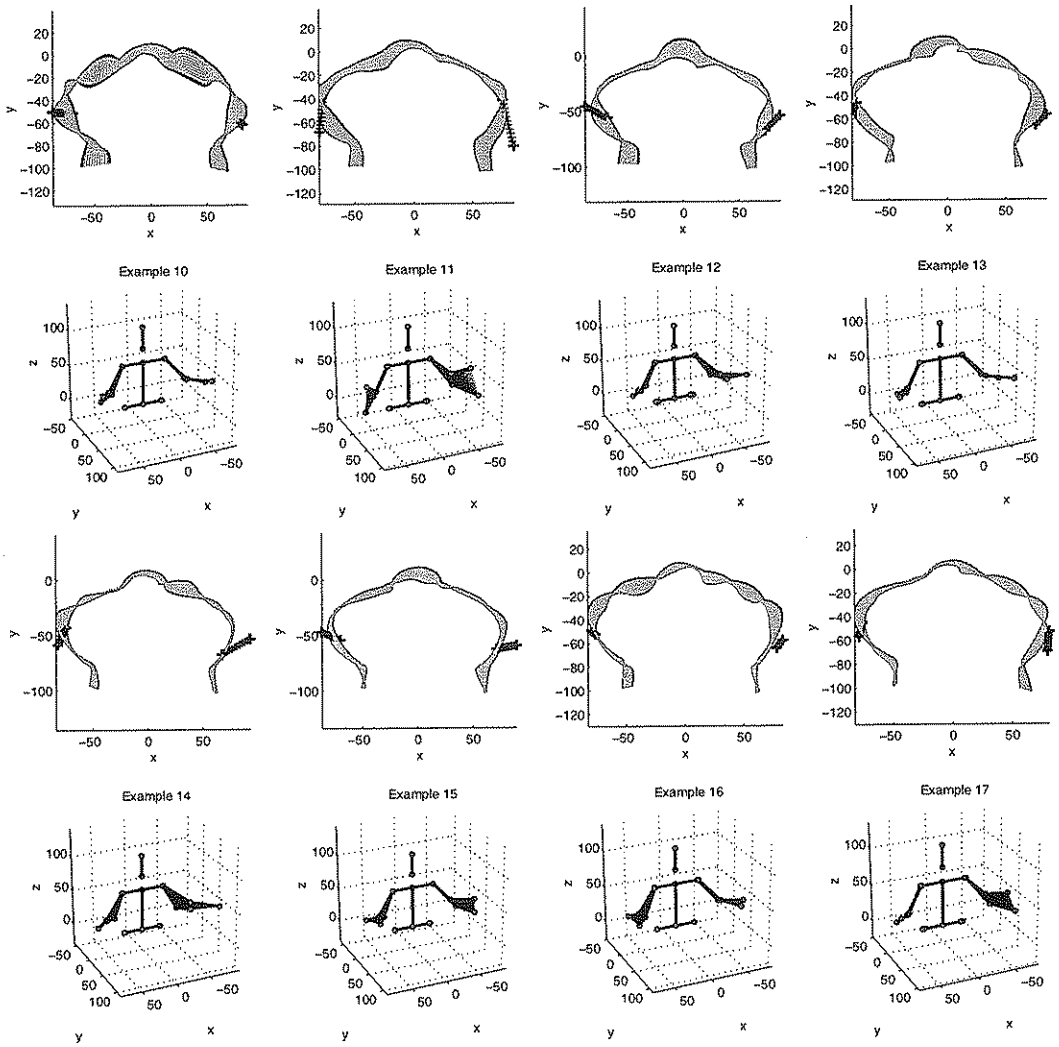


Figure 5.6: Visualisation of the 10th to 17th extracted example components as deviations from the mean vector. For the contours, the extreme deviations are shown as thick dark silhouettes. The lighter contours represent those in between the extremes. On the same section as the contour, crosses indicate the positions of the left and right hand. The bottom part shows the different 3-D skeletons the example captures.

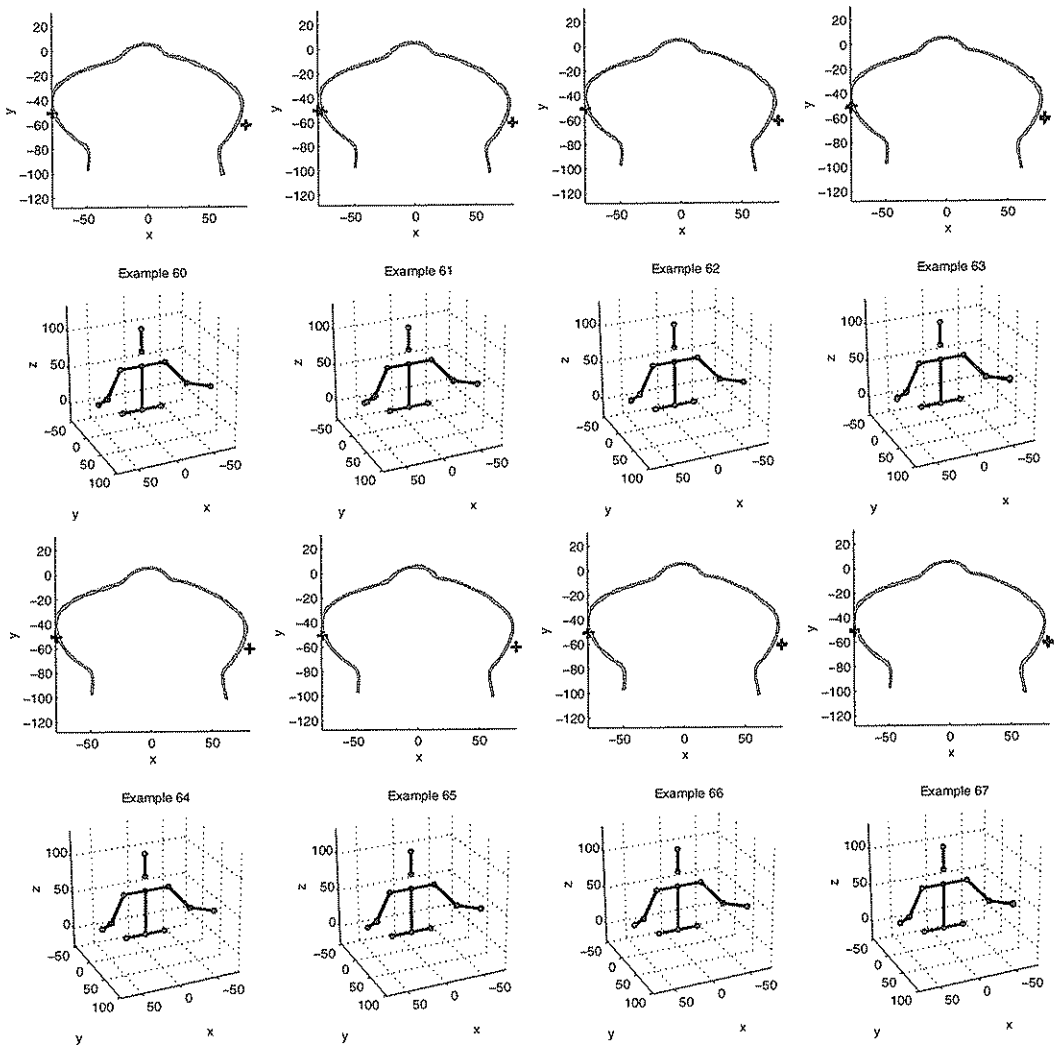


Figure 5.7: Visualisation of the 60th to 67th extracted example components as deviations from the mean vector. For the contours, the extreme deviations are shown as thick dark silhouettes. The lighter contours represent those in between the extremes. Along with the contour, the left and right hand position are indicated by crosses. The bottom part shows the different 3-D skeletons the example captures.

ditionally, the contribution in capturing different variations in the human body configurations is indicated by the prototype's associated eigenvalue. Such eigenvalues can be used to compute the number of prototypes as well as identifying and discarding redundant ones. To this end, PCA was extended into a probabilistic framework to identify the necessary number of examples for capturing all the possible human body configurations. There, a probability measurement of a set of examples modelling the hybrid vector training data is given. The number of prototypes that gave the highest probability value can be found. Insight into the degree of complexity of the human body motions can be gained from the number of required prototypes. The larger the degree of freedom the entire set of body parts is capable of, the larger the number of required prototypes.

We find that information fusion was achieved in the use of the prototypes for reconstructing a hybrid vector consisting of different types of information (visual and structural). This is because the prototypes are the eigenvectors of the hybrid vector correlation matrix. In other words, the prototypes set is a model for the correlation between the structural and visual appearance of the human body.

In using only a small set of prototypes to represent the valid body configurations, dimensionality reduction has also been achieved. Through a probabilistic PCA method, the number of prototypes necessary for reconstructing all the valid human body motions was found to be much less than the hybrid vector dimensionality. Specifically, the dimensionality of the hybrid vector was 240 while the necessary number of prototypes only amounted to 67 in the case considered. As such, each body configuration can be represented instead, by linear combination of coefficients instead of the entire hybrid vector. Here, this would amount to a fourfold reduction in the dimensionality of the hybrid vector. Furthermore, the original hybrid vector can be reconstructed by performing linear combination when necessary. Simultaneously, one can think of having indirectly learnt the kinematics parameters of the human body in the form of prototypes. The effects

of the body kinematics itself can be reproduced by the linear combination of the prototypical hybrid vector examples. However, there remains a missing part for completely reproducing the effects of the human body kinematics.

The prototypes only capture the salient linear variations across all the body configurations. Consequently, the non-linear nature of the human body motions is not addressed. This is because certain combinations of the prototypes can yield representations of unrealistic body configurations. Therefore, there is a need for constraining the possible linear combinations. The next chapter will concentrate on learning the constraints on the linear combinations such that only valid human body configurations can be generated.

Chapter 6

Learning the Body Kinematics Constraints

In the previous chapter, we discussed how to model different human body configurations using the example-based approach called Linear Combinations was discussed. Prototypical examples were used to capture the spatial dimensions spanned by the body configuration representation. Additionally, by linearly combining the prototypical examples, information on novel body configurations can be generated. Following this, it is natural to ask whether there is a need for constraints to be imposed on the possible prototypical example combinations?

In Ullman and Basri's [76] pioneering work on the linear combinations method, constraints on the valid coefficients were derived explicitly from the transformation equations applied to an object. However, these constraints were only considered for rigid transformation of a non-articulated object. Vetter and Poggio [87] later used such a method for modelling the human face in 3-D. Nevertheless, no constraints were imposed on the linear combination coefficients. This was due to the proposition that faces, whilst being flexible models and undergoes non-rigid transformations, still falls into the *linear object class*. It was proposed by Vetter and Poggio [86] that models which belong to the linear object class can be correctly described by a set of prototypical examples. This assumes that all possible combinations of the prototypical examples would yield reasonable results. Although

this may be true for rigid objects and even faces, we have seen in the previous chapter that this does not apply to representations of the human body. The aim of this chapter is therefore to investigate the causes of this shortcoming and propose a solution for it.

6.1 Chapter Overview

One cause for the generation of bad linearly combined examples lies in the violation of the underlying constraints that restrict human body motions. The space spanned by representations for human body motions ought to be constrained by this restriction.

Analysis of the structural restraints upon modelling the linear combinations is detailed in Section 6.2. We will see that explicitly modelling such constraints is often difficult. This is because the constraint model needs to account for highly non-linear characteristics, even for simple articulated objects with few degrees of freedom. We will also see an analogous investigation for the more complex case of the human body by analysing the linear combination coefficients of a set of training hybrid vectors. From this, we will see that the constraint model for the human body configurations would have to cope with highly non-linear characteristics.

Instead of explicitly defining the kinematics equations for the human body configurations, an alternative approach using a *learnt* generic cluster based model was used to represent the constraint surface. The details are given in Section 6.3. Following this, Section 6.4 describes the method chosen for learning the necessary non-linear kinematics constraints for human body configurations. Next, results on the learnt kinematics constraints are shown in Section 6.5. A comparison of the results of using different cluster models for reconstructing missing hidden information (3-D skeletons) is given in Section 6.6. Finally, Section 6.7 concludes the chapter.

6.2 Nonlinearity: Coping with Articulation Restrictions

6.2.1 Motivating Example

Let us now consider the problem of learning the constraints on the linear combination coefficients. First, the need for *learning* the coefficients constraints using the example of a 2D hierarchical articulated object (see Figure 6.1) is discussed. Supposed that the articulated object has three 2D vertices, p_1, p_2 and p_3.

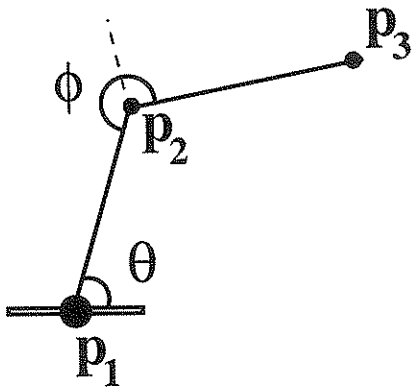


Figure 6.1: An illustration of a simple articulated object with 3 vertices. It consists of two fixed length parts (from p_1 to p_2 and from p_2 to p_3).

A hierarchical structure is imposed on the vertices of this object. The vertex, p_1, is the parent of all the other vertices. The second vertex, p_2, is linked to p_1, and the third vertex p_3, is linked to p_2. Therefore, both p_2 and p_3 are affected by any transformations on p_1. Additionally, any transformation on p_2 affects p_3. That is, the object consists of two fixed length parts (see Figure 6.1). Second, a "transformation-unified" representation can be made for the articulated object by concatenating all its vertices into a 6-dimensional vector $o = (p_{1,x}, p_{1,y}, p_{2,x}, p_{2,y}, p_{3,x}, p_{3,y})$. The movement constraints of this articulated object are defined by the following kinematics equations:

$$f_1(\theta, \phi) = q_x \tag{6.1}$$

$$f_2(\theta,\phi) = q_y \tag{6.2}$$

$$f_3(\theta,\phi) = r_x\cos(\theta) - r_y\sin(\theta) + q_x \tag{6.3}$$

$$f_4(\theta,\phi) = r_x\sin(\theta) + r_y\cos(\theta) + q_y \tag{6.4}$$

$$f_5(\theta,\phi) = s_x(\cos(\phi)\cos(\theta) - \sin(\phi)\sin(\theta)) - s_y(\sin(\phi)\cos(\theta) + \cos(\phi)\sin(\theta)) + r_x\cos(\theta) - r_y\sin(\theta) + q_x \tag{6.5}$$

$$f_6(\theta,\phi) = s_x(\cos(\phi)\sin(\theta) + \sin(\phi)\cos(\theta)) + s_y(\cos(\phi)\cos(\theta) - \sin(\phi)\sin(\theta)) + r_x\sin(\theta) + r_y\cos(\theta) + q_y \tag{6.6}$$

where the kinematics functions $\{f_1(\theta,\phi),...,f_6(\theta,\phi)\}$ produces the values for the components $(p_{1,x}, p_{1,y}, p_{2,x}, p_{2,y}, p_{3,x}, p_{3,y})$ respectively, r and s are the original local co-ordinates for p_2 and p_3. The position of the object in the world is given by q. The kinematics parameters θ and ϕ are the angle of rotation on p_2 relative to p_1 and the rotation angle on p_3 relative to p_2 respectively (see Fig.6.1).

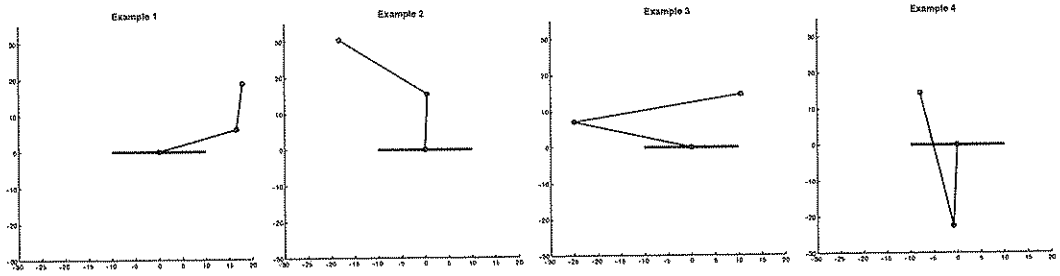


Figure 6.2: Four examples capturing the variations in the articulated object's three vertices. It can be observed that the examples themselves do *not* represent valid configurations of the articulated objects. For example, for all the examples, the lengths of the joints are different, violating the fact that the parts of the articulated object each have a fixed length. However, Figure 6.3 will show that certain linear combinations of these examples will generate valid configurations of the articulated object.

Suppose a set of training data that has been generated by different combinations of the kinematics parameters (θ,ϕ) exists. Using PCA as described in Chapter 5, the prototypes $\{\mathbf{e}_1,\mathbf{e}_2,...,\mathbf{e}_E\}$, where $E \leq 6$, can be obtained, as illustrated in Figure 6.2. These examples will be the axes of the E-dimensional coefficient space. The equations that all the coefficients must satisfy can be found

by projecting the kinematics equations into the normalised eigenspace, giving constraints:

$$c_i = \frac{1}{\lambda_i}(e_{i,1}f_1(\theta,\phi) + e_{i,2}f_2(\theta,\phi) + e_{i,3}f_3(\theta,\phi) + e_{i,4}f_4(\theta,\phi) + e_{i,5}f_5(\theta,\phi) + e_{i,6}f_6(\theta,\phi)) \quad (6.7)$$

where $i \in \{1,...,E\}$, $\mathbf{e}_i = \{e_{i,1}, e_{i,2}, ..., e_{i,6}\}$ and λ_i is the eigenvalue of $\mathbf{e}_i$. For all values of θ and ϕ, the i^{th} coefficient must satisfy Eq.(6.7) to reconstruct a valid form of the articulated object. The constraint surface for all the valid coefficient sets is visualised in Figure 6.3.

6.2.2 The Nonlinearity of Kinematics Constraints

It can be seen in Figure 6.3 that an articulated object's valid configurations can only be reconstructed by choosing coefficient sets that lie on a non-linear volume or surface in the coefficient space. The shape of this constraint surface (or high dimensional volume for a more complex representation type) is determined by the constraint equations, Eq. 6.7 in this case. Often, these constraint equations can be very complex. Consequently, it may not be realistic to explicitly model the constraint equations.

