Using Low-Level Motion for High-Level Vision

Ben Daubney

A dissertation submitted to the University of Bristol in accordance with the requirements for the degree of Doctor of Philosophy in the Faculty of Engineering, Department of Computer Science.

July 2009

Abstract

The work presented within this Thesis describes a method to extract the 3D pose of a human performing a gaited action using only the motion of a sparse set of tracked features. The features used are not extracted using specific limb detectors, but are selected and tracked automatically using a standard feature tracker. These features contain noise that is not Gaussian in nature, but systematic, as a result of edge effects. Furthermore, features are indiscriminately tracked on both the subject of interest and the background. This ensures that a method designed to exploit only the structure of the features would fail, the motion of the features must be exploited to extract useful information.

A method is presented that models the expected motion of a feature tracking a particular limb. Using these models the likelihood that each observed motion is caused by a specific limb can be estimated. Furthermore, this representation allows the temporal state of an action to be estimated for each feature independently. Integrating over all features, a Hidden Markov Model is shown to be able to accurately extract gait phase.

Given these initial likelihood estimates, an approach is presented to create probability maps that describe the likelihood of a limb being located at each position in the image. Following this, 2D pose in the image plane can be estimated. This is achieved using a Pictorial Structures representation that uses phase dependent priors. This search is performed via Dynamic Programming and conducted in each frame independently. Extracted poses are then enforced to be temporally coherent using a high-level motion model.

Finally, these methods are shown to be suitable for 3D pose estimation. Methods are presented to extract 3D motion from 2D image observations and to efficiently extract pose in $\mathbf{R}^3$. This is achieved by placing constraints on the search space and by mapping observations from the image plane into $\mathbf{R}^3$. To extract 3D pose 3D models are learnt directly from motion capture data no image data is used for training. Quantitative results are provided using the HumanEva data set. The approach is tested on a variety of scenes filmed from different viewpoints, with no prior expectation of the path the subject will walk through the scene. Despite this the same 3D model is used throughout. Unlike many current approaches, the presented method requires no initialisation of joint locations in the first frame, this is performed automatically. Furthermore, only a single viewpoint is ever exploited. The work in this thesis demonstrates that a set of sparse motion features contains enough information to extract the 3D pose of a person whilst performing gaited actions.

Declaration

I declare that the work in this dissertation was carried out in accordance with the Regulations of the University of Bristol. The work is original except where indicated by special reference in the text and no part of the dissertation has been submitted for any other degree.

Any views expressed in the dissertation are those of the author and in no way represent those of the University of Bristol.

The dissertation has not been presented to any other University for examination either in the United Kingdom or overseas.

SIGNED: DATE:

Acknowledgements

Firstly, I would like to thank Neill Campbell and David Gibson for their continued support and advice throughout my PhD. I have learnt a huge amount from them that has no doubt put me in good stead for the future. I am also grateful to Dave for the initial guidance and help he provided, this made the difficult transition into a new field and subject that much easier.

I would also like to thank Sion Hannuna, Dave Oziem, Lisa Gralewski, Tilo Burghardt, John McGonigle, Oli Cooper, Martin Thirkettle, Richard Sherley, Fang Siyuan, Frances Beckett and Janko Calic for their help and friendship during my time at Bristol. I am further indebted to both Sion and Franky for their constant encouragement which made this process more enjoyable than otherwise may have been.

Furthermore, I would like to thank my friends from outside the university who have often provided a useful distraction from my work. In particular Andy, Marcus, Dai, Tracey, Emma, Gareth, James and Mark.

For the support of EPSRC, who provided funding and made this research possible, I am also extremely grateful.

Above all, my heartfelt thanks goes to my parents and family who have provided endless encouragement, motivation and support. I know I wouldn't have got this far was it not for the opportunities that they have provided me with.

Publications

The work described in this thesis has been presented in the following publications:

1. B. Daubney, D. Gibson and N. Campbell. Monocular 3D Human Pose Estimation using Sparse Motion Features. *2nd IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS2009) - Held in conjunction with ICCV*, 2009.

2. B. Daubney, D. Gibson and N. Campbell. Real-Time Pose Estimation or Articulated Objects using Low-Level Motion. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

3. B. Daubney, D. Gibson and N. Campbell. Estimating Gait Phase using Low-Level Motion. *IEEE Workshop on Motion and Video Computing (WMVC)*, 2008.

4. B. Daubney, D. Gibson and N. Campbell. Using Low-Level Motion to Estimate Gait Phase. *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2008.

Contents

Chapter 1

Introduction

1.1 Introduction

There has long been an interest within the computer vision community to extract 3D pose from a person observed in a sequence of images. Whilst humans can perform this task with relative ease, extracting pose automatically, without the use of markers remains an unsolved problem. There have been many advances made over the last 20 years in the development of robust algorithms to extract pose, however, little attention has been paid to whether the correct type of cues are being used; the majority of approaches rely on exploiting the shape of binary silhouettes.

In the psychophysics community the human perception of a particular sparse stimulus has been studied for over 30 years. This stimulus, originally reported by Johansson [53], represented each of the main limbs as a single point. This representation contains no information about the shape or appearance of the actor and when stationary revealed little information. However, when set into motion, an observer could quickly recognise the sparse sets of points as a person. This type of stimulus was described as the Moving Light Display (MLD) and demonstrated that the human visual system was capable of extracting large amounts of data from the motion of just a few sparse points.

Motivated by these experiments this thesis examines whether the motion of a similar sparse set of features could be exploited to extract 3D pose from a single view. This has real worth to the computer vision community who currently almost exclusively rely on only appearance cues. An improved understanding of how to exploit motion could be used to improve many state of the art techniques, which currently discard this information rich cue.

However, automatically extracting a MLD representation from a sequence of images is in itself a very difficult problem. To achieve this, the localization of each of the main limbs first has to be estimated, this is the very problem this thesis attempts to solve. Using a MLD created in this manner and then attempting to extract pose would make little contribution to the research community. Instead a standard feature tracker is used to automatically select and track features across a sequence of images. If each feature being tracked is displayed as a single point a similar stimulus appears to that of the MLD. However, the key difference is that the display contains points that track both the background of the image and the subject being observed. Furthermore, the motion of the features contain noise, often a feature will become lost only to be replaced in a completely different location. Also entire limbs may not be tracked for a complete gait cycle or tracked at all. However, from this automatically generated MLD an observer can still recognise the presence of a person. An example of a normal MLD display and that of an automatically extracted display is presented in Figure 1.1.

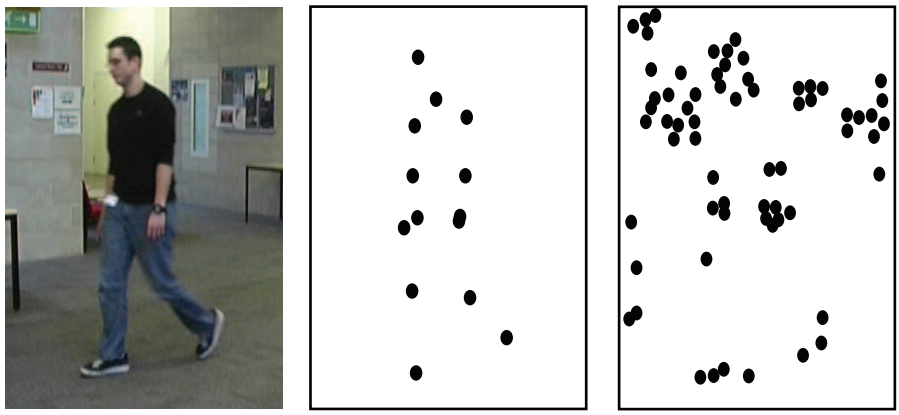


Figure 1.1: *Comparison of a MLD display for the frame shown on the left. An original MLD display is shown in the middle and an automatically extracted representation using the KLT feature tracker is shown on the right.*

Whilst there are many feature trackers and any would be equally valid, for all of the experiments contained in this thesis the Kanade-Lucas-Tomasi (KLT) feature tracker is used [88]. The feature tracker is used primarily as a black box and default parameters are used throughout. The exception to this is when even human perception of the displays becomes difficult, this occurs if a particularly cluttered scene is used and the majority of features track the background. However, no optimization is ever performed over KLT parameters and any changes made will be clearly stated.