The Nature of Human Body Linear Combinations Coefficients

In the case where the constraint equations are not explicitly known, training data can be used to gain some insights into the characteristics of the valid coefficients. This requires the ability to recover the necessary coefficients for reconstructing each training example. In the previous chapter, we note that the examples are all orthogonal. Therefore, recovering the coefficients ($\mathbf{s}_i = (s_{i,1}, ..., s_{i,N_E})$) for reconstructing a training example ($\mathbf{t}_i$) can be achieved by projecting it onto the example vectors ($\mathbf{e}_1, ..., \mathbf{e}_{N_E}$):

$$\mathbf{s}_i = \mathbf{t}_i' E \quad (6.8)$$

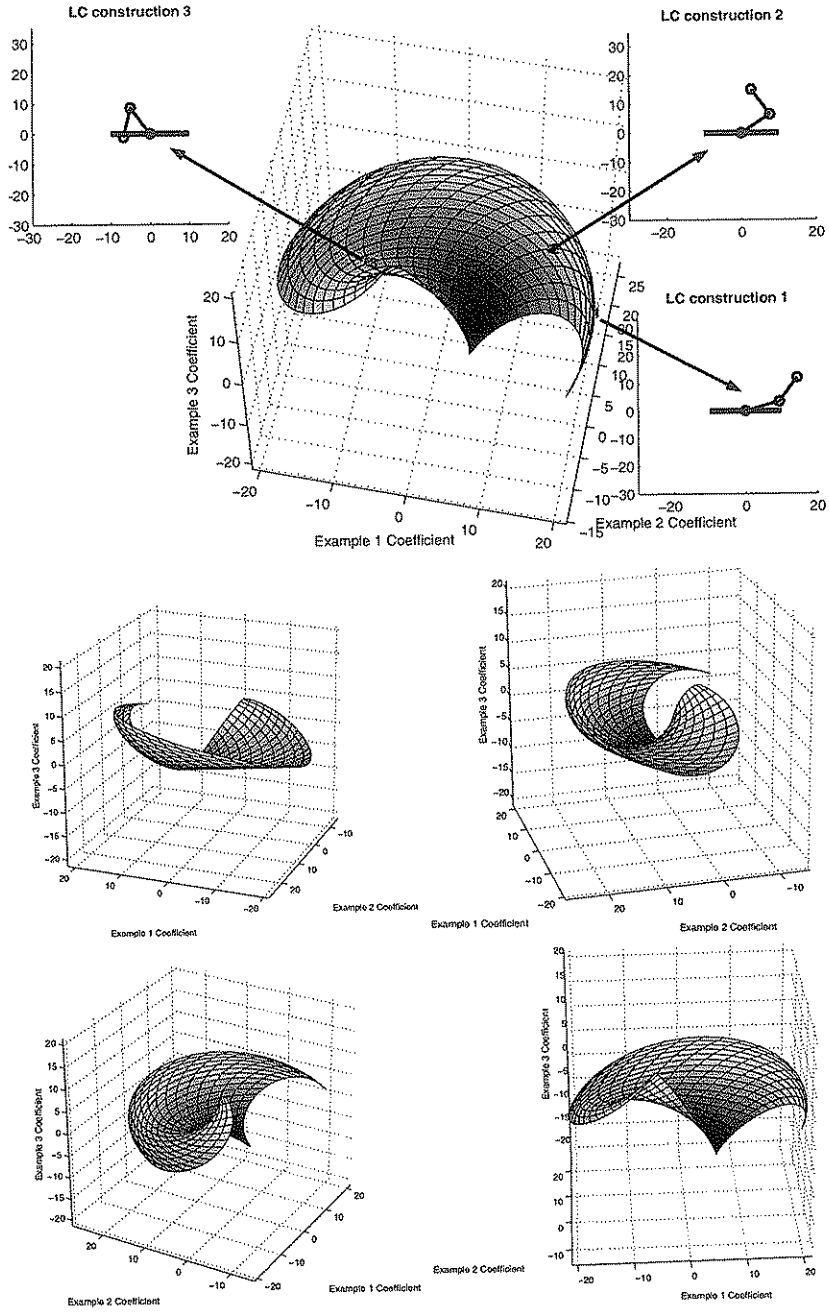


Figure 6.3: Visualisation of the constraint surface for Eq.(6.1) to Eq. (6.6). Displayed here are points with co-ordinates, (c_1, c_2, c_3), produced using different parameters θ and ϕ for the articulated object shown in Fig.6.1. The topmost figure shows the 3 instances of the articulated object. They were reconstructed by using points on the valid coefficients surface for linearly combining the examples shown in Figure 6.2. The middle and bottom part shows the surface at different viewpoints, allowing one to note the surface's non-linear characteristics.

where the example matrix is given by $E = (\mathbf{e}_1, ..., \mathbf{e}_{N_E})$.

Thus, we can now use this method to obtain the coefficients for reconstructing the data as possible human body configurations. This is given by the training hybrid vector set ($\mathbf{T} = (\mathbf{t}_1, ..., \mathbf{t}_N)$), which was described in Section 3.4. Each example is a hybrid vector of 240 dimensions, whilst 200 dimensions are contributed by shape information, 36 by the 3-D skeleton and 4 by the positions of both hands. A total of 1021 (i.e. $N = 1021$) training examples were obtained. Thus, with the 67 (i.e. $N_E = 67$) human body prototypical examples extracted in the previous chapter, we can apply Eq. 6.8 to recover the linear combination coefficients $(\mathbf{s}_1, ..., \mathbf{s}_N)$ for reconstructing the entire training data set. Since these coefficients represent instances of valid body configurations, we can also identify them as *valid coefficients*. As shown in Figure 6.4 the examples were sorted in descending order based on their magnitudes or the amount of linear variation in the hybrid vectors they account for. Consequently, the scale of the coefficient values decreases for the latter examples' coefficients.

The first few coefficient sets occupy a highly non-linear form. Any attempt to model the valid coefficients will therefore have to account for such a complex structure. Instead of attempting to explicitly model the constraints for the linear combinations based on explicit kinematics equations, the alternative of fitting a generic model onto the regions occupied by the valid coefficients has been adopted.

6.3 Cluster-Based Human Body Kinematics Constraints

As noted in the previous section, any coefficient constraints for valid body configurations would have to deal with subspaces of a non-linear structure. One powerful tool for tackling with this problem is a cluster model [10]. Here, a set of piecewise clusters is used to approximate the valid coefficient values, as illustrated in Figure

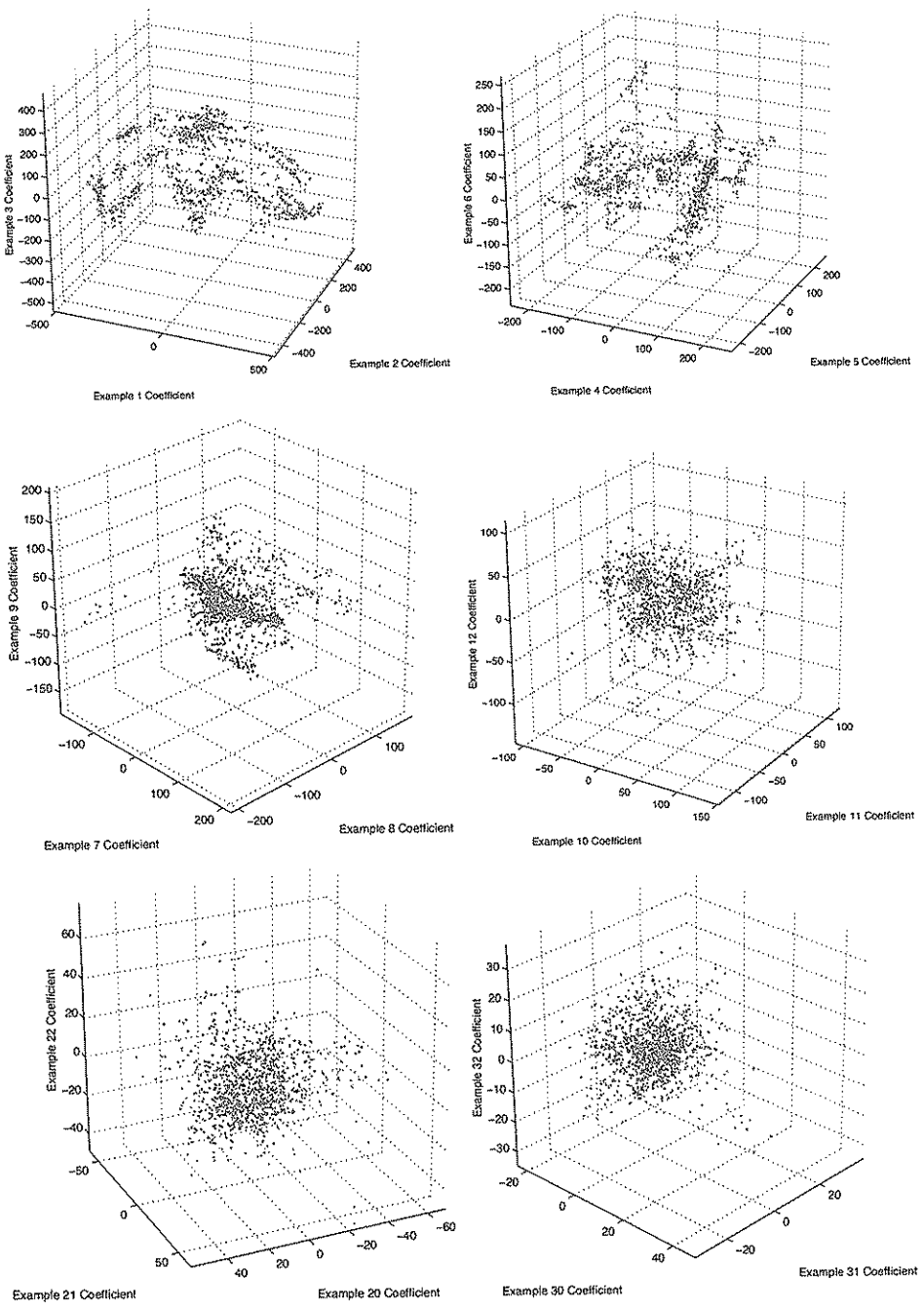


Figure 6.4: An illustration of the *valid coefficients* used for reconstructing the training hybrid vectors by linearly combining the prototypical examples.

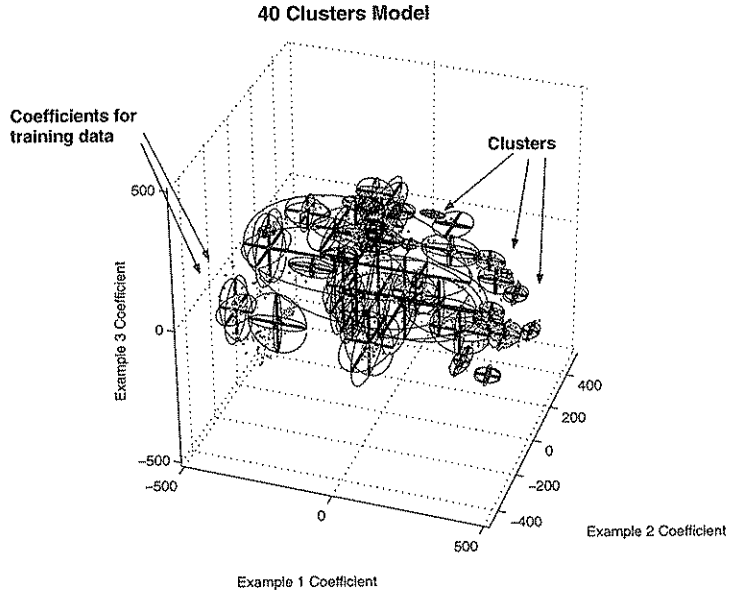


Figure 6.5: A cluster model for the linear combination coefficients

6.5.

Each cluster captures a subset of the linear combinations coefficients that can be used to reconstruct information on a body configuration. As each individual cluster only captures a *linear* portion of the coefficient space, it can be thought of as a model for a restricted range of linear human body motions, as illustrated in Figure 6.6.

Additionally, the more non-linear the motions a body is capable of, the more non-linear the shape valid coefficients will have to take. This would in turn increase the number of required linear clusters. Therefore, it can be reasoned that the required number of clusters can be used to indicate the degree of nonlinearities of the human body.

Formally, we can define a cluster model ($\mathbf{C}$) as a set of clusters, $(\mathbf{c}_1, ..., \mathbf{c}_{N_C})$, where N_C denotes the number of clusters. Since they are constraints for the coefficients of the linear combinations, the dimensionality of the space in which the clusters exist is equal to the number of prototypical examples (N_E) used.

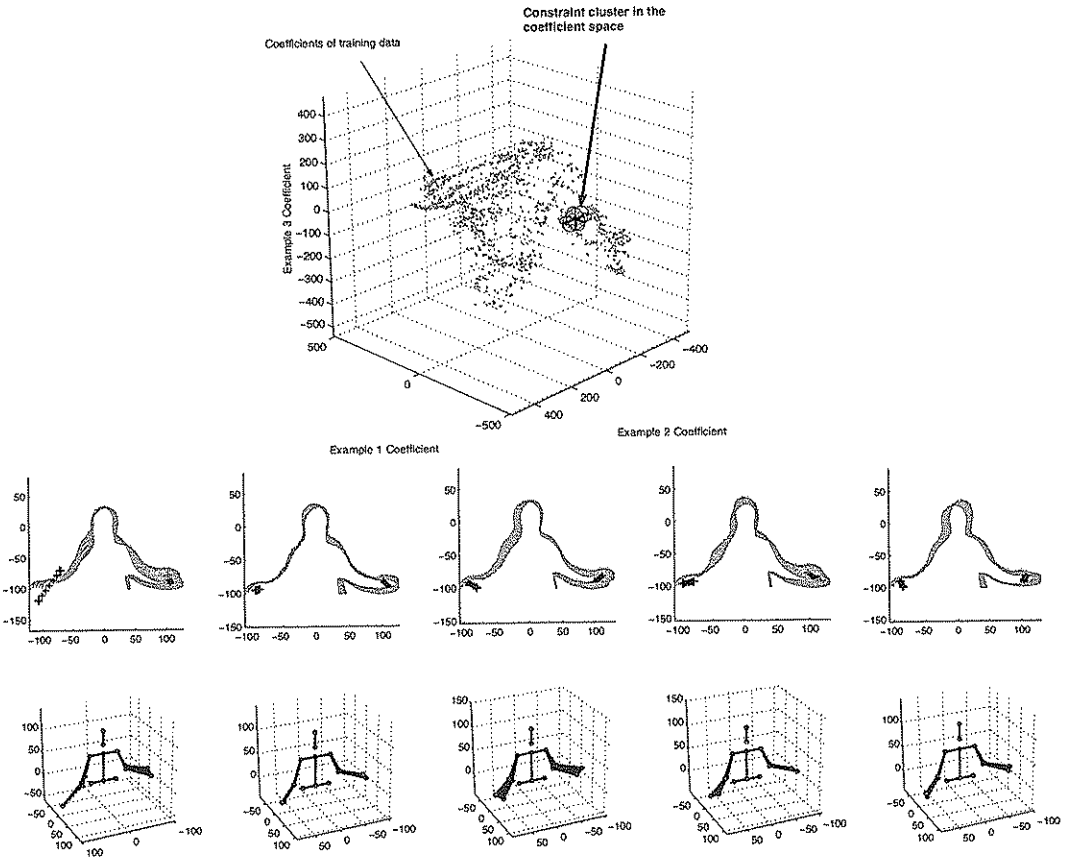


Figure 6.6: An illustration of the different sets of linear motions captured by a single cluster in the coefficient space.

Individually, each cluster ($\mathbf{c}_i$), where i is the index of the cluster, contains a number of parameters:

$$\mathbf{c}_i = (\boldsymbol{\mu}_i, N_{P_i}, \mathbf{P}_i, \boldsymbol{\Lambda}_i) \tag{6.9}$$

The mean position of the cluster is defined by a N_E dimensional vector ($\boldsymbol{\mu}_i$). Additionally, the cluster-shape is parameterised by a number (N_{P_i}) of N_E dimensional *normalised* principal axes vectors ($\mathbf{p}_1, ..., \mathbf{p}_2$), arranged column wise into a matrix $\mathbf{P}_i$. The corresponding axes variances are given by vector $\boldsymbol{\Lambda}_i$:

$$\mathbf{P}_i = (\mathbf{p}_1, ..., \mathbf{p}_{N_{P_i}}) \tag{6.10}$$

$$\boldsymbol{\Lambda}_i = (\lambda_{i,1}, ..., \lambda_{i,N_{P_i}}) \tag{6.11}$$

By setting the cluster shape covariance matrix to different forms, three types of clusters can be created: radial, diagonal covariance and full covariance clusters. In particular, the radial clusters' covariance matrix takes the form of a scaled identity matrix, sI. In a high dimensional space, the scale can be very large, as the cluster extends across different dimensions. This causes the cluster to span too large an area when the data is not spherical. Next a, diagonal covariance cluster derives its name from having a diagonal covariance matrix, where only the diagonal elements are non-zero. This allows the cluster to have different sizes along the different dimensions it exists in. However, its axes are restricted to be in the same directions as the major axes of the co-ordinate space in which it exists. Finally, the cluster with the most flexibility is that with a full covariance matrix, where all the elements are non-zero. However, for a high dimensional space, a full covariance cluster suffers from the need to determine a great number of parameters (i.e., all the elements of the covariance matrix).

Commonly, a diagonal covariance cluster is adopted, as it provides enough flexibility to account for non-spherically distributed data, while not containing as many parameters as the full covariance cluster. Consequently, all the axes of the clusters $(\mathbf{p}_1, ..., \mathbf{p}_2)$ are unit vectors aligned with the major axes of the coefficient space (e.g. $\mathbf{p}_1 = 1, 0, 0, ..., 0$, $\mathbf{p}_2 = 0, 1, 0, ..., 0$ and so on).

The usefulness of a collection of clusters comes from the ability of individual components (the clusters) to adopt independent positions and capture different sized regions. This allows even very highly non-linear spaces to be captured. However, this flexibility has disadvantages. For instance the complexity (number of clusters) and the structure (cluster parameters) of the LC coefficient constraint model must be determined. The next section will describe the method for learning the parameters of the clusters.

6.4 Learning the Linear Body Motions

To determine the parameters of clusters (mean and principal axes), a two stage procedure is adopted. The first step involves initialising the cluster with the K-means algorithm. Following this, the parameters are then refined using Expectation Maximisation (EM).

6.4.1 Initialising Cluster Parameters

In order to seed the initial parameters of a cluster model, a simple but effective method called *K-means* [35, 36] is adopted. This method allows one to iteratively calculate the mean and shape parameters of the clusters given a set of training data, ($\mathbf{T} = (\mathbf{t_1}, ..., \mathbf{t_2})$). At every iteration, each cluster is assigned a set of points that are nearer to it than any other clusters. Such a procedure is repeated until convergence, that is, the centres of the clusters do not deviate with more iterations. Having located the centres of each cluster, its shape can be estimated from the covariance matrix of the data the cluster accounts for. The algorithm is given below:

1) Assign the mean of each cluster randomly to a particular training example.

2) Assign each cluster ($\mathbf{c}_i$) the set of N_i training examples $\{\mathbf{t}_{i,1}, ..., \mathbf{t}_{i,N_i}\}$ that satisfies the minimum Euclidean distance rule (i.e. these N training examples are closer to this cluster than any others).

3) Compute the new mean position $\boldsymbol{\mu}_i$ for each cluster ($\mathbf{c}_i$) based on the training examples assigned to it.

$$\boldsymbol{\mu}_i = \frac{1}{N_i} \sum_{j}^{N_i} \mathbf{t}_{i,j} \tag{6.12}$$

4) If the mean positions of any clusters have changed, go to step 2 again.

5) Finally, compute the cluster variances along each of the coefficient space

dimensions using the cluster's final set of assigned training data:

$$\lambda_{i,j} = \frac{1}{N_i - 1} \sum_{k}^{N_i} (t_{i,j,k} - \mu_{i,k})^2 \tag{6.13}$$

where $j = \{1, ...N_E\}$ and N_E is coefficient space dimensionality.

6.4.2 Refining the Parameters: Expectation Maximisation (EM)

An alternative approach to determining the parameters of the clusters can be achieved by treating each cluster as a probability function. Formally, one can think of a cluster ($\mathbf{c}_i$) in an N_E dimensional space, at position $\boldsymbol{\mu}_i$, and covariance matrix $\mathbf{C}_i$, as a Gaussian distribution function (p_i):

$$p(\mathbf{t}|i) = \frac{1}{(2\pi)^{N_E/2}(det\mathbf{C}_i)^{1/2}} exp\left(-\frac{1}{2}(\mathbf{t} - \boldsymbol{\mu}_i)\mathbf{C}_i^{-1}(\mathbf{t} - \boldsymbol{\mu}_i)\right) \tag{6.14}$$

The determination of its parameter can then be treated as a maximum likelihood problem, modelled by a mixture of probabilistic clusters. Maximum likelihood can be computed iteratively using the Expectation Maximisation (EM) algorithm [7, 10]. This approach was developed for handling conditions when the *observed* data (i.e. training data) is incomplete. Haykin [73] defines "incomplete" as follows:

- The existance of two sample spaces X and Y represented by the observed data vector $\mathbf{x}$ and the complete data vetor $\mathbf{y}$, respectively.
- There is a one-to-many mapping from space Y to space X.

For completeness, a description of the EM algorithm will now be given. The algorithm consists of two major steps, the Expectation step and the Maximisation step, leading to the name of the algorithm. Generally, the purpose of the Expectation step is to compute probabilistically, how well a model fits the observed data. Here, the cluster set defines the model. The fitness is quantified by

the log likelihood of the cluster model, given the training data. Meanwhile, the Maximisation step is responsible for maximising the fitness of the cluster model by adjusting its parameters. Specifically, the EM algorithm can be described as follows:

Expectation

Calculate the posterior probabilities of each cluster for each training example using the cluster's current parameters:

$$P(i|\mathbf{t}^n) = \frac{p(\mathbf{t}^n)P(j)}{p(\mathbf{t}^n)} \tag{6.15}$$

where $P(j)$ is the prior of the cluster given in the next step, $p(\mathbf{t}^n)$ is:

$$p(\mathbf{t}^n) = \sum_{j}^{N_C} p(\mathbf{t}^n|j)P(j) \tag{6.16}$$

The cluster density $p(\mathbf{t}^n|j)$ is given in Eq. (6.14).

Maximisation :

The maximisation step is performed by evaluating the following three equations, which aims to adjust the parameters of a cluster $(\mathbf{c}_i)$ to maximise the model's fit to the training data set $(\mathbf{T})$ of N_T training examples:

$$\boldsymbol{\mu}_i^{new} = \frac{\sum_n^{N_T} P^{old}(i|\mathbf{t}^n)\mathbf{t}^n}{\sum_n^{N_T} P^{old}(i|\mathbf{t}^n)} \tag{6.17}$$

$$C_{i,i}^{new} = \frac{1}{N_E}\frac{\sum_n^{N_T} P^{old}(i|\mathbf{t}^n)||(\mathbf{t}-\boldsymbol{\mu}_i^{new})||^2}{\sum_n^{N_T} P^{old}(i|\mathbf{t}^n)} \tag{6.18}$$

$$P(i)^{new} = \frac{1}{N}\sum_{n}^{N_T} P^{old}(i|\mathbf{t}^n) \tag{6.19}$$

6.5 Learning Human Body Kinematics Constraints

In order to learn the human body kinematics constraints, cluster models were built to capture the valid linear combination coefficients. To determine the positional and size parameters of the clusters, the K-means and EM method described in

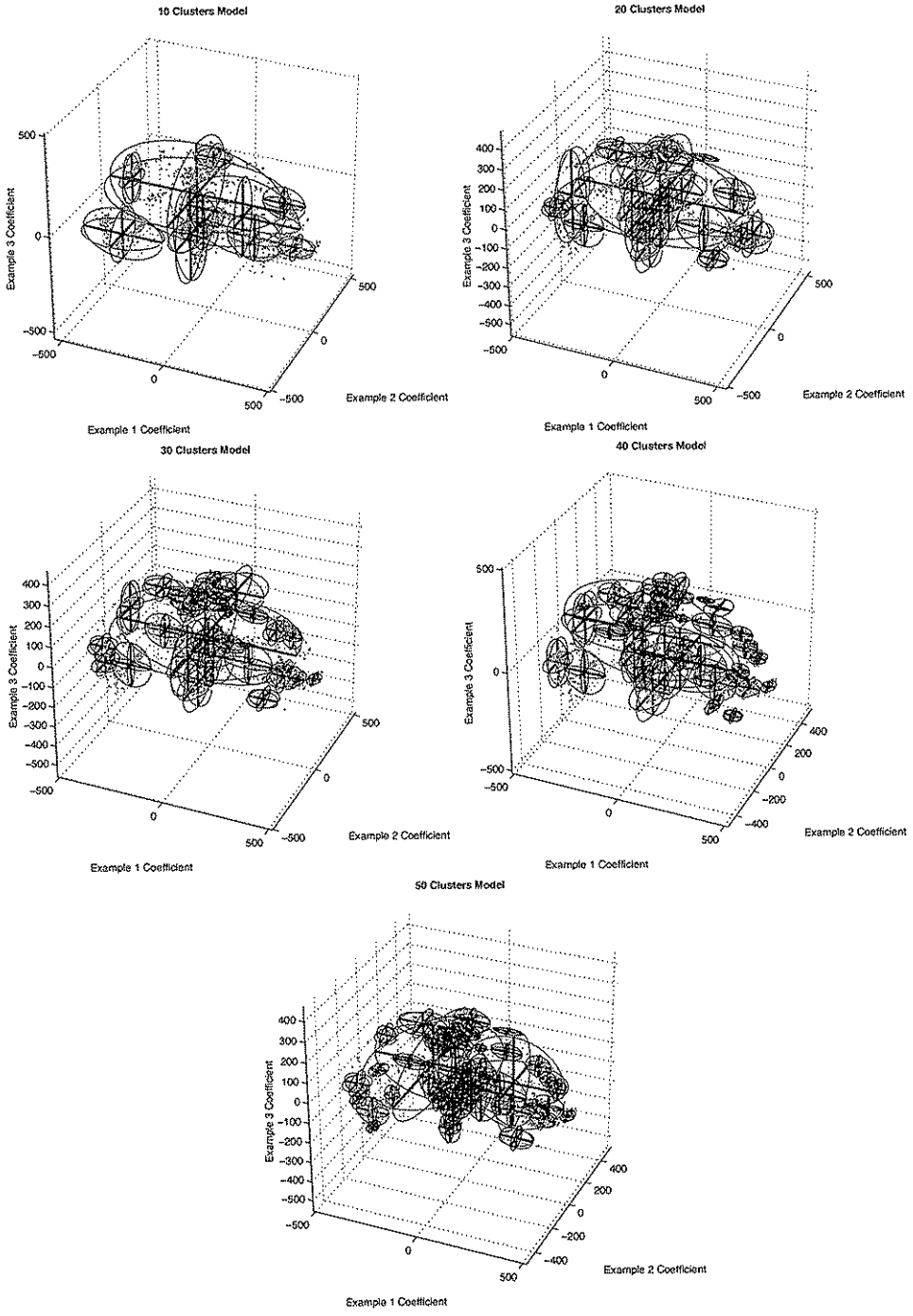


Figure 6.7: A visualisation of different cluster models capturing the valid linear combination coefficient values for the first three examples.

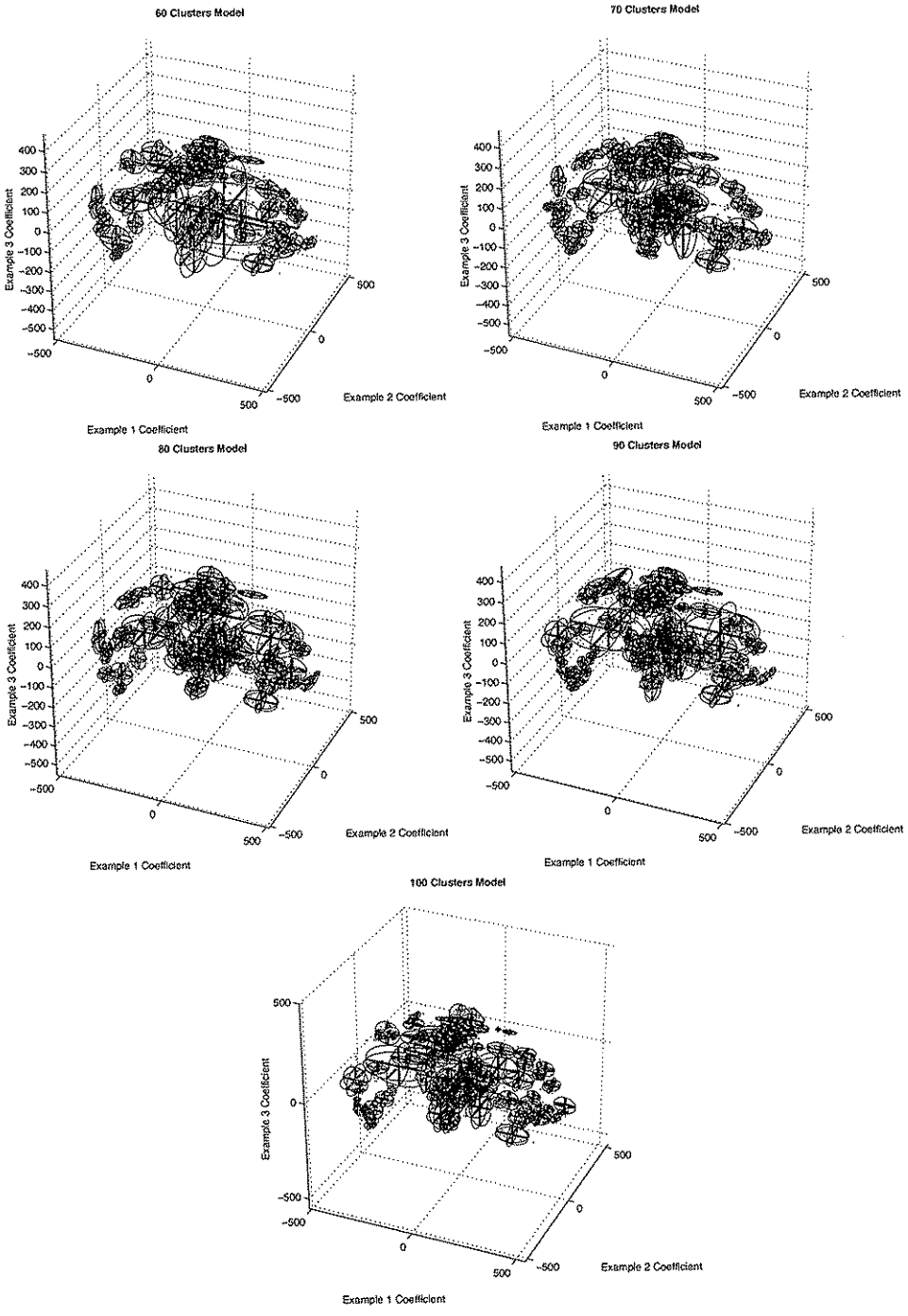


Figure 6.8: Further cluster models capturing the valid linear combination coefficient values for the first three examples.

the previous section were used. However, the required number of clusters is not determined by this method. It is therefore not clear how the accuracy of the constraint model differs when different numbers of clusters are used. Consequently, a number of cluster models, each with a different number of clusters were built. Specifically, 10 different cluster models with 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 clusters were built, as shown in Figure 6.7 and Figure 6.7. These figures show the cluster models in the space of the coefficients for the three largest examples. Additionally, the valid set of coefficients ($\mathbf{P}$) described in Section 6.2.2 is shown as dots in each cluster model, indicating the regions of valid coefficients.

The average size of clusters in each cluster model can be estimated by calculating the covariance diagonals average of all the clusters for a model. Formally, for a cluster model with a number of clusters, N_C, where each cluster $\mathbf{C}_i$ has a set of covariance diagonals $\{d_{i,1}, ...d_{i,N_E}\}$, the average cluster size (S) for the cluster model can be calculated by:

$$S = \frac{\sum_i^{N_C} \sum_j^{N_E} d_{i,j}}{N_E N_C} \tag{6.20}$$

where, $N_E = 67$, is the number of prototypical examples determined in Chapter 5. The results of the average cluster sizes (S) for cluster models with different numbers of clusters are shown in Figure 6.9.

It can be clearly seen from the cluster diagrams that as the number of clusters increases, the size of the individual clusters decreases. The effects the decrement in cluster sizes can be seen by visualising the variations captured by a single cluster. In order to accomplish this, hybrid vectors that can be reconstructed from coefficients encompassed within a typical cluster are shown (see Figures 6.10 and 6.11). In this figure, it can be seen that as the number of clusters increases, the hybrid vector variations captured by each cluster becomes increasingly specific.

It can also be observed that as the number of clusters increases, the shape of the space encompassed by the cluster model starts to resemble more accurately

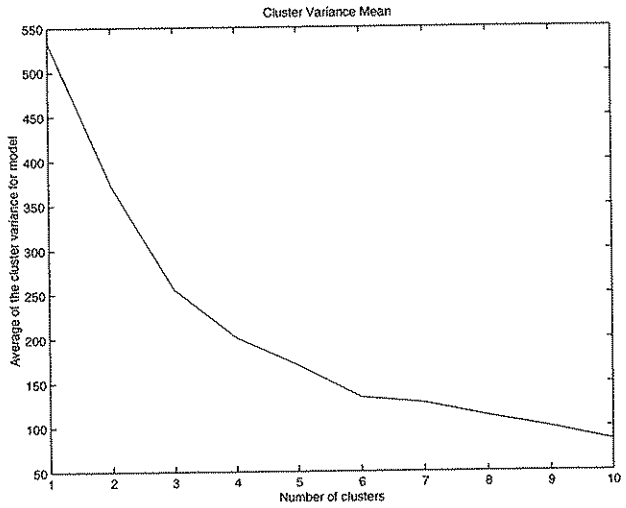


Figure 6.9: This graph shows the average of the cluster variances (S) as defined in Eq. (6.20) for cluster models of different sizes. The vertical axes shows the values of (S) while the horizontal axes shows the number of clusters. As the number of clusters increases, the variance of each cluster generally decreases.

the regions traced out by the valid coefficients. To provide a quantification of the accuracy of the cluster models in modelling the valid coefficients as the number of clusters increases, experiments aimed at quantifying the reconstruction from hybrid vectors with missing information were carried out. In real situations, the visual appearance information will be available, therefore, to make the experiments useful, the missing information was set as the 3-D skeletons. The next section will describe the process by which 3-D skeletal information can be reconstructed from available visual information and cluster models.

6.6 Reconstructing the 3-D Skeletal Information

The cluster models can also be used for reconstructing "missing" information in the hybrid vector. Intuitively, a hybrid vector containing incorrect information (e.g. inconsistent visual and 3-D skeleton information) will result in a set of linear combination coefficients which is also invalid. Since each cluster is assumed to model a region of valid coefficients, any invalid hybrid vectors will contain

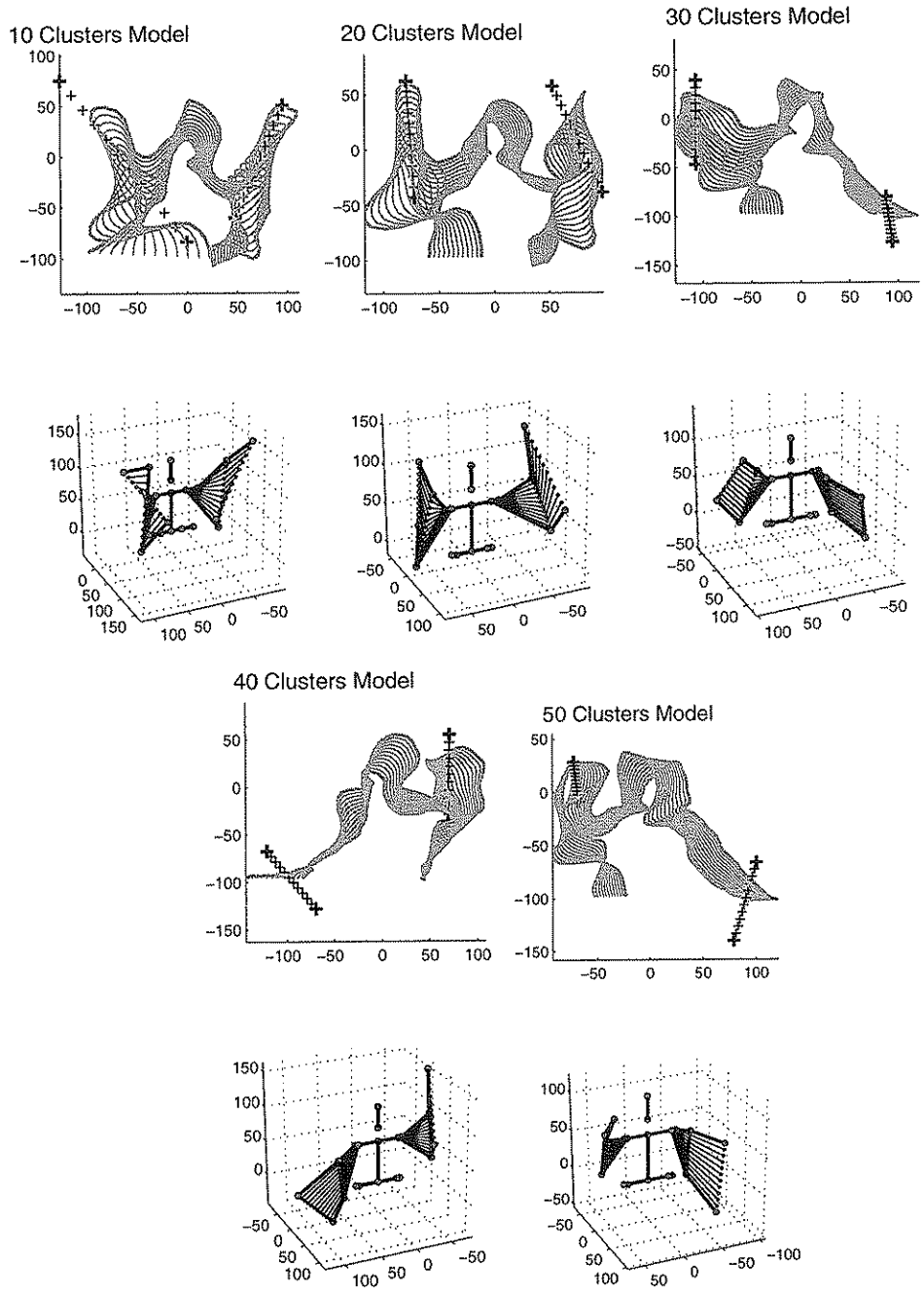


Figure 6.10: The magnitude of variations captured by the cluster based constraints as the number of clusters increases. The hybrid vector variations captured by a cluster picked randomly in each cluster model are shown.

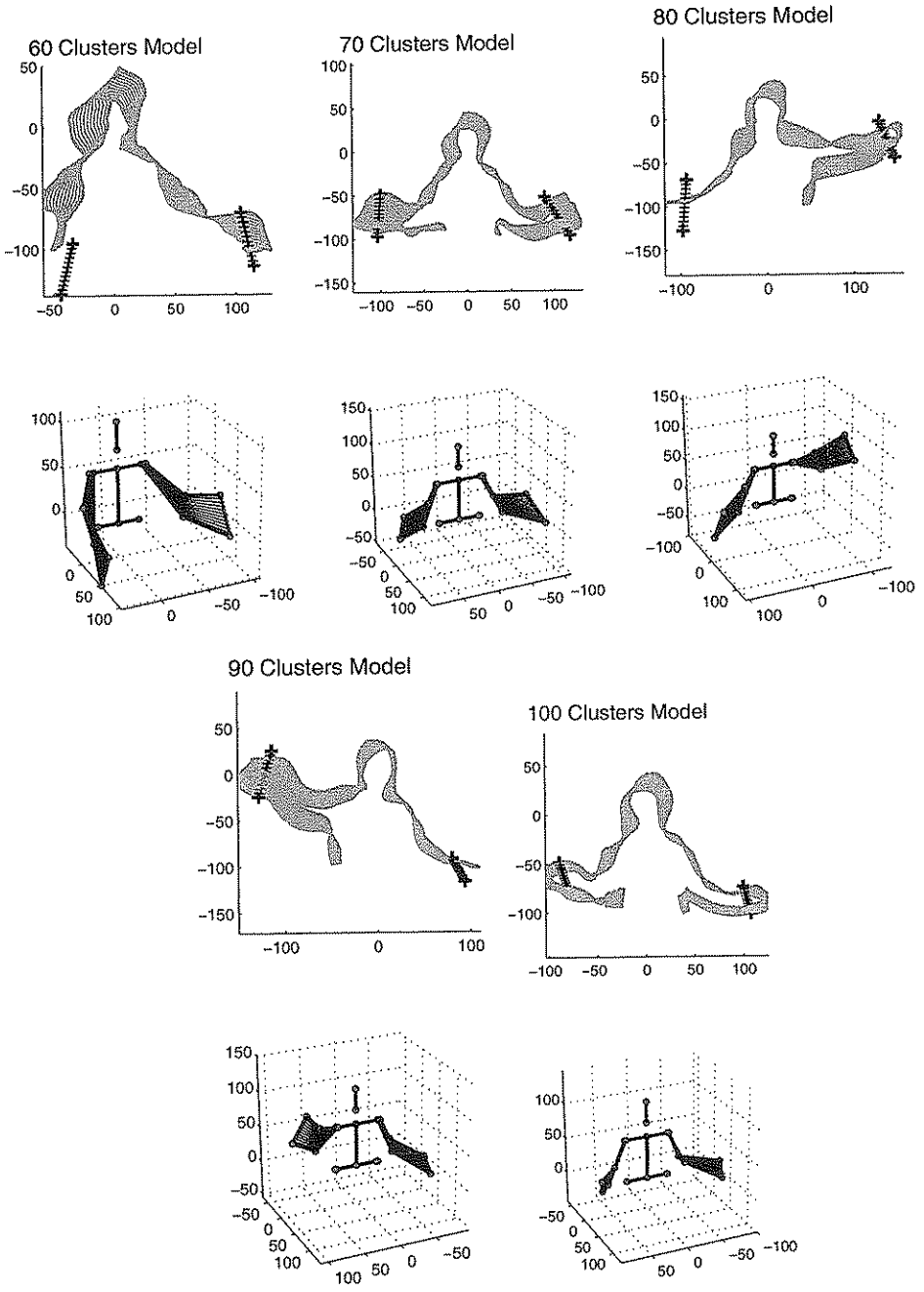


Figure 6.11: Further illustrations of the magnitude of variations captured by the cluster based constraints as the number of clusters increases. Again, the hybrid vector variations captured by a cluster picked randomly in each cluster model are shown.

coefficients which fall *outside* the cluster space. In order to find the closest valid coefficient set, it is advantageous to think of a set of linear combination coefficients as a *coefficient vector* in a *coefficient space* (i.e. the coefficient values are the co-ordinates of the coefficient vector). The dimensionality of the coefficient space is equal to the number of examples. The closest valid coefficient vector ($\mathbf{p}_V$) can then be found by choosing a point on a cluster that is closest to the original invalid coefficient vector ($\mathbf{p}_X$) [81].