In this thesis focus is placed on extracting pose for a particular subset of actions, those being gaited actions. Whilst this does effect the generality of any methods presented, many current state of the art methods are action specific, they have a prior expectation of what action will be performed.

1.2 Thesis Statement

The primary claim made in this work is that *it is possible to extract the 3D pose of gaited actions using only the motion of a sparse set of features automatically extracted and using only a single viewpoint.*

This claim is largely motivated by the experiments performed by the psychophysics community using the MLD. Whilst these experiments don't explicitly make clear that 3D pose is extracted before recognition can commence, here, it is assumed that this does occur or at least is feasible.

The emphasis on using automatically extracted features is for the reasons outlined in the previous section. However, using this set of features containing noise and that track both foreground and background will ensure that any methods described are dependent on motion. The subject in each sequence will be masked in a dense set of features meaning that simply attempting to exploit the structure of the KLT features will yield very little meaningful information.

A single viewpoint is used as this is how stimuli are presented to observers in psychophysics experiments. Whilst it is likely the use of more cameras would improve accuracy, it is assumed that this task should be achievable using a single viewpoint.

1.3 Contributions

In the presented work the main goal is to demonstrate that 3D pose can be estimated using just sparse motion features, however, some more specific contributions are listed below:

Extracting information from motion: This thesis demonstrates that the motion of a tracked feature, considered independently, contains information that can be extracted about the temporal state of a particular action and the likely origin of the feature. A novel representation of motion is provided to encode this information.

Gait phase extraction: A method is presented for extracting gait phase from the motion of the features.

Creating dense probability maps from sparse observations: A method is presented that allows dense probability maps to be created from sparse motion observations. This allows pose to be estimated without constraining limbs positions only to locations where features are detected.

Extracting 3D motions from 2D observations: It is shown that, for gaited actions 3D motion can be extracted from a 2D observation by assuming that the dominant motion occurs in the direction of travel.

Enforcing Temporal Coherence: A novel method is presented to enforce the temporal coherence of extracted poses across consecutive frames.

Efficient bottom-up 3D pose estimation: A method is presented so that bottom-up searches can efficiently be performed in $\mathbf{R}^3$. This is achieved through search space constraints and by mapping observations in the image plane into $\mathbf{R}^3$. This allows searches to take place directly in $\mathbf{R}^3$.

1.4 Thesis Outline

This section describes the work contained in each of the proceeding chapters.

In Chapter 2 a review of the relevant background literature is presented. Firstly, the motivation for this work is described and findings from psychophysical experiments using the MLD are discussed. Relevant questions that have arisen from these experiments are highlighted. Following this, a review of methods to detect humans and suspicious behaviour are presented. Approaches that are both motion based and appearance based are reviewed. Next, methods to discriminate between different actions are discussed, this is of particular interest since this is an area where motion is often exploited, not just for localisation, but to extract specific information. Then a review is given of techniques currently used to estimate pose, current trends are highlighted and discussed. Finally, a summary of motivations from the current literature is presented.

In Chapter 3 the representation of motion used throughout this thesis is presented. Models used are two dimensional and learnt from hand labeled training data of people walking on a treadmill. The models represent the trajectory a feature tracking a specific limb would be expected to follow as a person is walking. A method is presented to match observations to these models in a probabilistic framework and the models are investigated in three ways; Firstly, how well they can segment features tracking the foreground and the background. Secondly, how well each model can discriminate which limb a particular feature is tracking. Thirdly, how well the models can be used to recognise the temporal state or gait phase of the action being performed.

Chapter 4 uses the motion models described in Chapter 3 to estimate 2D pose in the image plane. From the previous chapter the temporal state of the action and the likelihood of each feature tracking a particular limb is known. Using these likelihoods dense probability maps are created that describe the probability of a limb being located at a particular position in the image plane. These are designed so that the position of a limb is not enforced to lie at the location of a tracked feature, however, the formulation does make this preferable. A 2D articulated model is created, this model has a different set of model parameters for each phase of gait. These are learnt from the same training data used in Chapter 3. Finally, a bottom-up technique using Dynamic Programming is presented to estimate pose given these dense probability maps, a method of enforcing temporal coherence is also introduced. This technique is shown to be applicable to both bipeds and quadrupeds.

Chapter 5 extends the approaches presented in the previous two chapters to extract 3D pose. 3D motion and spatial models are learnt from 3D motion capture data. A method is presented to estimate the trajectory of the subject in each scene, this provides information about their position in each frame and also their orientation. Given knowledge of their orientation the 3D motion of a feature measured in $\mathbf{R}^3$ that resulted in an observed 2D motion in the image plane can be extracted. This allows 3D motion models, similar to those presented in Chapter 3, to be used. A similar bottom-up method is used to estimate 3D pose as in Chapter 4, however, this search is conducted in $\mathbf{R}^3$. The location of observed features in the image plane is traced through $\mathbf{R}^3$, following which dense probability volumes can be created. The search can then be performed directly in $\mathbf{R}^3$ which allows the presented method to be considerably faster than many current approaches that require 3D models to be repeatedly projected into the image plane for comparison with observations. Both qualitative and quantitative results are provided for walking and jogging.

In Chapter 6 concluding comments are provided about the work in this thesis and ideas for possible future work are presented.

Chapter 2

Background

There is currently much interest in being able to automatically locate, recognize and track people in a sequence of images. Achieving this has a number of applications such as surveillance, human computer interaction (HCI), gait analysis and 3D animation. The level of accuracy needed is largely dependent on the application. For example, many surveillance applications may only require the location of each person to be known, whereas for animation full body pose must be estimated. Furthermore, each application may have constraints on the time required to process each frame, whilst surveillance will undoubtedly require real-time performance, motion capture could be performed offline.

The solution provided for each application is designed to accommodate these operational constraints and can be interpreted as asking an entirely separate question about the sequence provided. Some commonly asked questions and their descriptions are listed below:

- Human Detection - Is a person present and where are they?
- Human Tracking - Given a person's location in frame t, what is their position in frame $t + 1$?
- Activity Recognition - Given a person is present, what activity are they performing?

- Suspicious Behavior Detection - Given a person is present, are they behaving unusually?
- Gesture Recognition - Given a person is present, what gesture are they performing?
- Pose Detection - Given a person is present, how are their limbs configured?
- Pose Tracking - Given a person's pose is known in frame t, what is their pose in frame $t + 1$.

Whilst there is often some crossover, each individual question represents an entire subfield. The question being answered in this thesis is a combination of the bottom two; given someone is present how are their limbs configured and can their configuration be estimated over a sequence of frames.

In this thesis motion is used both to detect and track 3D pose. Whilst motion has rarely been used in pose estimation, it is frequently exploited in areas such as human detection and action recognition. Therefore, whilst the main focus of this background review will be on methods to estimate and track pose, attention will also be paid to these areas. Methods that exploit both dynamic and static cues will be discussed to understand the advantages that motion provides.