In order to do this, it is necessary to obtain a closest coefficient vector ($\mathbf{p}_{V,i}$) which lies on a given cluster (C_i). This can be achieved by firstly projecting $\mathbf{p}_X$ onto the principal axes ($\mathbf{P}_i$) of the cluster with the centre, $\boldsymbol{\mu}_i$, to obtain the projection vector, $\mathbf{r}_i$:

$$\mathbf{r}_i = (\mathbf{p}_X - \boldsymbol{\mu}_i)'\mathbf{P}_I \tag{6.21}$$

Each element of the projection vector ($\mathbf{r}_{i,j}$) is limited to lie within the range of $-\lambda_{i,j}$ to $\lambda_{i,j}$, where $\lambda_{i,j}$ is the variance for the j^{th} principal axes on the i^{th} cluster. Any element outside the bounds of this range is set to $-\lambda_{i,j}$ or $\lambda_{i,j}$ respectively. The closest coefficient vector ($\mathbf{p}_{V,i}$) for the i^{th} cluster can be obtained by reconstructing from the projection vector:

$$\mathbf{p}_{V,i} = \mathbf{P}_i\mathbf{r}_i + \boldsymbol{\mu}_i \tag{6.22}$$

Finally, the closest coefficient vector ($\mathbf{p}_V$) on the entire cluster can be obtained by choosing the cluster coefficient vector ($\mathbf{p}_{V,j}$) that minimises the distance to the original invalid point ($\mathbf{p}_X$):

$$D_j = \min_j(\mathbf{p}_{V,j} - \mathbf{p}_X)^2 \tag{6.23}$$

$$\mathbf{p}_V = \mathbf{p}_{V,j} \tag{6.24}$$

where D_j is the distance between the new coefficient vector and the original coefficient vector.

Using Eq. (6.21) to Eq. (6.24), an experiment for analysing the reconstruction or "correction" capability of different cluster models was carried out. First, a new hybrid training data set was constructed from the original training data set. In the new set, the 3-D skeleton components were set to 0, while the contour and hand positions components were not modified. The linear combination coefficients of the new training set was then acquired by projecting the training data to the prototypical examples (see Eq. (6.8)), yielding an *invalid training coefficient set.* Second, using a cluster model, the invalid training coefficient vectors were corrected by obtaining $\mathbf{p}_V$ for each training coefficient set. The corrected hybrid vectors were then regenerated by linearly combining the prototypical examples using the corrected coefficients. The sum of all the distances between the corrected hybrid vectors and the original training set (i.e. with the 3-D skeleton) was obtained for the cluster models with different number of clusters. The resulting reconstruction distance or reconstruction error graph can be seen in Figure 6.12. It is clear that as the number of clusters increases, the reconstruction capability's accuracy increases.

6.7 Conclusions

In this chapter, it was discovered that a constraint model for restricting the linear combination coefficients of articulated objects must cope with non-linear subspaces. This observation initially came from analysis of the linear combinations coefficients for a simple articulated object. A set of linear combination coefficients for generating a single object example was defined as a *coefficient vector* in a *coefficient space.* A coefficient vector that generates a valid instance of the articulated object can be defined as a valid coefficient vector. Visualisation of the coefficients revealed that the articulated object's valid coefficient vectors fall on a non-linear constraint surface.

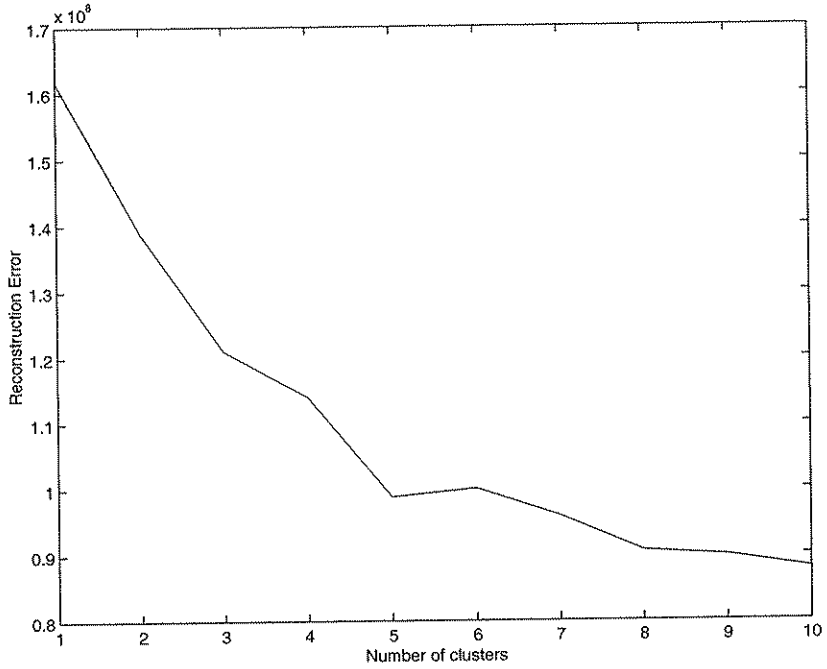


Figure 6.12: The graph shows the error in reconstructing the 3-D skeleton component of a hybrid vector given only the contour and body part positions using cluster-based constraints with increasing complexity (number of clusters).

Consequently, the structure of the valid coefficients for the human body hybrid vectors was similarly investigated. Visualisation of the hybrid vectors' valid coefficients similarly revealed a highly non-linear subspace. In order to model such a non-linear structure, a model composed of a set of clusters in the coefficient space was chosen. Each cluster accounted for a subset of linear combination coefficients, or alternatively, a subset of body configurations. The subset of body configurations assigned to each cluster was determined by the Expectation Maximisation (EM) clustering algorithm.

This cluster model can be used to reconstruct hybrid vectors containing missing information (e.g. the 3-D skeleton components are all 0, instead of 3-D vertex co-ordinate values). This was because, firstly, an incomplete hybrid vector does not constitute a valid instance of such a representation. Therefore, it would subsequently not yield a valid set of linear combination coefficients, or a valid coefficient

vector. An invalid coefficient vector would then fall outside the space occupied by the cluster model. Secondly, it is assumed that all the subspaces occupied by the clusters are valid. Therefore, a valid coefficient vector can instead be obtained by finding a point on the cluster model that is closest to the original invalid coefficient vector.

It was found that as the number of clusters was increased, the size of the clusters decreased (i.e., each cluster modelling an increasingly specific set of body configurations). The increasing specificity of the clusters was found to be advantageous when cluster constraints were used to recover missing information in hybrid vectors. This was empirically supported by an experiment performed to test the reconstruction accuracy of models with differing number of clusters. It was found that the larger the number of clusters, the lower the reconstruction error.

However, it has to be noted that such an experiment is not a good method for determining the optimal number of clusters required for capturing the body configurations' kinematics constraints. This is because only the training data was used for the reconstruction experiment. Consequently, it does not provide any indication on the generalisation capability of the clusters to capture valid body configurations that are *not* in the training set. This issue will instead be dealt in detail in Chapter 8.

Additionally, the reconstructions were performed on the assumption that the body contour information and hand positions were both available and reliable. Such an assumption may not be valid in most circumstances where it is not possible to obtain the contour or hand positions reliably (e.g. in a cluttered background). Therefore, in the next two chapters, we will deal with tracking the body configurations without explicit knowledge of the contour or hand positions. To this end, Chapter 8 will develop a dynamic platform which can be used to visually track body configurations using the prototypical examples and the cluster based constraints described here. One notes that the prototypical examples and

its cluster constraints only account for possible body configurations. It does not, however, account for the dynamics of the body configurations, or the way the coefficients evolve over time as the body configuration changes. Such dynamical knowledge would be necessary in a visual tracking platform as it allows for the prediction of future body configurations. To address this problem, the next chapter will be concerned with learning the dynamics of the hybrid vectors from available data.

Chapter 7

Learning Human Body Dynamics

In the previous chapters, a representation for the human body configuration in the form of a hybrid vector has been adopted. The spatial characteristics of the human body configurations were learnt by modelling a training set of the hybrid vectors in terms of prototypical examples. The prototypical examples accounted for variations in the visual and structural parameters of the human body. Information on novel body configurations can be generated by linearly combining the prototypical examples. Thus, a body configuration is described using a set of linear combination coefficients instead of the entire hybrid vector form.

Nevertheless, the linear combinations framework in itself does not provide constraints on the possible linear combinations. Therefore, hybrid vectors representing invalid body configurations can be generated. One solution for this problem is to impose kinematics constraints on the allowed human body configurations. To this end, cluster based constraints were exploited to restrict the possible linear combination coefficients.

This resulted in a constrained linear combinations framework as a computational model for reconstructing the visual and structural information of the human body. However, this constrained linear combinations framework still does not account for the dynamics of the human body motion patterns. Specifically, there is

no information on how the linear combination coefficients transform as the body configuration changes. Knowing the dynamics of coefficients transformation can add further constraints to reduce ambiguities when the coefficients are visually tracked from images.

There is the need to have a model for representing our knowledge on the body motion pattern dynamics. Such an undertaking would firstly require a computational quantification of such "knowledge". Additionally, a method for learning its parameters is also necessary. Therefore, this chapter will deal with learning more about the dynamics of the hybrid vectors when it is used to represent the configuration of the human body. The nature of the dynamical characteristics of the hybrid vector is firstly given in Section 7.1. We show how the visual information of a human body can sometimes exhibit discontinuous dynamics. An example of such a phenomenon is when the configuration of the body changes slightly while the visual information undergoes large deviations.

Next, with an understanding of the hybrid vector dynamics' characteristics, a computational model can be defined. A straightforward but effective method of a transition matrix is employed as is described in Section 7.2. Since the hybrid vector represents the body configuration, the transition matrix can be thought of a mechanism for computationally capturing the dynamics of the human body motion patterns. Following this, experimental results on the recovered global transition matrix along with a visualisation of different discontinuous behaviours modelled will be shown in Section 7.3. Finally, a summary is given and are conclusions drawn in Section 7.4.

7.1 Discontinuities in the Visual Observations

In order to understand the dynamical nature of the linear combination coefficients, an analysis on the dynamics of a hybrid vector sequence is performed. This se-

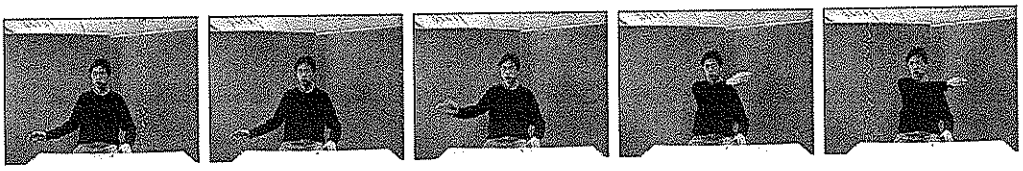


Figure 7.1: The images of a human body undergoing a continuous gesture.

quence follows a human body undergoing a continuous gesture (see Figure 7.1). The coefficients of the hybrid vectors were obtained by projecting the training data onto the prototypical examples extracted in Chapter 5. As the human body is undergoing a continuous change, it would be natural to assume that the coefficients would change smoothly as well. However, contradictions to this smoothness assumption was found when the acceleration in the coefficient vectors' magnitudes were plotted (see Figure 7.2). It can be seen that at various points on the graph, there is a large change in the speed magnitude. These points indicate the possible existence of discontinuous dynamics in the linear combination coefficients.

7.1.1 Ruling Out Skipped Frames

It may be that sudden large changes detected in the linear combination coefficients can be due to large changes in the body configuration. One common cause for such large changes is the skipping of a few frames during the sequence acquisition process. In order to rule out the effect of discontinuities caused by skipped frames, the magnitudes of the difference vectors between successive 3-D skeletons were first analysed. The effects of frames being skipped can be detected in regions where the 3-D skeleton difference vectors' magnitude is large. Following this, we only use segments of the sequence analysed where the difference vector's magnitude is not very large (see Figure 7.3).

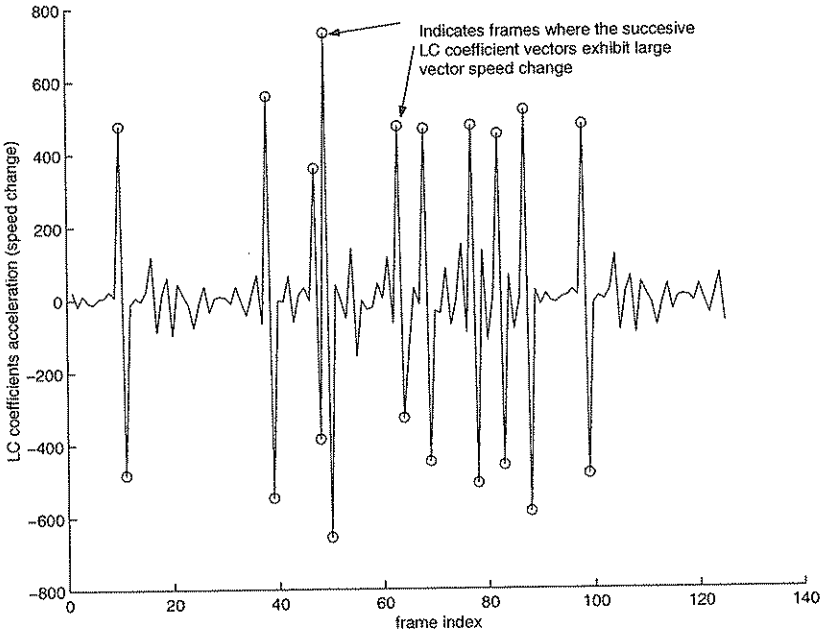


Figure 7.2: A graph showing the acceleration of the linear combination coefficient vector speed in a continuous gesture sequence. Where the coefficients suddenly changes, a large zero crossing will occur. Circles on the graph indicate these.

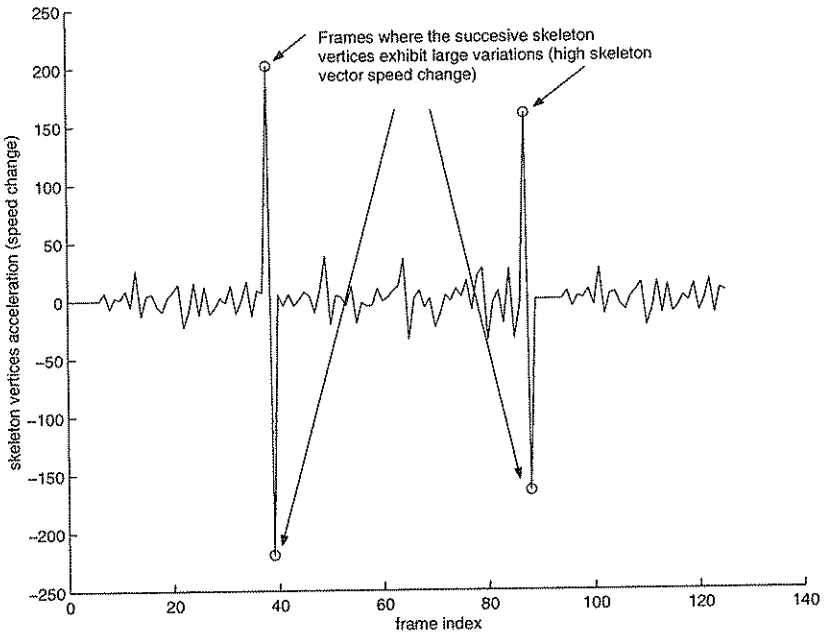


Figure 7.3: The acceleration of the 3-D skeletons vertex vector in a continuous gesture sequence. Where the speed of the 3-D skeleton suddenly changes, a large zero crossing will occur. Circles on the graph indicate these.

7.1.2 Discontinuities in the Body Contour

Evidence for discontinuous dynamics in the linear combination coefficients can be found by combining results of the analysis of both the 3-D skeleton accelerations and the coefficient vectors' acceleration. In Figure 7.2, it can be seen that there are points in the sequence where the skeleton changes slightly but the coefficients undergo large variations. In order to determine the cause of these discontinuities, the different components of the hybrid vectors with large speed change in the linear combination coefficients were shown. In Figure 7.4, it can be seen that the contour components at regions where the linear combination coefficients exhibit large speed variations was found to have changed significantly. A similar phenomenon was reported by Heap [81], where PDM contours were used to represent the shape of hands. There, it was found that at certain hand poses, a small deviation in the hand rotation or finger movements respectively would cause large changes in the contour.

The large shape variations were caused by the lack of correspondence between the contour vertices across different object shapes. This causes the position of the contour vertices to be dependent on the contour's length. It was found that there exist two situations where the length of the contour is likely to change drastically. The first situation involves the nature of the silhouette of the human body. Since the silhouette only follows the outline of the body shape, it only accounts for the outermost parts. The shapes from different parts previously responsible for the outline overlap, resulted in a silhouette changing considerably (see Figure 7.5a). The second situation for a large change in the shape's length originates from occlusions on the contour. This in turn resulted from the nature of the contour acquisition process. Since the contour is only acquired from a certain region of the image upwards, any shape contribution below is discarded, causing a large change in the contour shape (see Figure 7.5b).

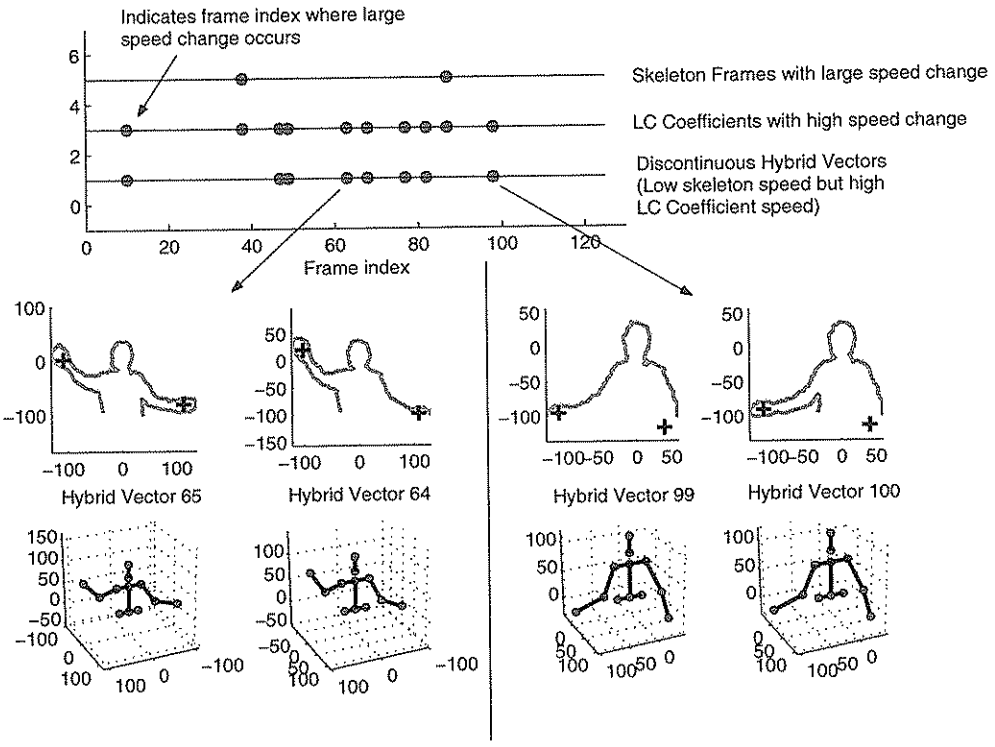
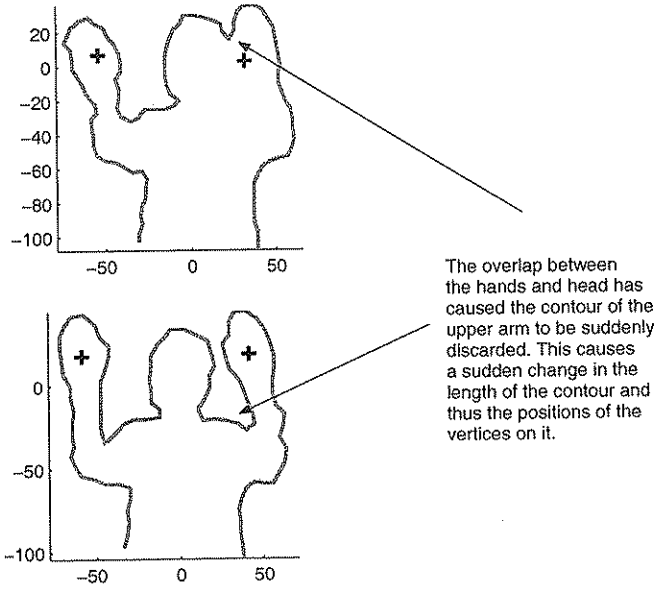


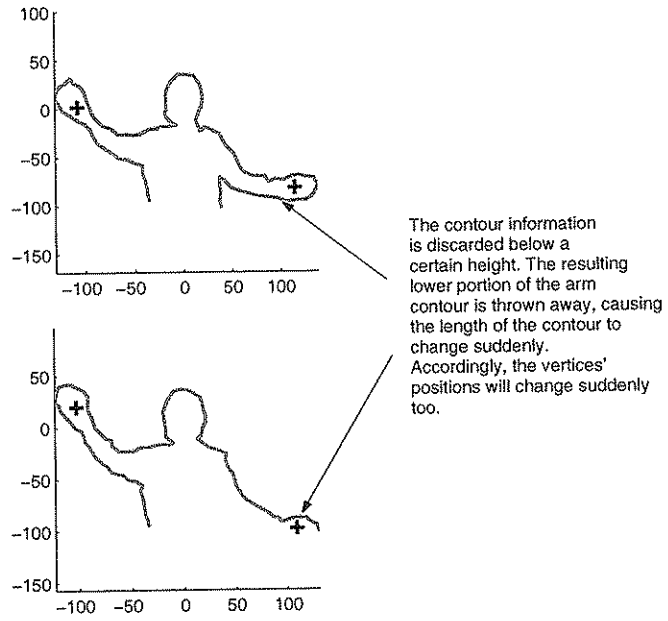
Figure 7.4: An illustration of the hybrid vectors which can cause discontinuous dynamics in the linear combination coefficients.

7.2 Global Dynamics: Transition Matrices

We have shown the problem and cause of discontinuous dynamics in the linear combination coefficients. The discontinuities occurred when the body configuration undergoes small changes, while the respective representing linear combination coefficients undergo a large deviation. Alternatively, one can think of such large linear combination coefficient deviations as a "jump" between two coefficient subspaces. Then, there is the need for consistently identifying the subspaces to which a linear combination coefficient set belongs in. For this purpose, the cluster-based constraints (see Chapter 6) can be used, since each cluster effectively models a subspace. The between subspace jumps can then be viewed as transitions between different clusters (see Figure 7.6). Consequently, to model the transitions between different clusters (or linear combination coefficient subspaces), a transition matrix



(a)



(b)

Figure 7.5: An illustration of a contour undergoing sudden changes due to: a) the overlapping of body parts on the image plane, b) nature of the acquisition process.

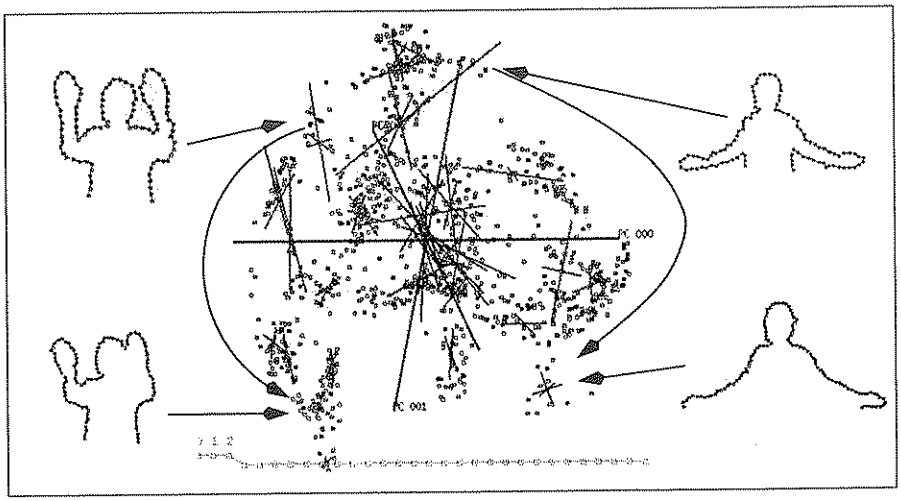


Figure 7.6: An illustration of the discontinuous nature of the hybrid representation.

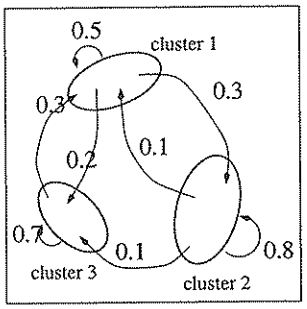
modelling the probability of a transition from one cluster to another can be used [81].

It is worth pointing out that, the use of a transition matrix is also equivalent to modelling the dynamics of the coefficient space with a Markov model. The states of the Markov model are the piecewise clusters from the cluster-based constraints. To account for potential large discontinuous behaviours in the dynamics, each state is fully connected to all the other states.

Next, a formal definition for the transition matrix will be given. Following this, the method for learning the transition matrix from available training data will be developed.

7.2.1 Definition: Transition Matrix

Formally, a transition matrix (**U**) can be defined as a two dimensional square array of transition probabilities. The number of rows and vectors of the transition matrix is equal to the number of clusters in the linear combinations coefficient constraint model. Each element in the transition matrix can be identified as $U_{i,j}$, where i is the row index while j is the column index of the element (see Figure 7.7). The i^{th} row vector of the transition matrix (**U**) consists of the transitional



Transition Matrix

	1	2	3
1	0.5	0.3	0.2
2	0.1	0.8	0.1
3	0.3	0.0	0.7

Figure 7.7: The transition matrix is illustrated in this diagram. The left image illustrates the clusters and the transition probabilities to other clusters.

probabilities of the i^{th} cluster. That is, the transition matrix element with the row index i and column index j would be the transition probability of jumping from cluster i to cluster j at the next time step.

7.2.2 Learning the Transition Matrix

The transition matrix can be constructed with the following algorithm:

1) Initialise all the elements of the transition matrix to 0.

2) For all the training sequences,

3) For every frame of a training sequence, the linear combination coefficients of its hybrid vector is recovered. Next, the memberships of the current and next frames' hybrid vector's cluster are found by determining the closest cluster.

4) If the current frame belongs to i^{th} cluster and the next frame to the j^{th} cluster, the element of the i^{th} row and j^{th} column of the transition matrix is incremented, $U_{i,j} = U_{i,j} + 1$.

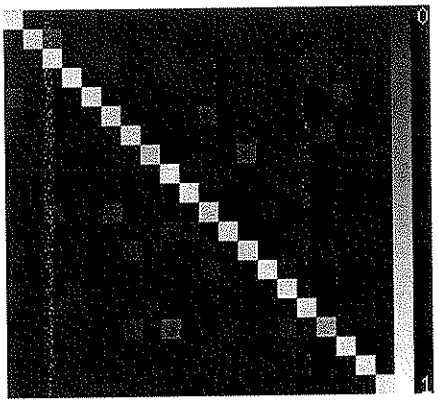
5) Finally, the values in each row vector of the transition matrix are normalised with respect to only that row.

7.3 Transition Matrices for Human Body Dynamics

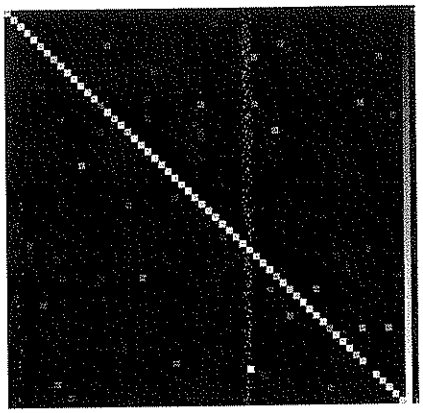
In attempting to carry out the experiments on learning the transition matrix using the algorithm given in Section 7.2.2, the prototypical examples (Chapter 5) and its cluster-based coefficient constraints (Chapter 6) were used. Each example has the form of a hybrid vector given in Chapter 3, and therefore a dimensionality of 240. In total, 69 prototypical examples were found sufficient to model the variations in the available training hybrid vectors. This resulted in a 69 dimensional linear combinations coefficient space.

The cluster based constraint models from Chapter 6 were used to model the different valid coefficient subspaces. In total, 10 different cluster models, consisting of 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 clusters respectively were used. For each cluster model of N_C number clusters, a N_C by N_C transition matrix was built. To build the transition matrix probability values, a total of 20 hybrid vector sequences were used. Each sequence represented a continuous human body motion pattern. The total contents of all the sequences were also used to learn the prototypical examples and the cluster-based constraints.

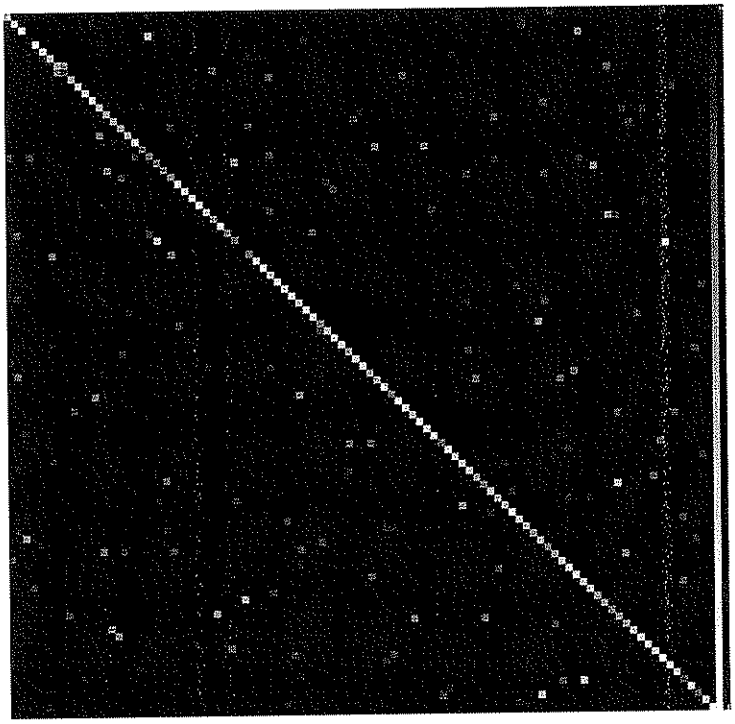
An illustration of the transition probability values captured by the matrix using different cluster models can be seen in Figure 7.8. It can be seen that for the transition matrix built from a small number of clusters, the diagonal elements tend to contain high probabilities. This implies that many generally hybrid vectors in a sequence will stay within this cluster due to its size. The transition to other clusters only happen occasionally. However, as the number of clusters increases, the transitions to other cluster become more apparent. This is because each cluster encompasses a smaller and more specific range of body configurations.



(a) 20 clusters transition matrix



(b) 30 clusters transition matrix



(c)100 clusters transition matrix

Figure 7.8: Three transition matrices for the cluster models with 10, 60 and 100 clusters respectively. The bar on the right side of the picture shows the intensity scale. The colour white indicates a transition probability of one while a colour of black indicates the transition probability of 0.

7.4 Conclusions

In this chapter, we have seen how linear combination coefficients for the hybrid vectors can have discontinuous dynamics. Analysis of the variations in the 3-D skeleton and coefficients allowed us to identify instances of hybrid vectors that exhibit discontinuous behaviour. By examining the components of these hybrid vectors, it was found that the cause of discontinuities result largely from the contour components. It was discovered that the discontinuities were due to the lack of correspondences between the contour vertices. Thus, the contour vertex positions were dependent on the length of the contour. As a consequence, sudden modification to the contour lengths would result in rapid shifts in all the contour's vertices.

Two factors that result in the sudden change of the contour length were identified. The first factor originates from the contour being a representation of the human body silhouette or shape outline. The second factor arises from the nature of the contour's acquisition process. It was shown that both these factors could significantly alter the contour of certain body configurations.

It was also shown that, the discontinuities in the linear combination coefficients could be treated as transitions between subspaces in the coefficient space. Furthermore, such a transition would allow us to exploit the coefficients' cluster-based constraints described in Chapter 6. This is because each cluster effectively encapsulates a coefficient subspace. The transitions between subspaces can be treated as transitions between clusters. It was shown that a transition matrix could be used to model the transition probabilities between the clusters. In the next chapter, we show how the transition matrix can be used in conjunction with the prototypical examples and its constraints to aid in predicting the human body configurations, despite the presence of discontinuous dynamics.

Chapter 8

Visual Tracking of Human Motions

In previous chapters, consideration was given to the task of learning a model that captures a broad range of human body configurations. A hybrid representation which combined the visual appearance and structural information was employed to represent body configuration information, as described in Chapter 3. Variations in the hybrid representation's components were captured using a set of prototypical hybrid vector examples or *prototypes*. Subsequently, hybrid vectors representing novel body poses (i.e., not prototypes) can be generated by linearly combining the prototypes. In order to increase the linear combinations model's accuracy, cluster based constraints were introduced to restrict the possible linear combinations, generating examples that represented valid body poses (Chapter 6. The dynamics of the linear combination coefficients were captured using transition matrices, as was described in Chapter 7. In this chapter, we will be concerned with the issue of *how* to generate novel hybrid vectors based on images for visually tracking a human body.

First, in Section 8.1, the problem of visually reconstructing the human body pose information is computationally defined as an optimisation task. Our approach to solving this problem uses a stochastic tracking framework called CONDENSATION, and is given in Section 8.2. Experimental results obtained using

the algorithm for visually tracking a subject from an image sequence are described in Section 8.4, before conclusions are drawn in Section 8.5.

8.1 Visually Reconstructing the Human Body Configuration

In representing the human body as an N-dimensional hybrid vector, one needs to estimate the coefficients $(\{a_1, a_2, ..., a_E\})$ for the linear combination of the known examples $(\{\mathbf{e}_1, ..., \mathbf{e}_E\})$ such that a body pose $(\mathbf{n})$ can be reconstructed accurately. In practice though, one usually does not know the *actual* values of the subject's body pose $(\mathbf{n})$.

However, an approximation $(\hat{\mathbf{n}})$ acquired by use of visual observations (e.g. by deforming an initial model to some visual observations representing the object) may be available. This approximation $(\hat{\mathbf{n}})$ may be corrupted by noise in the data and ambiguities in the deformation process. Therefore, corrupting elements from the deformed object must be removed. This entails reconstructing the closest object to $(\hat{\mathbf{n}})$ using linear combinations. Mathematically, this involves obtaining the coefficient set $(\{a_1, a_2, ..., a_E\})$, which minimises the magnitude of the "approximation residuals" vector $(res_1, ..., res_N)$,

$$res_1 = n_1 - a_1 e_{1,1} + a_2 e_{2,1} + \cdots + a_E e_{E,1} \tag{8.1}$$

$$res_2 = n_2 - a_1 e_{1,2} + a_2 e_{2,2} + \cdots + a_E e_{E,2} \tag{8.2}$$

$$\vdots \tag{8.3}$$

$$res_N = n_N - a_1 e_{1,N} + a_2 e_{2,N} + \cdots + a_E e_{E,N} \tag{8.4}$$

where $\mathbf{n} = (n_1, n_2, ..., n_N)$ and represents the object's current state. The i^{th} example vector's components are denoted by $(e_{i,1}, e_{i,2}, ..., e_{i,N})$.

8.2 Dynamic Linear Combinations: CONDENSATION

Typical methods for minimising Eq. 8.4 (i.e. obtaining the appropriate coefficients) involve a least-squares minimisation [76, 86] or some other optimisation procedures [77, 48]. Here, the CONDENSATION [52] framework was adopted for estimation of coefficients by tracking them over time. Hence, this algorithm allows spatio-temporal knowledge to be used, allowing for a more robust tracking. Spatial knowledge consists of learnt coefficient space (global eigenspace) and constraints (piecewise clusters) as described in the previous chapter. The temporal knowledge is modelled using two methods. For modelling the coefficient "structural-dynamics", a Markov model of transitional probabilities between different subspaces modelled with clusters is employed, as described in the previous chapter. For the finer scale of movements within the clusters, a Brownian motion model (random displacements) is used [52].

8.2.1 CONDENSATION algorithm

In this context, the CONDENSATION framework consists of an algorithm working on a set of samples. We define the number of samples as, N_S. Here, a sample is a coefficient set for a single linear combination. The algorithm propagates the samples in the coefficient space based on learnt dynamics of the coefficients. The samples are rated according to how well they fit the observable data that is used in the next iteration's propagation step.

A *sample* is defined as a point in the coefficient space (i.e. a coefficient set). Associated with each sample ($\mathbf{s}_n^t$) at a time instance (t), is a fitness measure (f_n) indicating its accuracy in representing the actual coefficient point. Initially, all the samples are assigned equal fitness values and their components are randomly distributed within their constraints. The algorithm then iterates over the following

steps:

1) The first step involves the selection of future samples based on their fitness. This can be done by firstly constructing a normalised cumulative histogram ($\mathbf{h}$) of the samples' fitness values. Each of the histogram's values ($h_i, i = \{1, ..., N_S\}$) can be determined as follows:

$$h_i = \frac{\sum_j^i (f_i + f_{min})}{\sum_k^{N_S} (f_k + f_{min})} \tag{8.5}$$

where f_{min} is the smallest fitness value across all the samples. A new population of samples can be selected using the following procedure [52]: At the time step $t+1$, for the n^{th} sample of the N_S samples,

   a) Generate a random number r between 0 and 1 from a uniform distribution.

   b) Find the smallest m for which $h_m \geq r$.

   c) Set the n^{th} sample at time $t+1$ as the m^{th} sample at time t, $\mathbf{s}_n^{t+1} = \mathbf{s}_m^t$.

2) The selected samples are then propagated based on a model of their dynamics. This propagation step effectively predicts the future state of the samples. Further details of the propagation step is given in 8.2.2.

3) The accuracy of the prediction step is then determined by measuring how well each sample "fits" with the observation data. The fitness values of the samples are replaced by this new fitness value. We describe this step in greater detail in Section 8.2.3.

4) The sample vector with the highest fitness value is selected and its reconstruction used. This has the effect of selecting the state vector that contains the tracked 2-D measurements most similar to those extracted from a given image. This state vector also contains the corresponding *valid* 3-D skeleton.

This is because all the points are constrained to the space covered by the clusters.

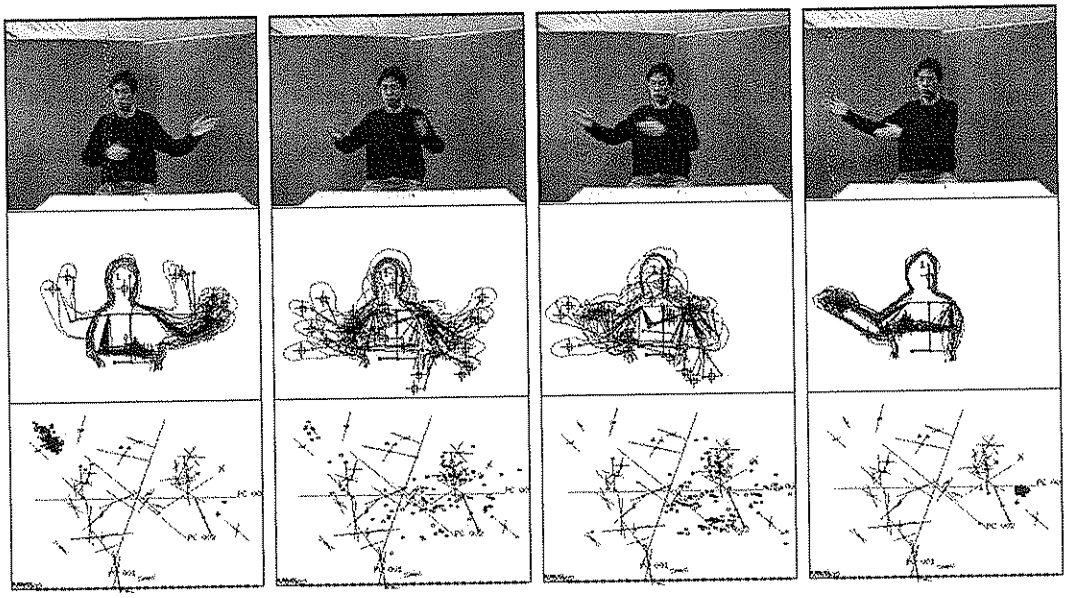


Figure 8.1: The tracking of a population of samples by CONDENSATION is illustrated here. The bottom row shows the coefficient space of the first three examples with the clusters' local principal components. The points in the image represent the propagated samples. The middle row illustrates the reconstruction of 10 out of 100 samples. The top row shows the input images.