As discussed in the previous chapter the motivation for this thesis arises from experiments using the Moving Light Display (MLD). For this reason work carried out by the Psychophysics community to try and understand human perception of biological motion is also explored.

2.1 Psychophysical Experiments using the Moving Light Display

Johansson first showed that humans could extract high level information from a sparse set of moving lights [53]. These displays were created by attaching a light bulb to each of the main joints of an actor and then filming the actor performing motions such as walking or running. Whilst static, the set of lights had no obvious

interpretation, however, when set in motion the lights became instantly recognisable as that of a human. The importance of this type of stimulus is that all visual cues were removed, in a sense this was the minimal representation of the human form. Furthermore, experiments that followed suggested that biological motion was perceptually special. For example, Johansson performed experiments using normal MLD displays of people walking and displays created using a puppet's walk [52], it was found that recognition deteriorated when presented with the puppets walk over that of a person. This suggested the motion of living things, biological motion, was processed and recognised differently from non-biological motion. It seemed likely that there were evolutionary benefits to being able to quickly recognize biological motion, to make a quick get away in the presence of a predator or equivalently, to identify a potential meal.

However, it was shown that not only could biological motion quickly be recognized, but that high-level information could be extracted such as gender [65], the subject's emotion [28], whether the person performing the action was known by the observer [20] or even whether the person performing the action was the observer [63]. It seemed that biological motion contained a wealth of information that the human visual system was capable of extracting.

Mather and West [66] investigated how well people could distinguish between the point light walkers of different animals, it was found that a static representation of each animal produced near chance recognition whereas a dynamic display produced much higher recognition rates. This demonstrated that human perception of biological motion was not just constrained to human motion.

The effect of altering the MLD stimulus was also investigated. In [27] the effect of inverting the MLD was explored and shown to significantly reduced performance. This suggested that the structural cues in recognizing the MLD play a significant role rather than just the local motion of each point. However, in earlier work by Sumi [97] it was reported that when inverted displays were presented to subjects they simply inferred the pattern as the motion of an upright individual, suggesting that humans' have a strong prior over orientation.

It has also been suggested that biological motion is integrated for longer time frames

than simple translation [72]. Using a stimulus created from a sub sample of the true MLD and masked by up to 1000 randomly selected points, participants were asked to discriminate the direction of the point light walker, compared to randomly selected features that were given uniform translation motion. It was shown that the biological motion stimulus was integrated for up to 3 seconds, however, simple translation was integrated for about 600ms.

Whilst the interpretation of this experiment is that neurological processing of biological motion requires a longer integration time, Thirkettle *et al.* [99] suggest that certain parts of the gait cycle are simply more recognisable than others. They perform experiments examining the relationship between accuracy and the starting phase of gait that a stimulus is presented in. It was found that parts of the gait cycle when the legs are at the furthest displacement from one another are the most descriptive. Furthermore, they reproduced the experiments using both moving point light displays and static examples. They found that although motion improved accuracy, the structure of the features is far more important than first thought. When the limbs were at maximum displacement it was found that the static representation was as descriptive as the moving representation. The conclusion could therefore be that biological motion is integrated for longer as the viewer must wait for certain characteristic poses to appear before being able to interpret the moving light display. Previous comparisons between moving and static displays could be biased by the set of poses displayed a dynamic display is far more likely to include very descriptive phases of gait compared to a static display that shows only a single phase of gait.

Methods to explain the interpretation of the moving light display have included the use of rigid planar motion [49]. This suggests that structure of the MLD can be recovered by assuming that motion occurs mainly in the direction of travel and limbs are joined by rigid bones. However, this interpretation can not be used to explain non-gait motions such as dancing or gesturing. In fact it is not clear whether interpretation of the MLD requires the extraction of full 3D pose or it is a problem in pattern recognition. Troje [100] assumes it is a problem in pattern recognition, PCA (Principal Component Analysis) is used to learn a gait space for male walkers and female walkers, gaits with differing amounts of maleness or femaleness could then be linearly interpolated between the two models. This approach assumes that the structure of the features is of importance and that the change in this struc-

ture represents motion, this can be described as high-level motion. Approaches that exploit low-level motion assume that the local trajectories of individual parts are most significant. This is as emphasis is placed on the movement of each light without reference to higher level structures, such as the configuration of the other point lights. These two techniques could be regarded as top-down and bottom-up respectively. Whilst top-down approaches assume the high-level structure is of importance, bottom-up approaches assume it is the local motion of each point that is of primary importance.

Whilst experiments using the moving light display have demonstrated that large amounts of high-level information can be extracted from a sparse set of features, the processes that allow this information to be extracted are still not clear. Perhaps the biggest open question is whether the interpretation of the MLD is based on high-level recognition of changes to the structure of the MLD or whether it is low-level recognition of local motion?

Another open question is whether observers need to extract the 3D pose of the person represented by a point light walker or whether MLD interpretation is a 2D pattern recognition problem. As described in the thesis statement the interest here is in extracting 3D pose from sparse motion features. The work described in this section makes no suggestion about human performance at inferring 3D pose, so the proposal of a computer vision approach to achieve this, particularly in the presence of noisy data, is of real interest to those in the field of biological motion.

2.2 Human Detection

With the ever increasing number of CCTV cameras in use there is a growing need to automate the process of monitoring each camera for suspicious or unusual behaviour. In a 2006 report written for the Information Commissioner it was claimed that the number of CCTV cameras in the UK had risen to 4.2 million [1] (one for every fourteen people), it is simply not possible to manually monitor this many cameras effectively. However, it is not just real-time online surveillance that is of interest, often after an event has been reported the incident must then be found in hours of

CCTV footage and often it would be desirable to be able to link a person or vehicle across several cameras.

As realtime performance is often desirable in surveillance systems motion cues, which if using simple image differences are fast to compute, have often been exploited to infer whether there is a person present or not. In particular, periodic motion is used as a strong cue for biological motion [19, 46, 75].

Polana and Nelson [75] use the difference in intensity between consecutive images as a measure of motion, this can be used for foreground-background segmentation to produce a set of blobs. Each blob is then partitioned into a grid and the motion measured as a function of time. The magnitude of motion at each grid position produces a one dimensional signal across time; these signals can then be inspected for periodicity. A similar approach is used by Cutler *et al.* [19], but rather than using a grid to split each blob into separate parts, the self similarity of the blob over time is measured and Fourier analysis is used to detect periodic motion. The approach is relatively simple yet detects moving people accurately, furthermore the approach demonstrates a car; which moves without exhibiting periodic motion, is not detected as a person.

Gibson *et al.* [46] use the periodicity of features tracked using the KLT feature tracker to detect quadrupeds in sequences of images. To overcome the sparse and noisy tracking the object is split into quadrants, the motion of each quadrant is then taken to be the average motion of the features within it. Vertical motions are shown to be particularly robust for periodicity detection and are insensitive to any net horizontal displacements. A prior learnt eigengait model can then be warped to the observations that allows pose to be estimated in each frame.

Viola *et al.* [105] use a cascade of simple image filters to detect pedestrians. The set of filters used are selected using AdaBoost, this allows a series of weak classifiers to be constructed in such a way that high detection and low false-positive rates can be achieved. The filters are applied to both image data and simple motion data, extracted from consecutive frames through image difference. Even using this simple measure of image motion is shown to increase performance compared to only using static image data.

Whilst these approaches use motion for discrimination, they don't attempt to model or reason about its underlying cause. Song *et al.* [95] use a sparse set of features, tracked via the KLT algorithm, to detect the presence of a person in a scene. This is achieved by learning a compact representation of both the structure and motion of a person walking. They use a triangulated graphical structure and marginalize over all possible model configurations. The maximum a posterior labelling can also be extracted which gives an indication of the pose and localization of the subject in each frame. They show that the presented technique can distinguish between the motion of a person and a moving chair; however, it is unclear whether the motion or the structure of the extracted features is most significant in this result. For each limb the motion model learnt is represented as a single Gaussian, which as the authors suggest, does not particularly well describe the training data. Furthermore, whilst they present a method so that information can be processed over a number of frames, they assume that each feature is tracked over only two frames. This is largely as their representation of motion is not descriptive enough to exploit the complexity of biological motion over a longer time frame.

The motion extracted from KLT features was also exploited to count and track people in crowded scenes [76]. The observed trajectories of the features were clustered based on their location and motion. Motion clustering used RANSAC to find sets of features that moved across frames under affine transformation, this condition was then used to form or grow different clusters. One of the limits with this approach is that it is assumed that the sets of features being tracked are members of a rigid object. Attempting to track an entire person using this method would result in separate clusters for each part of the body. However, the authors demonstrate this approach on sequences where just the upper body is available, where the rigidity assumption is more likely valid, or a high camera angle is used that also alleviates this problem since the legs and feet are generally occluded.

The appearance of a person is also often used to detect and track people. Baumberg and Hogg [7] use active contours to detect and track the silhouette of a person through a sequence of frames, a Kalman filter is used to update the model at each frame. Gavrilla [45] uses a hierarchy of shape templates to detect pedestrians in each frame independently. Each shape is represented by its outline and observations are compared using the Chamfer distance. The advantage of this approach is that there

are no expectations of the pedestrians to be moving, so stationary people can also be detected. Dimitrijevic *et al.* extend this idea using spatio-temporal templates [26]. This method allows information from a number of frames to be integrated by using a sequence of shapes rather than just a single instance. This is shown to reduce the false-positive rate, this is also achieved by weighting different parts of the Chamfer match depending on how discriminative each part is, for example it was found that the legs are far more discriminative than the torso.

The problem with these template matching approaches is that performance degrades in the presence of noise or occlusions. This can be overcome by also searching for local features. Leibe *et al.* use difference of gaussian patches to find a set of local features [61], using these patches, if some parts are occluded it would be hoped that the majority were still present to discriminate whether or not there was a pedestrian present. This approach was shown to detect multiple people walking simultaneously in groups.

This approach is also investigated by Dalal and Triggs [21] who use a dense set of Histograms of Orientated Gradients (HOG Features) to detect people. Training examples of people are split into a dense grid of patches and for each patch a HOG feature is extracted. Each feature is created by binning the gradient at each pixel location, weighted by the gradient magnitude. Given a training set of positive and negative examples a Support Vector Machine (SVM) is used to learn a non-linear classification boundary.

This idea is further expanded by Felzenszwalb *et al.* so that after initial detections a smaller scale search is performed to attempt to detect individual parts [36]. This approach is shown to improve false-positive rates. This is also in part improved by their search for hard negatives, which are used to retrain the classifier. Hard negatives are examples that after initial classifier training are misclassified as detections. The classifier is then retrained using this set of hard negatives. The advantage of this approach is that the set of images provided as negative examples (i.e. that don't contain a person) are likely to be very broad and capture varied scenes, mining for hard-negatives finds the set of scenes that are most similar to the class which is to be detected, without being a member of it, so will train the classifier to exploit the subtleties between the classes.

These appearance based methods have became more popular and are largely driven by the need to detect people in single images, this is of particular interest for semantic labelling of the huge number of pictures available on the internet rather than surveillance applications. The clear advantage of these approaches is that they do not assume the subject to be moving. However, most approaches developed to detect people will exploit motion if it is available. Motion has been shown to be a strong cue for finding areas of interest in images. Methods that exploit motion are generally more robust than those that don't and are also less complex. Less complex solutions are of interest for two principal reasons. Firstly, real-time performance is more likely. Secondly, and perhaps most importantly, it's more intuitive to understand how these systems work and when they will break. This makes it easier for the operator of any system to understand its limitations and use it more effectively.

2.3 Detecting Suspicious Behaviour

Whilst being able to detect and locate people is of interest, for many applications it would prove far more useful to be able to say whether given a detection of a person they are behaving suspiciously. What constitutes suspicious is usually defined as anything that is not normal or usual. It should be noted that suspicious behaviour detection does not in general address the problem of detecting specific criminal activity, it is simply an attempt to detect activity that warrants further investigation by a surveillance operator.

Adam *et al.* [2] learn the motion flows at determined points in the image. Unusual behaviour is then any motion that occurs different to the expected motion flows. For example, a person running through a slowly moving crowd or a person walking against the flow of traffic would cause detection. The benefit of this approach is its relative simplicity, models can be learnt online and slowly adapt as changes to the expected motion patterns occur.

Zhong *et al.* [111] detect unusual activity in a sequence of images by representing sequences using motion and image features. A set of prototype features or codewords are learnt using K-means on the training data provided for learning. The

co-occurrence between different prototypes are learnt so that unusual activities are then described by those sets of features which do not often occur simultaneously in the training set.

Vaswani *et al.* [104] track multiple people by modeling the deformation of a shape where each person represents a vertex, any deviations from the expected trajectory of the shape results in a detection. Boiman and Irani [11] also use shape, they collect a small amount of training data from the scene of interest, then given an unseen sequence explore how much of the previously unseen sequence can be explained using the training data. Any sequences which can not be sufficiently explained by the training data are described as suspicious.

Rather than simply learning peoples behaviour Dee and Hogg [22] attempt to reason about a normal persons motivation for the actions they take. For example in a car park people tend to walk from their car to an exit point and vice versa when returning. This motion is usually direct, unless their trajectory is affected by for example a moving car. If a person is spotted loitering or walking in an unexpected trajectory, their behaviour is flagged as suspicious. This kind of approach is particularly appealing since rather than just being trained on examples of normal behaviour as in the previous approaches, it models higher level reasoning about a detected person's intentions.

It is interesting that the methods described to detect suspicious behaviour are almost exclusively trained on normal activity, the task to detect suspicious behaviour is then the case of finding events that are different from the training set. This is different to approaches presented in the previous section where often large numbers of both positive and negative examples are required for training. This is perhaps as normal and suspicious behaviour, as two separate classes, are very poorly defined.

Motion is commonly used in these approaches and whilst methods such as Adam *et al* [2] don't explicitly need to track people, many of the other methods described first need to segment and track the person in each frame. Following this the shape of the trajectory is often used to determine whether the observed motion is suspicious or not. The trajectories of low-level motion features will be exploited in this thesis. However, unlike the work presented in this section, it will not be known what the

trajectories are representative of, whether they are the motion of a foot, hand or even a feature located on the background. This is in contrast to the techniques in this section where it is assumed the observed trajectory is that of a person.

2.4 Action Recognition

Action and gesture recognition is an area where motion is almost exclusively exploited. Performing a particular activity or motion will result in a temporal pattern which can then be used for classification, the main problem is how to efficiently represent and extract this pattern.