For completeness, we will describe next the CONDENSATION propagation step originally developed for tracking hand contours [81]. Following this, it is shown how we modified this method for tracking the human body configurations. This was done by adapting the algorithm for estimating the linear combinations coefficients, the sample prediction step is modified to make use of the transitional probability matrix (U) as described in Section 7 and the set of N_C number of coefficients cluster constraints ($\mathbf{C} = \{\mathbf{c}_1, ..., \mathbf{c}_{N_C}\}$) as described in Chapter 6. This allows the tracker to propagate samples across different subspaces, thus allowing it to cope with any discontinuities in coefficient space. An illustration of the CONDENSATION tracker can be seen in Fig.8.1.

8.2.2 Propagating the samples

The sample prediction stage is split into two steps:

1) The first step involves finding out the new cluster membership, labelled as b, for a sample at time t, $\mathbf{s}_n^t$, based on the transitional probabilities given by the a^{th} row vector of the transition matrix, where a is the sample's current cluster membership.

2) The new position of the sample, $\mathbf{s}_n^{(t+1)}$, is determined by displacing it linearly in the directions of the principal components of the clusters:

$$\mathbf{s}_n^{(t+1)} = \begin{cases} \mathbf{s}_n^{(t)} + \mathbf{P}_b \mathbf{\Lambda}_b \mathbf{\Omega}, & a = b \\ \boldsymbol{\mu}_b + \mathbf{P}_b \mathbf{\Lambda}_b \mathbf{\Omega}, & a \neq b \end{cases} \tag{8.6}$$

where as described in Section 6.3, $\boldsymbol{\mu}_b$, $\mathbf{P}_b$ and $\mathbf{\Lambda}_b$ are the mean, the matrix of the principal axes, and the eigenvalues of the b^{th} cluster ($\mathbf{c}_b$) respectively. $\mathbf{\Omega}$ consists of the vector whose elements are unit Gaussian distributed random values [81].

The position of the new sample, $\mathbf{s}_n^{(t+1)}$, is then constrained to lie within the bounds of its newly assigned cluster $\mathbf{c}_b$. This is achieved by projecting the sample as follows:

$$\mathbf{r} = (\mathbf{s}_n^{(t+1)} - \boldsymbol{\mu}_b)\mathrm{T})\mathbf{P}_b \mathbf{\Lambda}_b^{-1} \tag{8.7}$$

The restriction is made to obtain a plausible reconstruction of this sample. All the elements of $\mathbf{r}$ are then limited to lie within the range of -1 to $+1$. Any element outside the bounds of this range is set to 1 or -1 respectively.

The sample's elements are then reconstructed by taking a linear combination of the principal components of cluster $\mathbf{c}_b$:

$$\mathbf{s}_n^{(t+1)} = \mathbf{\Lambda} \mathbf{P}_b^{\mathrm{T}} \mathbf{r} + \boldsymbol{\mu}_b \tag{8.8}$$

The estimated state vector ($\mathbf{v}$) is reconstructed from the components of a given sample in the same manner, by taking a linear combination of N_E number of prototypes ($\mathbf{e}_1, ..., \mathbf{e}_{N_E}$):

$$\mathbf{v} = \sum_{i=1}^{N_E} s_{n,I} \mathbf{e}_i \tag{8.9}$$

8.2.3 Measuring the Samples' Fitness

In the original work by Heap, the reconstructed vector ($\mathbf{v}$) consisted of only contour vertices. In order to adapt this method to suit our needs, we replaced the reconstructed vector ($\mathbf{v}$) with the hybrid vector, $\mathbf{v} = (\mathbf{v}_S, \mathbf{v}_C, \mathbf{v}_T)$. The accuracy of both contour ($\mathbf{v}_S$) and the body parts ($\mathbf{v}_C$) are measured individually and then combined to yield the final fitness value. In this section, we first describe how the samples' fitness is measured. In the next section, details of the process for obtaining the observation information used for the measurement process will be described.

A prediction accuracy value (f_S) for the contour can then be computed as follows:

1) Assign the prediction accuracy value, $f_S = 0$.

2) For all the N_C number of vertices of contour: (a) Find the distance, s, from the vertex position to the pixel of greatest intensity gradient by searching along its normal, (b) add this to f_S; $f_S = f_S + s$.

We now deal with measuring the accuracy of the state vector's prediction of the body parts positions, (x_{p1}, y_{p1}) and (x_{p2}, y_{p2}), is. For each frame, three skin coloured regions of interests were tracked using colour models composed of Gaussian mixtures [75]. These positions correspond to the positions of both hands and the face.

A combinatorial operation is then performed to determine which two out of the three positions are the hand positions. To achieve this, two out of these three

positions are chosen. Suppose we denote the index of the two positions chosen as, $m1$ and $m2$, and that they can each take the values of 1, 2 or 3. Additionally, we also require that, $m1 \neq m2$. The two positions can then be ordered into a 4 dimensional vector $\mathbf{m}_b = (x_{m1}, y_{m1}, x_{m2}, y_{m2})$ where (x_{m1}, y_{m1}) and (x_{m2}, y_{m2}) are the co-ordinates of the first and second position respectively. We now define the accuracy of the prediction of the body parts' positions as f_C whose value is determined as follows:

$$f_C = \min_{m1,m2}\{\sqrt{(x_{p1} - x_{m1})^2 + (y_{p1} - y_{m1})^2} + \sqrt{(x_{p2} - x_{m2})^2 + (y_{p2} - y_{m2})^2}\} \quad (8.10)$$

The final fitness value, (f_n), for $\mathbf{v}$ of the n^{th} sample, $(\mathbf{s}_n^{(t+1)})$, combines both the individual fitness measurements in the following manner:

$$f_n = -Of_C - Rf_S \quad (8.11)$$

where O and R are the constants used to even out differences in scale between the two weighted fitness measurements of f_C and f_S. Therefore, a less negative value of f_n would represent a higher fitness. Furthermore, since the values of f_C and f_S are sums of a number of Euclidean distances, their scales can be normalised by dividing by the number of contour points (N_C) and hands (2) respectively:

$$O = \frac{1}{N_C} \quad (8.12)$$

$$R = \frac{1}{2} \quad (8.13)$$

8.3 Acquiring the Measurement Data

8.3.1 Acquiring the Contour Observations

To increase the accuracy of the contour measurement process, a binary image containing mainly the silhouette of the subject was first extracted. This was achieved in a similar fashion to the training data acquisition process described in Section 3.4. In particular, background subtraction was used to recover an image

containing mainly foreground pixels. The colour of the background pixels was set to black. In order to clean up spurious foreground pixels (e.g. due to image noise), a single step of the morphological dilation operation was performed on the foreground image.

However it has to be noted that the use of background subtraction only works in limited environments. In particular, in environments where the lighting conditions and background objects can be fairly well controlled. Nevertheless, there have been recent developments on more robust background and foreground detection methods that can be used in a more general setting [30, 37, 65].

8.3.2 Acquiring the Body Parts Positions Observations

In order to obtain the observed body parts positions, a skin colour model using a mixture of Gaussians [75] was initially built prior to the tracking process. Recent advancements in colour modelling for tracking has also allowed for one to dynamically build the colour models whilst tracking [65].

This model was then used to determine which of the foreground pixels are skin coloured. Subsequently, k-means clustering was performed on the positions of the skin coloured pixels. Here, the number of clusters was set as 3, two for the hands' positions and one for the position of the head.

However, there exist problems with adopting such a method for detecting the positions of any body parts positions. Perhaps the most prominent amongst these issues lies in the existence other parts of the body that have a similar colour. For example, should a subject decide to wear clothing that are skin coloured, the resulting pixels corresponding to the skin coloured clothing would be incorporated into the k-means process for detecting the positions of the hands and head. This in turn would yield an inaccurate estimation of the true image positions of the required body parts positions. Moreover, the use of k-means clustering makes the acquisition process susceptible to self occlusions, since it always assumes a fixed

number of objects. However, there does exist more robust alternative methods for tracking certain body parts positions, for example as shown by Sherrah and Gong [42]. There, Bayesian networks are adopted to infer the most probable position of the hands based on colour and motion image features.

8.4 Experiments and Analysis

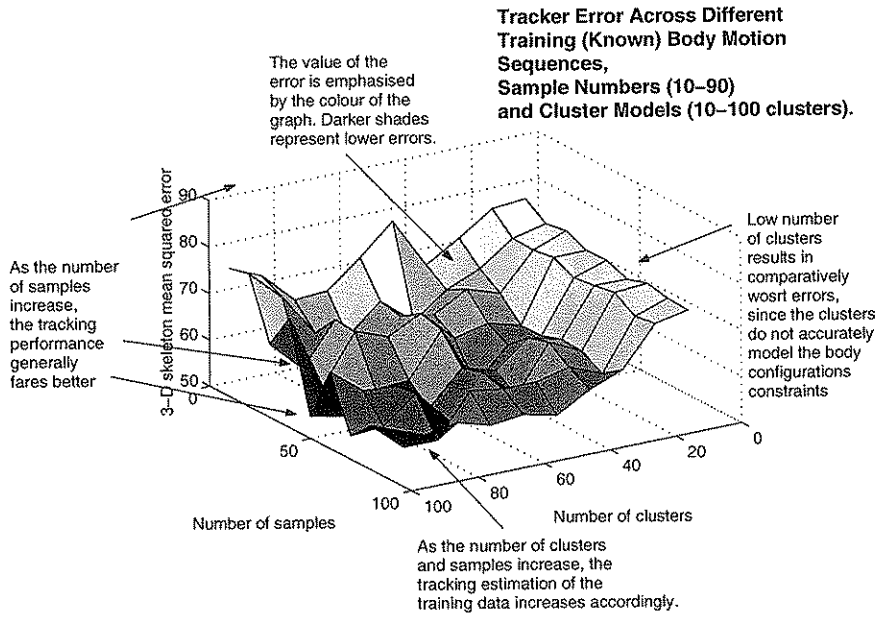
A tracker utilising CONDENSATION was implemented. The tracking processing time was roughly between 1 to 3 seconds for each frame.

A series of different experiments on tracking the 3-D skeletons using a single view was performed. Due to the lack of sufficient training examples, the transitional probability matrix built was not an accurate representation of its real values. This has slowed down the transition of the samples across different clusters. It was found that iterating the CONDENSATION process for a number of times over the same frame (5 iterations were sufficient for the experiments carried out here) allowed the samples to converge on the correct subspace.

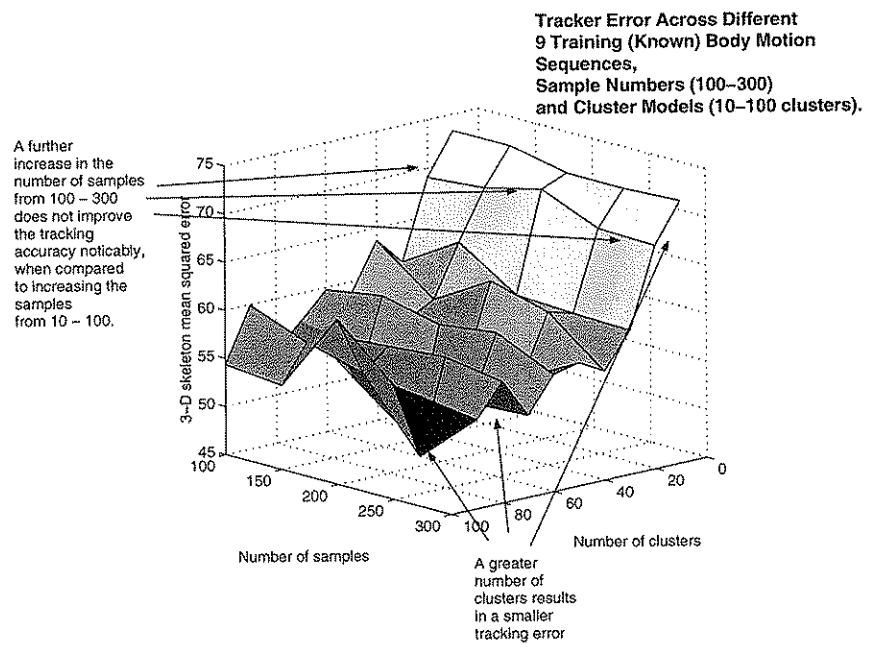
8.4.1 Tracking Known Motion Sequences

Initially, to determine how accurately the training poses were learnt when the example based kinematics model (i.e. prototypical examples and cluster constraints) was used, the tracker was made to track the 3-D skeleton of the subject in the training sequences. In all, a total of 9 continuous training motion sequences whose data were used for building the cluster models, examples and transition matrices were used. Each of the 9 sequences contains a different motion sequence. This resulted in a total of 502 frames of different body configurations.

Using the 9 training sequences, a set of experiments whereby the tracker was set with different cluster models, transition matrices and number of samples were performed. In particular, different cluster models with 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 clusters were used, along with their transition matrices. For



(a)



(b)

Figure 8.2: The error surfaces of the tracking experiments carried out on training data when different number of samples and cluster models are used.

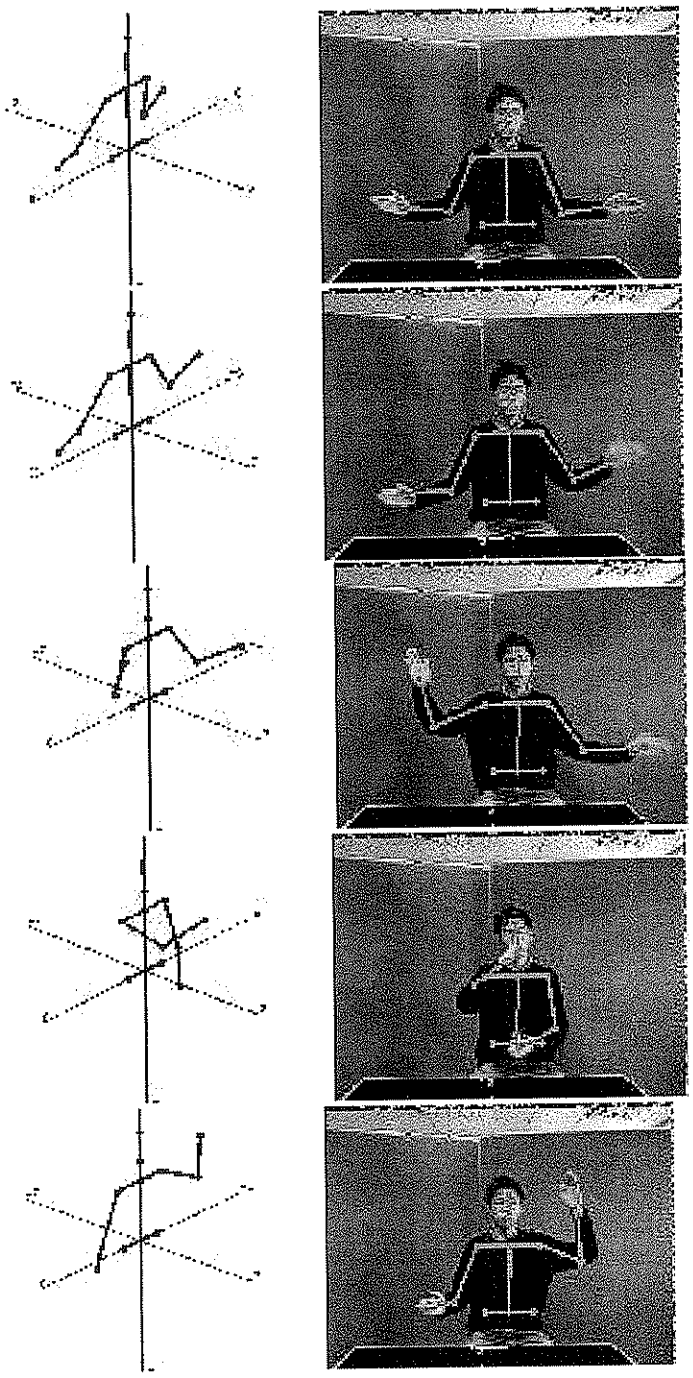


Figure 8.3: This figure shows the tracking of 3-D skeletons using the CONDENSATION algorithm on the training sequences. Every 10th frame is shown. For each frame, the left part shows the 3-D skeleton while the right shows the input image. Additionally, the 3-D skeleton projected on the image plane is overlaid on the input image.

each of the cluster models, an attempt to track the training sequences' body configurations (represented by 3-D skeletons) with 10 to 90 samples (increments of 10 samples) was initially made. The results of the different combinations of cluster models and number of samples can be seen in Figure 8.2a. In order to investigate the effects of further increasing the number of samples, an additional set of experiments were carried out, where 100, 150, 200, 250 and 300 samples were used. The resulting tracking error surface can be seen in Figure 8.2b. A visual illustration of an example of the tracking results can be seen in Figure 8.3.

It was found that as the number of clusters and samples increased, there was also a general trend for improvement in the tracking accuracy. The increasing number of cluster allowed more accurate and realistic body configuration information to be produced. However, as only training sequences are used for this experiment. Therefore, it is not clear as to how well the tracker can generalise to novel body motions using the different parameters and cluster models.

8.4.2 Tracking Novel Motion Sequences

The subsequent experiments were aimed at evaluating the tracker's ability to generalise and track novel body motions using different cluster models and sample numbers. The test motion sequences used are novel in that they contain different segments of body motions of different training motion sequences.

The first experiment studies the case of tracking in fairly controlled conditions. A blue screen was placed behind the subject to provide a homogeneous background. A visual illustration of some tracking results is shown in Figure 8.4. Having a homogeneous background allowed for a fairly accurate segmentation of the subject in the input image. This allowed the tracker to more accurately compare the contour components of its samples to the edges of the segmented image.

The second experiment investigated a subject being tracked in the presence of

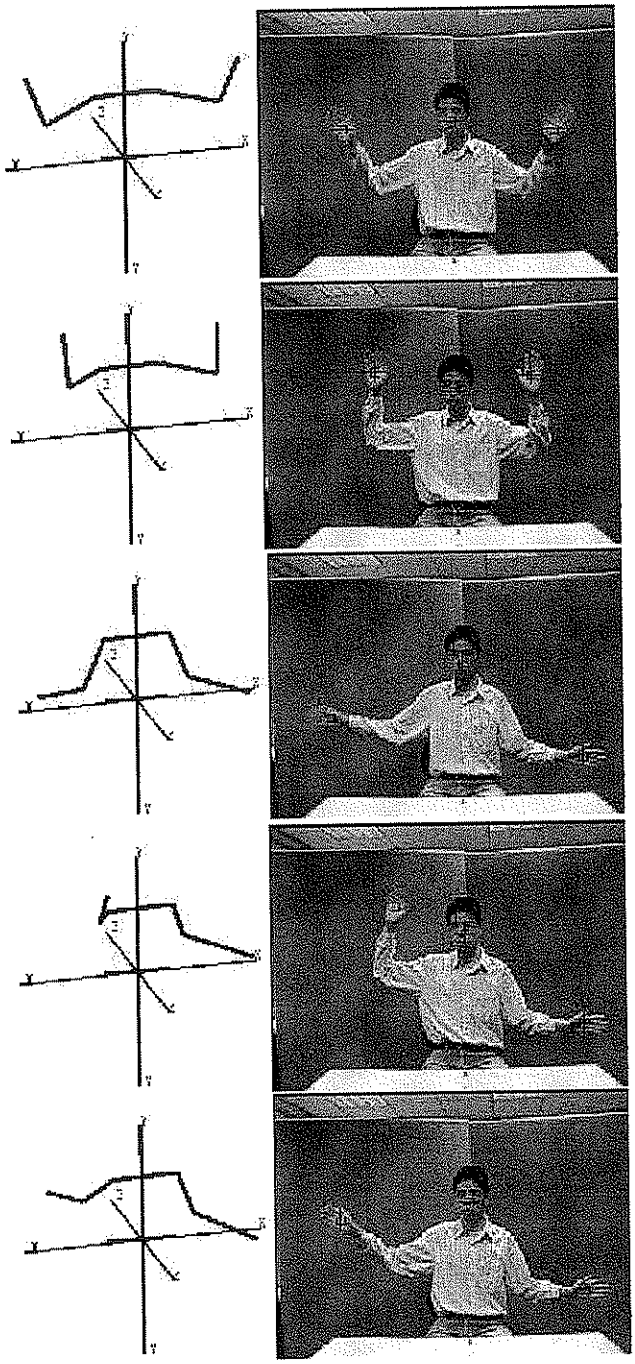


Figure 8.4: Single view tracking using the CONDENSATION algorithm. This shows the tracking of a novel gesture in a controlled environment. The 10th, 17th, 30th, 40th and 50th frame is shown from top to bottom respectively. Again, the left part shows the tracked 3-D skeleton while the right shows the input image.

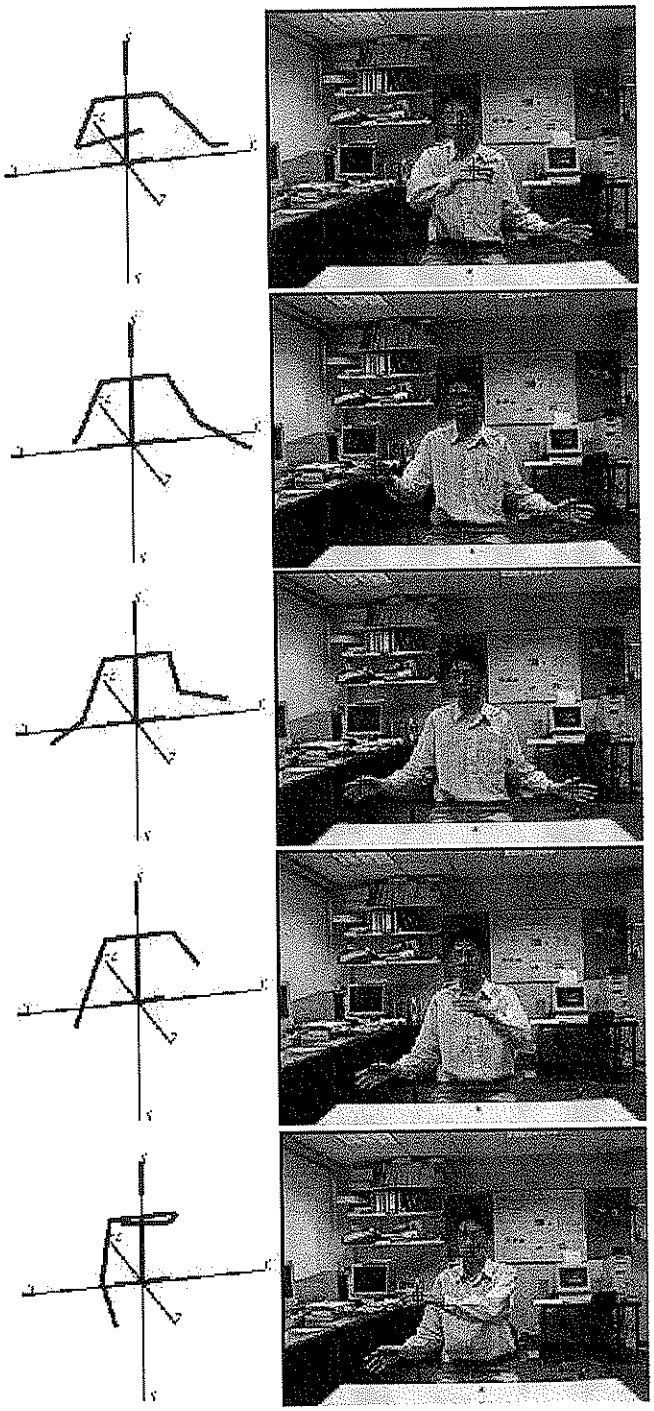


Figure 8.5: This figure illustrates the tracker working in the presence of a cluttered background. The 3rd, 10th, 14th, 25th and 30th frame of the continuous sequence is shown, from top to bottom respectively. Again for each frame, the 3-D skeleton is on the left while the input image is shown on the right.

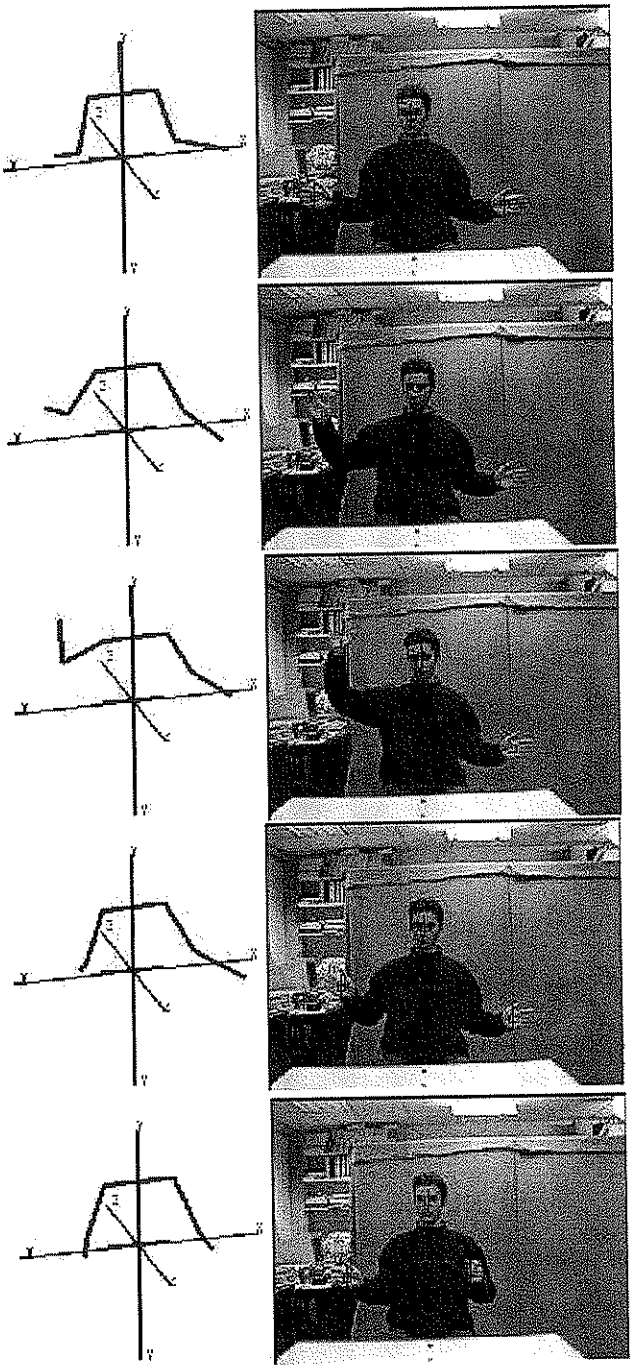
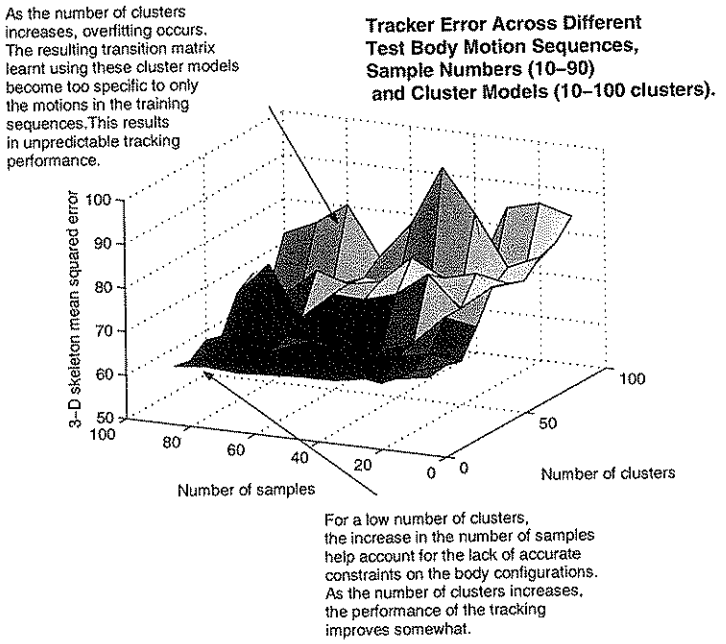
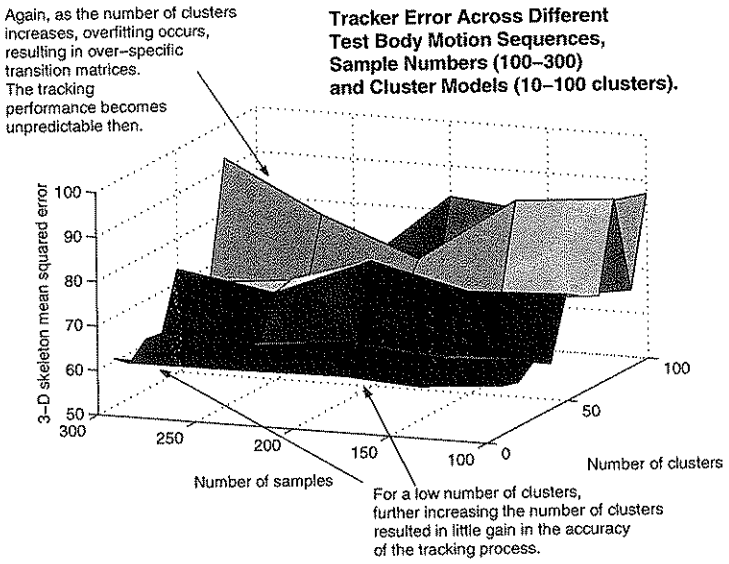


Figure 8.6: This figure illustrates the tracker working on a novel subject that is not present in any of the training sequences. The 3rd, 16th, 29th, 50th and 61st frame of the continuous sequence is shown from top to bottom respectively. Similar to the previous figures showing the tracking results, the tracked 3-D skeleton is shown on the left while the input image is shown on the right.



(a)



(b)

Figure 8.7: The error surfaces of the tracking experiments carried out on test sequences of novel body movements with a different number of samples and cluster models with different number of clusters.

a cluttered background. Again, the subject performed motions that were combinations of different training movement segments. In such situations, it was found that the positions of the hands were important in disambiguating the poses in the presence of contours matched inaccurately to spurious edges. The results can be seen in Figure 8.5.

Finally, to evaluate the tracker's performance in generalising to novel subjects, the third experiment was performed with a subject that was not present in any training sequences. The results can be seen in Figure 8.6. The overall error surfaces for the three experiments are shown in Figure 8.7. From the results, it was observed that the tracking performance deteriorates and becomes more unpredictable as the number of clusters increase. Such results bear the implication that cluster models with too large a number of clusters have overfitted the data. The resulting transition matrix for the cluster models would account only for the training motion patterns. A novel sequence may contain transitions between clusters that were not modelled using the training data.

8.4.3 Recovering from Tracking Failure

Finally, it was observed in some experiments the ability for the tracker to recover from failure in tracking. Figure 8.8 shows an example of a sequence where the initial body pose was wrongly initialised. However, the subsequent frames showed the tracker having recovered and the estimation of the body configuration close to the known configuration.

Additionally, there are instances where the tracker does fail in estimating the body configuration in the middle of a motion sequence, as shown in Figure 8.9. However, again, the tracker was shown to have recovered from such a failure.

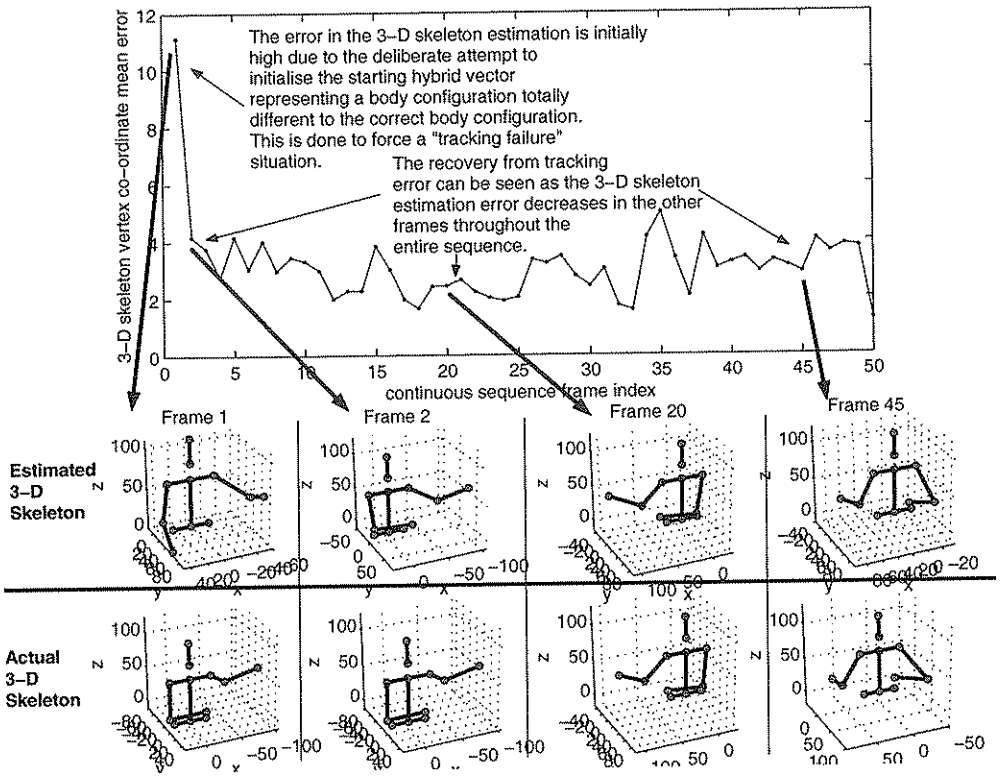


Figure 8.8: An example where the tracker recovered from failure in tracking the subject's body configuration. The graph shows a bad initial estimate of the body configuration. However, the error measurements for the subsequent frames show the tracker recovering from this initial error.

8.5 Conclusions

In this chapter, the stochastic algorithm of CONDENSATION was chosen to visually track the human motions (i.e., continuously changing body configurations). In the context of the linear combinations method, it was shown that the CONDENSATION algorithm is equivalent to a method for estimating the coefficients to reproduce a novel hybrid vector that represents the configuration of the object in the image. Alternatively, the CONDENSATION algorithm has also provided means of inferring the body configuration represented by 3-D skeletons from images.

It was shown that the CONDENSATION algorithm provided a generative

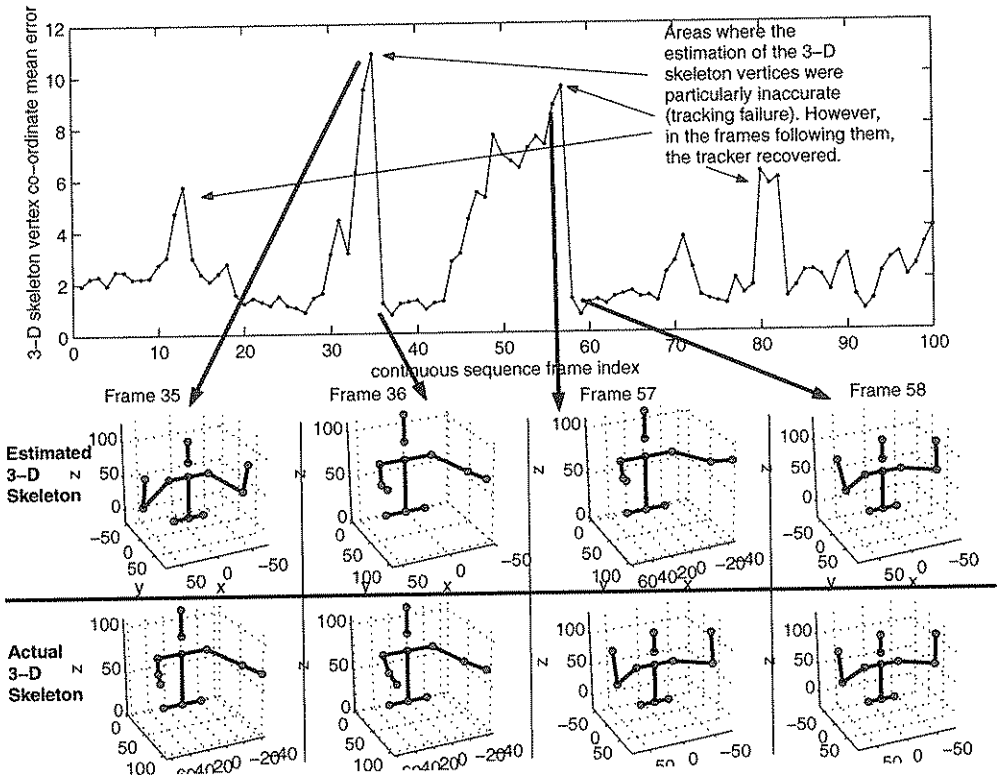


Figure 8.9: Another example of the tracker recovering from failure during the tracking process. The graph shows a number of occasions where the tracker failed to track the body configuration. However, the tracker managed to regain track of the body configuration after a period of time, as indicated by the error graph.

mechanism combining both the example-based kinematics model and the dynamics model for the purpose of tracking the human body configurations. The kinematics model was used to generate valid information on the human body configuration. The dynamics model in turn allowed one to predict the future configuration of the human body. This was achieved in the presence of discontinuous and non-linear dynamics in the human body's hybrid vector representation.

Tracking using the CONDENSATION algorithm involved the propagation of a set of human body configuration "samples". Each sample is represented by a set of linear combinations coefficients. In order to determine the accuracy of a sample, the original hybrid vector was regenerated using linear combinations. The contour

and hands positions components of the hybrid vector was then compared to the input image. It was also shown that the CONDENSATION method provided a robust tracking mechanism that can recover from failure in estimating the correct body configuration.

Chapter 9

Conclusion

The previous chapters have described the issues involved in attempting to visually track the motions of the human body. This chapter summarises the discoveries and work covered in the previous chapters and suggests areas of interests for future work.

Hybrid Vector Representation

Firstly, Chapter 3 considered the usage of a hybrid representation that combined a human body's visual appearance and structural information to represent the different configurations of body parts. Specifically, two types of information or *modalities* defined the visual appearance of the human body. These two modalities were the hand positions and contour surrounding the silhouette. The structural information was defined as a skeleton consisting of a set of 3-D vertices. The utilisation of both 2-D and 3-D based modalities allowed us to account for the 3-D nature of the human body while exploiting available visual information from input images. The latter element allowed avoidance of the complex task of synthesising the visual appearance of a human body using computer graphics methods.

However, it is inevitable that there would be ambiguities in the visual appearance modalities since it is 2-D based while the human body is a 3-D entity. Therefore, issues pertaining to using a hybrid vector for representing a human body configuration in the presence of these ambiguous 2-D information were ad-

dressed in Chapter 4. This led to the discovery that contours are a less ambiguous 2-D based modality than hand positions. However, hand positions were found to have an important use, owing to its computational reliability when recovered from input images.

Example-Based Kinematics Framework

Chapter 5 described an example-based kinematics framework that was employed as a generative mechanism for the hybrid vectors. More specifically, the linear combinations of examples framework was adopted. A small number of prototypical hybrid vectors were combined using different weights or *coefficients* to generate new hybrid vectors that represented novel body configurations (i.e., not the prototypes). It was found that when a set of training hybrid vectors was available, Principal Component Analysis (PCA) could be used to recover of the contents of the prototypes. Furthermore, a Bayesian method was employed for determining the probability that the training set could be modelled when given a number of prototypical examples. The number of required prototypes was determined by selecting the number that gave the highest probability. It was also found that some prototypes account for a more significant or larger range of human body configurations than other examples.

Learnt Cluster-Based Kinematics Constraints

It was found that certain linear combinations could yield hybrid vectors that represented implausible human body configurations. Thus, in Chapter 6 we addressed the need for constraining the linear combinations coefficients to generate only plausible body configurations. This was achieved by firstly treating the set of coefficients for generating a single hybrid vector as a high dimensional *coefficient vector*. A coefficient vector that generated a valid body configuration hybrid vector was defined as a *valid* coefficient vector. Also, the space of the coefficient

vector was defined as the *coefficient space*. The valid coefficient vectors for all valid body configurations must span a region within the coefficient space. Modelling the constraints for the plausible linear combinations can be achieved by modelling only this valid coefficient region.

To this end, a study of the valid coefficient vectors' characteristics was carried out by visualising the coefficients of different prototypes. It was found that the coefficient vectors occupied a highly non-linear region in the coefficient space. This made clear the necessity for a model that can cope with non-linear regions. For this, a model consisting of a set of clusters was chosen. Each cluster was allowed to occupy any position and encompass a hyper-cubic region. The clusters' parameters (i.e., position and region shape) were then estimated using the Expectation Maximisation (EM) algorithm.