Often important parts of the body, e.g. the hands, are first located and tracked; the motion of this trajectory is then used as a feature [44, 81, 107]. Gao *et al.* [44] use face detectors to locate the position of the head and then estimate the position of the hands by placing them where areas of coherent motion are detected. In [107] Hidden Markov Models (HMM) are used to classify the trajectories of the hands. These trajectories are extracted using stereo cameras and are represented as 3D motion tracks. HMM's are particularly well suited for action recognition tasks as they are robust to variation in the temporal length of an action.

In [81] actions are described in terms of 'dynamic instants' and 'intervals'. Given an observed motion of a hand the spatiotemporal curvature of the trajectory carved out by the hand is calculated. This spatiotemporal curvature captures both changes in direction and the magnitude of the velocity as a single positive number. The maxima of the curvature, which represent significant changes in direction or speed, are defined as Dynamic Instants and an 'Interval' is the time between two consecutive instants. This allows an action to be represented as a sequence of instants and intervals, the same action filmed from a different viewpoint should be composed of the same sequence of these descriptors giving the method some invariance to changes in viewpoint. Each interval represents a building block of the action being performed. The benefit of this approach is that only 2D tracking of the hand is required and an action can be recognised from different camera views using only a single representation.

These methods assume that the hands can be accurately tracked; often this is not the case, particularly if they are occluded from view. Another class of approaches, rather than attempting to extract positions of individual limbs, attempts to learn templates of the motion of the entire body [10, 30, 78, 98]. Activity recognition can then be solved as a matching problem between templates and observed motions.

Bobick and Davis [10] use Motion History Images (MHI) and Motion Energy Images (MEI). Given a person performing an action, Motion Energy Images capture a temporally accumulated binary silhouette. This effectively represents motion as shape. Motion History Images are similar but also capture temporally when each motion was observed. This description is much richer and more discriminative than the MEI. However, both methods first assumes a good localization of the observed person in the image.

Rahman and Ishikawa [78] use a similar approach to Bobick and Davis, except rather than using binary silhouettes, edge maps are exploited. Given a motion, edges are extracted and the resultant image is concatenated into a single vector. PCA is then used to reduce the dimensionality of the motion images. Edges are used as the authors claim this alleviates the dress problem, that people wearing different clothes have different appearances.

In the previous approaches the observations used in each algorithm are a set of binary silhouettes or edges. This means that first these must be computed and the algorithm will perform poorly if the output of this operation is noisy. Sun *et al.* use the image motion directly for activity recognition [98]. They learn a probability distribution of observing a given action using Gaussian distributions. The temporal progression of the action is then modeled using a HMM. This approach is shown to be able to recognise different martial art actions.

Rittscher et al. [83] place each image frame back-to-back to create xyt cubes, slices of this xyt cube are then taken along the temporal dimension. Repetitive motions appear as interweaved signals as, for example, the legs cross one another. For each slice a skew parameter is estimated and the values for all slices are constructed into a vector which can then be used for classification.

Efros *et al.* [30] extract separate motion fields for each of the axis in both the positive

and negative direction. These motion fields collected over a number of frames can then be constructed into a motion volume, which acts as the motion descriptor. An unknown sequence can be classified using k-nearest neighbours. This approach assumes that a person can easily be detected in each frame. The authors report high accuracy rates on groups of actions such as Ballet, Tennis and Football.

As with the template methods described in Section 2.2, these approaches suffer with a major flaw, that any occlusion or significant changes in appearance can result in deterioration of performance. Furthermore, exemplar methods often require the storage of the original training set, as often the descriptive features are very large, this places a physical limit on how many training examples are used. Whilst methods such as PCA [78] or extracting Hu moments [10] alleviate some of these problems a more compact and robust set of features needs to be extracted.

To alleviate this, approaches similar to those described in Section 2.2 can be used where a set of interest points are extracted from an image sequence. However, the features extracted for action recognition are spatio-temporal; rather than just trying to find interest points in a single frame, interest points are searched for along the temporal axis of a sequence of images. Laptev and Lindeberg [58] detect temporal corners by extending the Harris corner detector to three dimensions. This detects sudden changes in motion such as a bouncing ball. Applied to gait some stable features exist, such as when the legs change direction, which were shown to be suitable for action recognition [86]. The advantage of this approach is that each action can be represented compactly as a set of spatio-temporal features. Furthermore, if some spatio-temporal features are absent, for example if a subject is wearing baggy clothes, enough features should still exist for recognition.

However, it is suggested by Dollar *et al.* [29] that whilst temporal corners are suitable for recognising actions such as gait or large gestures, e.g. waving, for more subtle actions, e.g. facial movements, temporal corners rarely exist. Their solution is to use a set of 1D Gabor filters along the temporal dimension of an image, interest points can still then be found as local maxima in the resultant signal. The advantage of the Gabor filter is that it will typically also detect periodic motions. This approach is shown to discriminate between different mouse activity such as grooming, eating and sleeping.

Batra *et al.* [6] suggest using a set of space time shapelets, these are three dimensional filters designed to detect events such as a moving edge. The bank of features used is learnt in an unsupervised manner and are shown to be able to detect common actions such as walking and waving. Fathi and Mori [33] use an approach similar to that of Efros *et al.*'s, except rather than using the motion field of the entire image they run AdaBoost to select a subset of the motion image that is the most discriminative. This subset of features is described as mid-level features.

Action recognition is an area where motion is often exploited; however, the level of the motion used varies. In template matching approaches, such as Motion History Images, high-level motion is used that represents a change of structure or shape. However, the field has recently adopted methods using low-level motion, such as the work of Efros *et al.* [30]. Furthermore, searching for areas of interesting local motion have became popular as it is believed these areas will be the most discriminative at classifying different motions. In many senses a sparse set of spatio-temporal features is a similar representation to the MLD. It should be noted that the sparse features of the MLD are not selected as they are temporally interesting, but because they represent the location of a set of key joints. However, it is likely that many of the joints selected to represent human form in a MLD would give rise to interesting spatio-temporal features. The recent success of exploiting low-level motion for action recognition suggests that it is a cue that has potential to be exploited for other tasks, this makes investigating its use for pose estimation of particular interest.

2.5 Pose Estimation

Estimating 3D pose of an articulated object from a sequence of images is a particularly challenging task. This is due to the high dimensionality of the problem; the human body is capable of large and varied motions. A consequence of this is that entire limbs may be occluded from view making inference of their positions very difficult. Occlusion can often be overcame with the use of priors, these priors typically assume that the configuration of the person was known at some point in the past, this knowledge can then be projected to the current frame to make a prediction of where the occluded limb is most likely located. However, the longer a limb is

occluded the larger the uncertainly in the predicted location. This propagation of priors given previous detections represents tracking and is captured by the equation of a Recursive Bayesian Filter [25]

$$\underbrace{p(\theta_t|X_t)}_{posterior} = \underbrace{p(X_t|\theta_t)}_{observational} \underbrace{\int_{\theta_{t-1}} p(\theta_t|\theta_{t-1})p(\theta_{t-1}|X_{t-1})}_{prior} \tag{2.1}$$

Where X_t represents an observation made at time t and θ_t is an estimation of the model configuration. Three components are needed for any pose estimation technique. The first is an observational model $p(X_t|\theta_t)$, this allows the likelihood of an observation to be estimated given a set of model parameters. The second is a predictive dynamic model $p(\theta_t|\theta_{t-1})$, this provides the likelihood of a particular pose given the estimated pose in the previous frame, this term is independent of any observations. The third component is a method to efficiently calculate Equation 2.1 and estimate some useful measure of it, such as the set of model parameters that maximise it. From Equation 2.1 it is clear that the posterior distribution at time t will be propagated though the predictive model to become the prior distribution at time $t + 1$. This is the recursive nature of the filter.