Since each cluster occupied a limited region of the coefficient space, a cluster accounted for a limited range of valid body configurations. Provided the number of clusters was sufficient, the entire set of clusters could account for the different known valid body configurations. It was found that as the overall number of clusters was increased, each cluster accounted for an increasingly specific range of body configurations. Also, the ability for one to use the cluster constraints to reconstruct missing information in a hybrid vector was improved.

Learnt Human Body Dynamics using Transition Matrices

In Chapter 7, the dynamical characteristics of the human body were investigated by considering how the coefficient vector evolves under different human body motions. It was found that certain human body motions caused the coefficient vectors to exhibit discontinuous dynamics. The discontinuities were found to originate from the lack of correspondence between the vertices of different contours. This caused the vertices to "slide around" in the case of a sudden change to the contour length. As a result, there would be a large change in the hybrid vec-

tor's contour components. Consequently, the linear combination coefficients for generating the hybrid vector would undergo a large change. To deal with the discontinuities, an occurrence of a discontinuity was treated as a transition between different subspaces. The cluster model could then be exploited as a model of coefficient subspaces. Transitions between subspaces could then be treated as transitions between clusters. Finally, to model the cluster transitions, a transition matrix was employed.

Visual Tracking of Body Configurations

Finally, Chapter 8 described a stochastic framework which could be used to visually track different patterns of body movements. The framework proposed follows the CONDENSATION framework originally developed by Isard and Blake [52]. In this framework, a coefficient vector or *sample* was used to represent the system's hypotheses for the actual body configuration in an input image. To determine the accuracy of a sample at representing the actual body pose, a hybrid vector was generated from the sample using the linear combinations method. Subsequently, a fitness measurement was obtained by comparing the visual appearance components of the hybrid vector with the image information. Specifically, the hand positions were compared against tracked skin coloured regions and the contour against the image's edge information. To ensure that the sample represented valid body configurations, the coefficient space cluster model was used. All samples were then restricted to fall within the cluster space.

To account for human body motion, a sample must be displaced or *propagated* accordingly. This can be thought of as an attempt at predicting the human body's next configuration. To this end, random noise was used to propagate the sample within a cluster subspace. Furthermore, the transition matrix was used to guide the propagation of the sample across the different cluster subspaces. The use of the transition matrix has the advantage of being able to cope with the discontinuous

nature of the coefficient vectors.

Finally, in order to cope with unexpected movements in the body motions or to aid recovery from tracking failure, a set of samples were used. Each sample was associated with its fitness value. Initially the samples were assigned equal fitness and distributed randomly in the coefficient cluster space. The process of visually tracking a human body in motion then consisted of repeating the following steps: 1) A new set of samples were selected according to their fitness values such that, samples with low fitness values are more likely to be discarded and vice versa. 2) Each sample was then propagated as described above using the transition matrix or random noise. 3) Next, the fitness values of the samples were recovered. 4) The sample with the best fitness value was then chosen as the approximation of the current body configuration. In the experiments carried out, it was found that an increase in the number of clusters and samples decreased the body configuration estimation's error.

9.1 Future Work

9.1.1 Training Data Acquisition Process

Since the parameters of the models for the human body configurations were all acquired through learning processes, it is inevitable that their accuracy is dependent on the training data that was available. Should the training data be insufficient or inaccurate, the resulting models would have accordingly contained errors or inaccuracies. For example, should the training set have omitted a range of body configurations, the resulting cluster models too would not cover them. Therefore, a set of body configurations would have wrongly been labelled as invalid body poses.

Currently, the existing hybrid vector acquisition system requires a subject performing a gesture to sit in front of a camera set in a blue screen background.

A video sequence of the gesture is first recorded. The contour and hand positions are then extracted using background subtraction and colour tracking respectively. The 3-D skeleton vertices are then hand labelled by manually locating various joint positions on the body. Such a method lends to inaccuracies due to image noise for the 2-D information and human error for the 3-D skeleton vertices. Additionally, delays in the video recording process may cause certain body configurations to be discarded unnecessarily.

A possible solution for overcoming such problems would be to synthesise the training data. Advances in computer graphics techniques by now make this a valid option. In the learning stage where time and computing resources are more abundant, the components of the hybrid vector could be synthesised using computer graphics. There, the 3-D skeletal information could be made available from the 3-D virtual model of a human body. Subsequently, the contour and positions of the hands can be obtained from the projection of the 3-D model onto the virtual camera plane. There would be no image noise affecting the contour or body parts positions' accuracy. Additionally, the rendering system could be configured to automatically synthesise as many different possible body configurations as possible.

9.1.2 Kinematics Constraints Learning Methods

The EM algorithm used for determining the parameters of the coefficient space cluster-based constraints suffers two shortcomings. Firstly, there are no mechanisms in the algorithm that allow the estimation of the required number of clusters. Secondly, since the parameters are updated locally, the algorithm suffers from convergence into local minima. Both of these factors can transpire to a sub-optimal cluster model for capturing the valid coefficient vectors. In other words, the constraints of the valid body configurations were modelled inaccurately. Consequently, this results in the possible generation of information for implausible

body configurations during the visual tracking process. An indication of this was found when increasing the number of clusters was seen to improve the tracking accuracy.

A potential solution for overcoming these shortcomings may be the method of Entropic Minimisation (Brand [50, 49]). There, the entropy of the cluster model is minimised such that the resulting model has the simplest structure with the best fit to available training data. Another possible solution could lie in the method by Bishop and Winn [11], where variational inference methods are exploited for determining the parameters of a cluster model, including the number of clusters. Their framework comprises of a mixture of sub-space components where both the number of components and their subspace dimensionality are determined automatically as part of the Bayesian inference procedure. Therefore, this framework has the advantage of not requiring the user to set any significant adjustable parameters. Additionally, the number of clusters is inferred from the data, without the need for performing model optimisation using computationally expensive cross validation methods.

9.1.3 Integration with Inverse Kinematics

While performing visual tracking, multiple hypotheses of different body configurations or samples were used. It was found that an increase in the number of samples improved the accuracy of the tracking process. However, certain samples were wasted, in that they contained visual information that was inconsistent with that extracted from the image. For example, the visually tracked hand positions may be totally different to the hand positions of the samples.

One can potentially improve this situation by integrating the inverse kinematics process into the propagation process of the samples. Using the hand positions extracted from the image, inverse kinematics could be used to infer possible 3-D skeleton configurations that have their hand positions. The clusters could then

be used to "complete" the contour information such that it is consistent with the 3-D skeleton.

Bibliography

[1] A.Azarbayejani and A.Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. Technical Report 363, Perceptual Computing Section, MIT Media Lab, Cambridge, MA, USA, 1996.

[2] A.Azarbayejani, C. Wren, and A. Pentland. Real-Time 3-D Tracking of the Human Body. In *Proc. of IMAGE'COM 96*, Bordeaux, France, May 1996.

[3] A.Baumberg and D.Hogg. An Efficient Method for Contour Tracking using Active Shape Models. Technical report, School of Computer Science, University of Leeds, April 1994.

[4] A.Baumberg and D.Hogg. Learning flexible models from image sequences. In *European Conference on Computer Vision*, May 1994.

[5] A.Bobick, S.Intille, J.Davis, F.Baird, C.Pinhanez, L.Campbell, Y.Ivanov, A.Schtte, and A.Wilson. The kidsroom: A perceptually-based interactive and immersive story environment. In *Presence: Teleoperators and Virtual Environments*, pages 367–391, 1999.

[6] A.Broggi, M.Bertozzi, A.Fascioli, and M.Sechi. Shape-based Pedestrian Detection. In *Proc. of the IEEE IV-2000, Intelligent Vehicles Symposium*, pages 215–220, October 2000.

[7] A.Dempster, N.Laird, and D.Rubin. Maximum Likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[8] A.Heap and D.Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. In *Proc. of the International Conference on Computer Vision '98*, Bombay, 1998. IEEE Computer Society Press.

[9] C.Barclay, J.Cutting, and L.Kozlowski. Temporal and Spatial factors in Gait Perception that Influence Gender Recognition. *Perception and Psychophysics*, 23(2):145–152, 1978.

[10] C.Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1996.

[11] C.Bishop and J.Winn. Non-linear Bayesian Image Modelling. In *Proc. of the ECCV*, 2000.

[12] C.Cedras and M.Shah. Motion-based recognition: A survey. *Image and Vision Computing*, 13(2):129–155, 1995.

[13] C.Wren, A.Azarbayejani, T.Darrell, and A.Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

[14] D.Decarlo and D.Metaxas. The integration of optical flow and deformable-models with applications to human face shape and motion estimation. In *Proceedings of the IEEE Computer Society on Computer Vision and Pattern Recognition*, June 1996.

[15] D.Gavrila. The Visual Analysis of Human Movements: A Survey. *Computer Vision and Image Understanding*, 73(1):82–98, January 1999.

[16] D.Gavrila and L.Davis. Towards 3-d model based tracking and recognition of human movement:a multi-view approach. In *FG'95*, Zurich, 1995.

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

[18] D.Lowe. Fitting Parameterized Three-Dimensional Models to Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[19] D.Marr and S.Ullman. Directional selectivity and its use in early visual processing. *Proceedings of the Royal society of London*, pages 151–180, 1981.

[20] D.Metaxas and D.Terzopoulos. Shape and Nonrigid Motion Estimation Through Physics-Based Synthesis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(6):580–591, June 1993.

[21] E.Ong and S.Gong. A Dynamic 3D Human Model from Multiple Views. In *British Machine Vision Conference*, pages 33–42. BMVA, September 1999.

[22] G.Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14(2):210–211, 1973.

[23] G.Johansson. Visual motion perception. *Sci. Am*, 6(232):76–88, 1975.

[24] H.Graf, E.Casatto, and T.Ezzat. Face Analysis for the synthesis of Photo-Realistic Talking Heads. In *Proceedings of the 4th Internation Conference on Automatic Face and Gesture Recognition*, pages 189–194, March 2000.

[25] H.Sidenbladh, F.D.Toerre, and M.J. Black. A Framework for Modelling the Appearance of 3D Articulated Figures. In *Proceedings of the 4th International Conference on Automatic Face and Gesture Recognition*, pages 368–375. IEEE, March 2000.

[26] I.Haritaoglu, D.Harwood, and L.Davis. W4: Who? when? where? what? a real time system for detecting and tracking people. In *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 222–227, 1998.

[27] I.Kakadiaris and D.Metaxas. Model-Based Estimation of 3D Human Motion with Occlusion Based on Active Multi-Viewpoint Selection. In *CVPR*, San Francisco, June 1996.

[28] J.Aggarwal and Q.Cai. Human Motion Analysis: A Review. *Computer Vision and Image Understanding*, 73(3):428–440, March 1999.

[29] J.Davis and A.Bobick. The Representation and Recognition of Action Using Temporal Templates. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 1997.

[30] J.Davis and A.Bobick. A Robust Human-Silhoutte Extraction Technique for Interactive Virtual Environments. In *Modelling and Motion Capture Techniques for Virtual Environments*, Geneva, Switzerland, November 1998.

[31] J.Davis and A.Bobick. Virtual PAT: A Virtual Personal Aerobics Trainer. In *Proc. of the Workshop on Perceptual User Interfaces*, November 1998.

[32] J.Deutscher, B.North, B.Bascle, and A.Blake. Tracking through singularities and discontinuities by random sampling. In *Proceedings of the Seventh International Conference on Computer Vision*, pages 1144–1149, September 1999.

[33] J.Foley, A.Damm, S.Feiner, and J.Hughes. *Computer Graphics: Principles and Practice.* Addison Wesley, 1991.

[34] J.MacCormick and A.Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. of the International Conference on Computer Vision '99*, pages 572–578, September 1999.

[35] J.MacQueen. Some methods for classification and analysis of multivariate observation. In *Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.

[36] J.Moody and C.Darken. Fast learning in networks of locally tuned processing units. *Neural Computation*, 1(2):281–294, 1989.

[37] J.Ng and S.Gong. Learning pixelwise signal energy for understanding pixel semantics. In *BMVC*, September 2001.

[38] J.Regh. *Visual Analysis of High DOF Articulated Objects with Application to Hand Tracking.* PhD thesis, School of Computer Science, Carnegie Mellon University, April 1995.

[39] J.Regh and T.Kanade. Model-Based Tracking of Self-Occluding Articulated Objects. In *Proc. of the 5th International Conference on Computer Vision*, pages 612–617, Cambridge, MA, June 1995.

[40] J.Rourke and N.Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. PAMI*, 2:522–536, 1980.

[41] J.Sherrah and S.Gong. Fusion of Perceptual Cues for Robust Tracking of Head Pose and Position. In *Pattern Recognition: Special Issue on Data and Information Fusion in Image Processing and Computer Vision*, 2000.

[42] J.Sherrah and S.Gong. Tracking Discontinuous Motion using Bayesian Inference. In *Proc. of the Sixth European Conference on Computer Vision*, Dublin, Ireland, June 2000.

[43] J.Sullivan, A.Blake, M.Isard, and J.MacCormick. Object Localization by Bayesian Correlation. In *Proc. of the International Conference on Computer Vision*, pages 1068–1075, September 1999.

[44] J.Wang and E.Adelson. Layered representation for motion analysis. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–366, 1993.

[45] K.Akita. Image sequence analysis of real world human motion. *Pattern Recognition*, 17(1):73–83, 1984.

[46] K.Sung and T.Poggio. Example-based learning for view-based human face detection. Technical report, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, December 1994.

[47] L.Xu and D.Hogg. Neural Networks in Human Motion Tracking. *Image and Vision Computing*, 1997.

[48] M.Black and A.Jepson. EigenTracking: Robust matching and tracking of articulated objects using a view based representation. *Int. J. Computer Vision*, 1(29):5–28, 1998.

[49] M.Brand. Structure learning in conditional probability models via an entropic prior and parameter extinction. Technical report, Mitsubishi Electric Research Laboratory, Mitsubishi Electric Information Technology Center America, August 1998.

[50] M.Brand. Pattern discovery via entropy minimization. In *Proceedings of Uncertainty '99 (AI and Statistics)*, 1999.

[51] M.Brand. Shadow Puppetry. In *Proc. of the Seventh Internation Conference on Computer Vision*, pages 1237–1244, Kerkyra, Greece, September 1999.

[52] M.Isard and A.Blake. Condensation - conditional density propogation for visual tracking. *Int. J. Computer Vision*, 1998.

[53] M.Jenkin. Tracking Three Dimensional Moving Light Displays. In *Proc. of the International Workshop on Motion: Representation and Perception*, pages 171–175. Elsevier, 1986.

[54] M.Jones and T.Poggio. Model-Based Matching of Line Drawings by Linear Combination of Prototypes. Technical report, M.I.T. A.I. Lab, 1995.

[55] M.Leung and Y.Yang. First sight: A human body outline labelling system. *IEEE Trans. PAMI*, 17(4):359–377, 1995.

[56] M.Leventon. Bayesian estimation of 3-dhuman motion from an image sequence. Technical report, Massachusetts Institute of Technology, Cambridge, MA 02139, July 1998.

[57] M.Minsky. Steps towards artificial intelligence. In *Proceedings of the Institute of Radio Engineers*, pages 8–30, 1961.

[58] M.Murray. Gait as a Total Pattern of Movement. *American Journal of Physical medicine*, 46(1):290–333, 1967.

[59] M.Silaghi, R.Plànkers, R.Boulic, P.Fua, and D.Thalmann. Local and Global Skeleton Fitting Techniques for Optical Motion Capture. In N. Magnenat-Thalmann and D. Thalmann, editors, *Modelling and Motion Capture Techniques for Virtual Environment, CAPTECH'98*. Springer-Verlag, 1998.

[60] M.Sonka, V.Hlavac, and R.Boyle. *Image Processing, Analysis and Machine Vision*. Thomson computer press, 1993.

[61] M.Yamamoto, A.Sato, and S.Kawada. Incremental Tracking of Human Actions from Multiple Views. In *CVPR*, 1998.

[62] M.Yamamoto, Y.Ohta, T.Yamagiwa, and K.Yagishita. Human Action Tracking Guided by Key-Frames. In *Proceedings of the 4th International Conference on Automatic Face and Gesture Recognition*, pages 354–361. IEEE, March 2000.

[63] N.Oliver, B.Rosario, and A.Pentland. A Bayesian computer vision system for modelling human interactions. In *Proceedings of the 1st International Conference on Computer Vision Systems*, pages 255–272, January 1999.

[64] M. Oren, C.Papageorgiou, P.Sinha, E.Osuna, and T.Poggio. Pedestrian Detection Using Wavelet Templates. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[65] P.KaewTraKulTong and R.Bowden. Adaptive visual system for tracking low resolution colour targets. In *BMVC*, September 2001.

[66] Q.Delamarre and O.Faugeras. 3-D articulated models and multi-view tracking with silhouettes. In *Proc. of the IEEE International Conference on Computer Vision*, pages 716–721, September 1999.

[67] R.Bowden, T.Mitchell, and M.Sarhadi. Reconstructing 3D Pose and Motion from a Single Camera View. In *BMVC*, pages 904–913, Southhampton, 1998.

[68] R.Kass and A.Raftery. Bayes factors and model uncertainty. Technical Report 254, University of Washington, 1993.

[69] R.Polana and R.Nelson. Low level recognition of human motion (or how to get your man without finding his body parts. In *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 77–82, Austin, 1994.

[70] R.Rosales and S.Sclaroff. Learning and synthesizing human body motion and posture. In *Proceedings of the 4th International Conference on Face and Gesture Recognition*, pages 506–511. IEEE, March 2000.

[71] S.Gong, E.Ong, and S.McKenna. Learning to associate faces across views in vector space of similarities to prototypes. In *Proc. British Machine Vision Conference*, Southampton, September 1998.

[72] S.Gong, E. Ong, and P. Loft. Appearance-based face recognition under large rotations in depth. In *Proc. Asian Conference on Computer Vision*, Hong Kong, January 1998.

[73] S.Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall International Editions, 1994.

[74] S.McKenna and S.Gong. Tracking Faces. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pages 271–277, Vermont, US, October 1996.

[75] S.McKenna, Y.Raja, and S.Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17:225–231, 1999.

[76] S.Ullman and R.Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992–1005, October 1991.

[77] T.Cootes, C.Taylor, D.Cooper, and J.Graham. Active Shape Models-their training and applications. *Computer Vision and Image Understanding*, 1(61):38–59, 1995.

[78] T.Cootes, G.Edwards, and C.Taylor. Active Appearance Models. In *Proc. of ECCV '98*, pages 485–498, 1998.

[79] T.Darrell and A.Pentland. Robust estimation of a multi-layered motion representation. In *Proc of the IEEE Workshop on Visual Motion*, pages 173–178, 1991.

[80] T.Darrell and A.Pentland. Space-time gestures. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 335–340, New York, 1993.

[81] T.Heap. *Learning Deformable Shape Models for Object Tracking.* PhD thesis, School of Computer Studies, University of Leeds, UK, September 1997.

[82] T.Heap and D.Hogg. Towards 3-D hand tracking using a deformable model. In *FG'96*, pages 140–145, 1996.

[83] T.Heap and D.Hogg. Improving specificity in pdms using a hierarchical approach. In *BMVC*, pages 80–89, Essex, UK, September 1997.

[84] T.Moeslund and E.Granum. Multiple Cues used in Model-Based Human Motion Capture. In *Proceedings of the 4th Internatinoal Conference on Automatic Face and Gesture Recognition*, pages 362–367. IEEE, March 2000.

[85] T.Vetter. Synthesis of novel views from a single face image. *International Journal of Computer Vision*, 28(2):103–116, 1998.

[86] T.Vetter and T.Poggio. Linear Object Classes and Image Synthesis From a Single Example Image. *Pattern Analysis and Machine Intelligence*, 19(7):733–741, July 1997.

[87] T.Vetter and V.Blanz. Estimating coloured 3-D face models from single images: An example based approach. In *Proceedings of the ECCV'98*, Freiburg, Germany, 1998.

[88] Y.Ivanov, C.Stauffer, A.Bobick, and W.Grimson. Video surveillance of interactions. In *IEEE Workshop on Visual Surveilance*, Ft. Collins, CO, 1999.

[89] Y.Iwai, K.OGaki, and M.Yachida. Posture Estimation using Structure and Motion Models. In *Proceedings of the 7th International Conference on Computer Vision.* IEEE, March 1999.

[90] Y.Wu and T.Huang. Capturing Articulated Human Hand Motion: A Divide-and-Conquer Approach. *Proc. of ICCV'99*, pages 606–611, 1999.