Early attempts to track articulated motion often used gradient descent methods and optical flow to estimate limb dynamics [14, 54]. These methods often assumed high frame rates and therefore expected small motions across consecutive frames. When using lower frame rates their approaches typically became caught in local minima and were unable to overcome occlusions.

Learning a prior model of the expected motion helped to improve the performance of tracking algorithms and could also be used for action recognition [13, 108]. It was found that a Hidden Markov Model (HMM) could be used to switch between different models that well represented the dynamics of the motion at a particular phase of gait, the Kalmen filter was used to approximate linear dynamics [13] and the Extended Kalman filter to model non-linear dynamics [108].

The use of more complex dynamic models greatly improved the performance of

tracking. However, if the dynamic models used to describe the expected motion were insufficient then the target could be lost and recovery was unlikely. This was to be greatly improved with the development of the particle filter or CONDENSATION algorithm [50]. Originally presented for the tracking of active contours, the benefit of this approach was that rather than effectively just trying to keep track of the maxima of Equation 2.1, a set of samples (or particles) were used to approximate the posterior distribution $p(\theta_t|X_t)$, these particles could then be propagated into the next time instance through similar dynamic models as used previously, but now noise could be injected to model uncertainty in the underlying dynamic system. If the object being tracked did not behave as expected it would be hoped a few particles at least would find an area of $p(\theta_t|X_t)$ with a high likelihood. Further to this, a method of selecting a set of new samples from the old set was introduced. This allowed a new set of samples to be generated from regions that had been found to have a high posterior probability. This technique ensured that focus was kept by the particles on regions of the posterior that had a high likelihood.

This technique was shown to be capable of robustly tracking articulated objects [24, 89]. Sidenbladh *et al.* showed that to achieve accurate tracking of articulated motion it was beneficial to have a prior learnt motion model of the action that was being observed [89]. This was as despite the large number of particles (10000), the approach suffered from local maxima and the unyielding size of the search space. Deutscher *et al.* provides a solution to this in the Annealed particle filter [24]. This approach initially smoothes the posterior distribution to allow the particles to be distributed over a wider range and stopped the samples being "peaked" around the first local maxima found. At each iteration the particles are resampled and the degree of smoothing lessened, it is hoped that this method allows the majority of the particles to converge near to the global maxima. This method was shown to track people performing tasks without the need for prior knowledge, for example performing a handstand [23]. An alternative method to improve the performance of a particle filter is to locally maximise each individual particle [93], however, this could result in all particles converging onto the same local maxima. This is often the case that a particle filter would hope to avoid.

Many suggestions have been put forward to improve the performance of the particle filter, these mostly relate to learning better predictive models to improve the target-

ing of the particles across consecutive frames. These include the Rao Blackwellised particle filter that allow the correlation between opposing limbs to be modelled [109] and more commonly approaches where the particles are propagated in a low dimensional pose space [59, 62, 82, 110].

Whilst there are many methods to reduce the representation of data to a low dimensional space, techniques that allow particles to be propagated in this space must have an important property, there must be an inverse mapping from the low dimensional space to the high dimensional pose space. Dimensionality reduction techniques commonly used include Principal Component Analysis (PCA) [110], Locally Linear Coordination (LLC) [62] and Gaussian Processes [82]. PCA can only be used to well represent linear data whereas LLC and Gaussian Processes can be used to represent non-linear data. One of the problems with these approaches is that there is a loss in generality as it is already assumed what motion will be observed. Furthermore, it is not clear how well these approaches generalise to gaits that are notably different from those contained in the original training set.

The success of approaches that first learn the action being performed in a low dimensional space has lead some to suggest that the use of stochastic filters is no longer necessary [31, 102, 103]. These approaches use local search methods, such as gradient descent, to find local minima, which it is suggested are often global. Linear PCA is applied to a sequence of poses in [102] to learn a low-dimensional embedding for pose examples. The key difference between this and other approaches is that each complete gait cycle is represented as a single data point, so the embedded space is representative of gait space rather than pose space.

In [103] a Gaussian Process Dynamic Model is used to embed the high-dimensional training data in a low dimensional space. A high-level motion model is then learnt in the low-dimensional space to ensure that transitions through different phases are smooth. Gaussian process models are popular as they require little training data, often a single gait cycle. In [31] Gaussian Process Latent Variable Models are used to embed both 3D poses and corresponding binary silhouettes. Given an example of a binary silhouette the goal is to find the position in the latent space that maps to a similar silhouette, from this location the corresponding 3D pose can be extracted. This can again be performed via gradient descent in the low-dimensional

space. Whilst these approaches have received much attention it's currently unclear how well they perform in the presence of noisy data and how far from the training examples the models generalise.

A further approach to constrain the search space, so that particle filters can be applied without learning a prior over the action being performed, is to model the physics of gaited motion [15, 16, 17, 106]. This work is largely driven by advances in robotics where understanding of physical models has allowed bipeds to successfully walk upright. The approach presented by Brubaker *et al.* model the equations of motion and collisions as feet touch the ground, the toe off motion is provided by an impulse of the foot against the ground [17]. The approach modeled the legs as straight rigid objects; however, extra joints such as the knees could still be inferred. In [15] the method was extended so the model was now also controlled by applying torque forces to intermediate joints such as the knees, the new method could also be applied to people walking up or down slopes and captured subtle differences in pose, such as slightly leaning backwards when walking downhill, or forwards if walking up. Vondrak *et al.* [106] suggest a more complex 3D model that allows the switching on and off of physics based dynamics, the limitation with this approach however, is that it relies on training examples of the motion being performed. One of the key benefits with these approaches is that any resultant motions look realistic and as knowledge of the ground is known, the feet do not "slip" along the ground plane as is common with many other approaches.

Whilst all of the approaches discussed so far attempt to solve the tracking problem represented by Equation 2.1, there are certain situations where there are no prior observations to be relied on, the most common being the first instance that a person is observed. In this situation the pose in the first frame often has to be manually initialised. This need for manual initialisation is the practical limitation with this approach; these methods all assume initialisation to be a separate problem. However, particularly unrealistic is the assumption that if such an automatic initialiser did exist it would output ground truth quality poses, if this were the case then the use of tracking algorithms would become redundant since the initialiser could simply be used in each frame independently.

A further limitation with the approaches discussed so far is that once the estimated

pose starts to diverge from the correct solution recovery is very unlikely. All trackers will fail eventually and the poorer the initialisation the quicker this will occur.

There have been some notable attempts to overcome this problem. Whilst Loy *et al* rely on the manual labeling of key frames [64], Fossati *et al.* attempt to detect key poses corresponding to when the legs are at their maximum separation [42]. From these initial detections, for which the pose is assumed to be the same as that in the training data, the phase of gait between detections is estimated using a Dynamic Programming solution. From this a crude estimate of pose is provided for each frame which can then be refined using the observations available from each frame. The intention of this approach is that pose estimates near the key detections will be very accurate so in effect at these points the tracking is reinitialised.

Bouchrika and Nixon [12] detect heel strikes identified using the Harris corner detector. The moving object is first segmented and then the Harris corner is then applied to the segmented object, they reason that whilst the foot is stationary on the ground a larger number of corners will be detected than whilst it is moving. These dense regions of detections can then effectively be used as anchor points to fix the position of the feet. The 2D position of the remainder of the joints is then inferred using motion templates.

Mori and Malik [70] go one step further and use a completely exemplar based method. Objects are represented and modeled using shape contexts, which represent the edges of each shape as a set of sampled points. At each point on the shape a histogram is calculated that captures the location of the remaining features. Using the entire set of histograms provides a rich shape descriptor. Given a query image, a similar training exemplar is located for which pose is known, a localised search can then be performed to estimate 2D pose, from which 3D pose is then extracted.

An exemplar based approach is also taken by Ong *et al.* [73]. 3D training data is provided via motion capture, from which 2D images of each person can be synthesized as if filmed from different viewpoints. Similar poses are clustered using the motion capture data and k-means. Motion flows are learnt within each cluster that describe the temporal progression of pose through a particular motion. Visual

matching is performed via the Chamfer distance of extracted edgelets. Tracking is performed by propagating a set of particles in the clustered pose space. The problem with these approaches is that the extracted poses are not unique to the sequence observed. Whilst this is useful for many applications such as action recognition or pose initialisation, often poses that characterize an individuals gait may be required.

2.5.1 Bottom-up Approaches

The approaches discussed in the previous section can be categorized as top-down approaches. Top-down approaches attempt to estimate the entire pose of an object in a single pass. Conversely, bottom-up approaches attempt to break the problem down into smaller subproblems, in the case of pose estimation this consists of attempting to first detect individual limbs independently before then assembling detections into likely configurations. This often makes bottom-up approaches more suitable when there is no prior knowledge of pose (i.e. in the first frame of a sequence). Furthermore, if used in each frame of a sequence, bottom-up approaches do not tend to drift from the correct solution since each frame is searched independently from any previous frames.

The reason that top-down approaches are unsuitable for automatic initialisation is not because they are not descriptive enough, but as a result of the curse of dimensionality [8]. The curse of dimensionality is that the size of the search space grows exponentially with the dimensionality of the model, the simplest 2D stick man represented by 11 parts already has 22 dimensions, assuming an xy coordinate is needed to define each part. If a possible set of just ten locations are considered for each dimension the search space is already of the order of 10^{22}! Consider a typical approach that uses ten thousand particles to approximate the posterior distribution. If initially they are randomly distributed though the search space, the fraction of the space covered would be insignificant, there would be no hope of finding a part of the posterior near a global maximum. Hence top down approaches will only work if first initialised. This also highlights why recovery of particle filters to the correct solution is so unlikely and why so much of the work discussed in the previous section is focused on developing better predictive models, to try and prevent for as long as possible the inevitable tracking failure.

The process of estimating pose given no prior temporal information is often called detection, this detection differs from that described in the context of surveillance in Section 2.2. Here it is assumed that a person is known to be present and detection, in this context, attempts to estimate the pose of the person present.

Common bottom-up approaches used to detect pose attempt to exploit the relatively simple structure of most articulated objects. Approaches such as Dynamic Programming (DP) [35] and Belief Propagation (BP) [43] allow inference to be performed on trees in polynomial time. These approaches allow the entire search space to be explored ensuring that the maxima found is global. The main limitation with these approaches is the simple relations that are allowed between parts. This means that correlations between opposing parts could not be captured. Further difficulties are that it is hard to stop observations being over counted; often the optimal solution may have both arms assigned to the same arm like observation. Felzenszwalb and Huttenlocher [35] overcame this by sampling from the posterior and comparing the result to a set of example poses, this method ensured that two separate arms and legs were detected. Ramanan *et al.* [80] first searched for just the nearside limbs, then excluded the far side limbs from the space occupied by these limbs.

Lan and Huttenlocher [57] introduced a common factor into a simple tree graph to capture relations between opposing parts. The advantage of this approach was that the parts were not directly linked to one another creating large cliques, but connected only through the common factor (a joining node in the graph). This meant the graph had a maximum clique size of 3. A further approach was to move away from tree models and allow the problem to be represented on a trellis type structure [51], this allowed weak edges to be introduced to the graph. These could be used to add constraints between parts so that, for example, opposing limbs couldn't occupy the same space or to enforce that opposing parts should have a similar appearance.

All these approaches try to avoid representing a human as a graph with large cliques and loops; this is as the time to search the space is exponential with clique size. However, a solution to this was provided with loopy belief propagation, attributed to Judea Pearl [74]. This iterative approach could be used on graphs with arbitrary clique sizes and loops, it simply assumed the loops didn't exist. The approach was not guaranteed to find the global minima, however, empirically was found to perform

well. This method was successfully used to model both the relationships between connected and opposing limbs across time [79].

The problem with these approaches was that despite being able to search a large space more efficiently than top-down methods, the space was still quite often too large to allow 3D pose estimation to take place. Several approaches simply performed 2D inference then tried to map 2D observations to 3D [43, 60, 67, 71]. This was achieved by a variety of methods such as comparing the extracted pose to exemplars for which the 3D pose is known [67, 71] or using Markov Chain Monte Carlo methods to fit a 3D model [60]. However, the above examples were applied to only upper body tracking, which is a far more constrained problem.

One of the fundamental problems with bottom-up approaches is the search is usually performed on a grid, the location of this grid is obvious in a 2D image plane, however, in three dimensions a 3D grid must be used. How should this grid be defined and constrained? Since $\mathbf{R}^3$ is an infinite space, inference can't be performed over the entire grid. A method that attempts to overcome this problem and allows inference to be performed using bottom up methods in three dimensions is Non-parametric Belief Propagation [96]. This method, when applied to the problem of pose estimation, rather than searching the entire space, uses a set of samples to approximate the location of each limb [91]. Whilst similar to a particle filter the key difference is that particles are not used to represent the entire human pose state space. In effect a particle filter was designed that could exploit the fact that estimating the pose of an articulated object could be decomposed into smaller sub-problems. Whilst this approach was not guaranteed to converge to global maxima it was shown empirically to achieve good results.

A similar approach is presented by Mitchelson and Hilton where hierarchical sampling is used [68]. Initially a set of particles are generated representing the entire state vector of the object being tracked, these particles are then decomposed into their constituent parts. Following which a hierarchical method is applied to refine the estimate of each part, starting with those that can be accurately located and tracked independently, such as the torso. This method uses multiple cameras and is demonstrated accurately estimating pose for multiple people whilst performing unseen actions.

Using detection in every frame can be described as tracking by detection, the most likely pose is determined in each frame independently. These approaches allow automatic initialisation and prohibit the accumulation of errors since every frame is in a sense, a fresh start. In terms of Equation 2.1 they attempt to estimate $p(X_t|\theta_t)$ for each frame independently.

However, top-down approaches showed that passing information across consecutive frames proved to be beneficial, so there is no reason to suppose the same wouldn't be true of bottom-up approaches. In [79] a model of the appearance of the subjects being tracked is learnt to improve the robustness of the tracking algorithm. Lan and Huttenlocher [56] used a HMM to switch between different priors of pose corresponding to different camera viewpoints and phases of gait. The difference between this and the high-level motion models used in top-down solutions was that estimating the state of the HMM was dependent on the pose extracted by the algorithm in the first place and the prior provided by the HMM was only represented over pose space (i.e. $p(\theta_t)$), information about the pose in the previous frame was not explicitly provided. Sigal *et al.* [92] explicitly pass information between consecutive frames by treating limbs in consecutive frames as connected nodes in a graph. This complex graph is solved using Non-parametric Belief propagation. The advantage of these approaches over top-down methods is that the best solution can be found over an entire sequence using both temporal information and information from spatial searches.

HMMs are also used in a low-dimensional pose space by Andriluka *et al.* [4]. A Gaussian Process Latent Variable Model (GPLVM) is used to model the pose space of a pedestrian walking side on to the camera. Whilst pose is searched for in each frame independently, priors are provided via the GPLVM. The search is similar to that presented by Felzenszwalb and Huttenlocher [35]. However, whilst in [35] appearance models were based on the overlap of rectangular templates with a binary silhouette, in [4] more complex appearance models are learnt for each limb independently. Each limb is represented using a bag of features approach.

Ramanan *et al.* [80] used a similar approach to that of Fossati *et al.* [42]. A pose corresponding to when the legs are at maximum displacement are detected and from these detections the appearance of each of the limbs is learnt, this model of the

appearance can be used to track each person in intermediate frames. This approach only tracked in 2D and required the key pose to be performed by the person being observed, however, it was shown to extract poses from varied activities such as ice skating and playing baseball.

There are several methods that combine the ideas of top-down and bottom-up. Often the assumption is made that some parts can be reliably enough detected such that all other limbs can be searched for conditioned on the known location of the detected parts [43, 71]. The obvious problem with these approaches is that if these key parts can't be detected with great accuracy the approach will fail. Ferrari *et al.* [38] take a less hit or miss approach to 2D pose estimation, they attempt to progressively reduce the search space by first finding possible locations of the torso to fix the scale of the model being used, however, rather than just finding the best torso candidate often more than one will be recovered. If detections are missed pose will be estimated using information from frames where detections were made. This temporal information can also be used to prune any false-positives, in this sense it is similar to the work presented in [80].

2.5.2 Discriminative Methods

All of the methods discussed thus far rely on the combination of priors and observations and can be defined as generative approaches. The distribution of interest is the posterior distribution $p(\theta|X)$ of a particular pose θ given an observation X. The most likely pose is then often the value of θ that maximises $p(\theta|X)$. Generative approaches calculate the posterior distribution $p(\theta|X)$ for each frame through Bayes' theorem

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\sum_{\phi} p(X|\phi)p(\phi)} \tag{2.2}$$

where the denominator of Equation 2.2 is a normalising constant calculated as summation over all possible model configurations.

Whilst generative approaches estimate the posterior in each frame through Bayes' equation, discriminative approaches attempt to simply learn the distribution $p(\theta|X)$ offline before pose estimation takes place [101]. These approaches often require large volumes of training data to be learnt so that a sufficient area of $p(\theta|X)$ is known. Typically a relevant sub area of $p(\theta|X)$ is learnt that an expected motion is likely to occupy. However, a limitation with these approaches is that if an observation is made that is different to those included in the training set the resultant pose is likely to be inaccurate and furthermore it is difficult to understand how these methods will perform in such cases.

An early approach to estimate pose using a discriminative method was made by Rosales and Sclaroff [85]. They attempted to learn a mapping between the Hu moments of a binary silhouette and 2D marker positions. Data clusters of 2D marker positions were learnt and a neural network was trained to perform the mapping from image observations to these clusters. The same neural network could then be used to estimate the 2D pose of an unobserved silhouette.

Often the difficulty in these approaches is in representing training data in a low dimensional form and then structuring it so that new observations can rapidly be queried against training examples. Rogez *et al.* [84] use a set of Histogram of Orientated Gradients (HOG) features to represent the appearance of each person in the training set. The most discriminative subset of features is learnt during training so that the vector used to represent each pose is low dimensional. Given these sets of training data Randomized Trees are used to learn the mapping from an input image to 3D pose. HOG descriptors are also used by Shaknarovich *et al.* along with Parameter Sensitive Hashing [87]. Sminchisescu *et al.* use shape context features to represent a binary silhouette and a Bayesian Mixture of experts for inference [94]. Shape contexts are also used by Agarwal and Triggs; however, mapping is performed via a Relevance Vector Machine [3].

Bissacco *et al.* [9] extract motion maps from consecutive images. Multi-dimensional boosting is then used to learn a mapping between a set of Haar features and 3D pose. The Haar features are applied to both the original image and the extracted motion field. Using motion is shown to improve performance over only using image data. It is particularly well suited for discriminating ambiguities such as differentiating

between nearside and far side limbs.

Whilst these approaches are capable of estimating pose faster than most generative approaches. The problem is that often they are learnt and tested on very similar sequences. Using feature sets such as HOG descriptors, it is not clear how well these would generalise to unseen individuals - it is difficult to know what is being learnt. Furthermore, these approaches are limited to the camera views from which the training data was initially collected. In effect the approach is only as good as the training data provided.

There is also an inevitable limit on how many training examples can be stored and used in terms of memory, learning and indexing. Whereas generative approaches use a highly compact representation that can be manipulated to create an infinite number of possible poses.

Compared to discriminative approaches generative methods are more engineered, a 3D model used to represent a human could be applied to detect and track any motion being performed. However, to achieve competitive quantitative results the flexibility of generative approaches is often compromised to only be applicable to certain motions in a controlled environment. It is this that has allowed discriminative approaches to gain so much ground in recent years. The question of whether generative or discriminative methods are best suited for pose estimation tasks is similar to that of whether recognition of the MLD is simple pattern matching or whether a real understanding of the underlying object creating the observed motion needs to be known. Could humans extract 3D pose of objects if we were never allowed to explore the three dimensional world?

2.6 Summary

In this chapter a review of the current literature has been presented. Experiments using the Moving Light Display have been discussed and whilst they provide much of the motivation for the work carried out in this thesis they provide little insight as to whether 3D pose can be extracted from a set of sparse moving features. When

the lights are in motion we experience a familiarity with what is being observed and an understanding of its cause, however, when the display is paused the task of trying to extract pose becomes more a case of remembering the state of the walker at the point the pause button was pressed, as opposed to trying to find the best configuration for the current static display. It is perhaps these personal experiences with the moving light display that leads to the suggestion that although the work of Thirkettle *et al.* [99] suggests structural cues are perhaps more important than dynamic cues, their work provides more a commentary on the current methods used and conclusions made by the psychophysics community when exploring biological motion. In personal experiences the dynamic nature of MLD's are crucial for their understanding.

Whilst in many of the approaches discussed motion is virtually always used in some form, often to extract binary silhouettes, the interest in this thesis is trying to exploit the patterns of motion as a result of people performing gaited actions, not simply to provide segmentation or extract areas of interest. Currently this use of motion has been exploited mostly in action recognition tasks.

Attention has also been drawn to different types of motion, namely high-level and low-level motion. The key distinction made is that high-level motion refers to a change in the high-level configuration of an object, such as the changes in shape or change in structure, e.g. pose. Low-level motion refers to local motion, the motion of individual pixels, rather than the motion of an object that has first been given some semantic label, e.g. a hand.

In action recognition, data driven approaches using sparse spatio-temporal features were shown to be able to accurately discriminate between different activities. As discussed in Section 2.4 the features used in a MLD are not selected because they're temporally interesting and neither are the features extracted using the KLT feature tracker. However, as suggested in the psychophysics literature, some parts of the gait cycle may be more revealing than others. Whilst this feature will not be directly exploited, it will be expected that models used will exhibit this property, in a sense this confirms the validity of the models learnt.

There are many techniques to estimate pose, however, any technique used to mimic

human performance using the MLD should be self initializing. Furthermore, work carried out in surveillance suggests that motion is a good cue as it reveals a large amount of information and can be extracted in a relatively short time frame, therefore the use of motion should provide considerable efficiency gains compared to appearance based approaches.

One of the main observations from the previous sections is that virtually all pose estimation techniques exploit appearance, often this is the shape of a binary silhouette. A method of estimating pose using only motion cues would be of interest to the field as this remains a cue that is relatively unexplored for pose estimation tasks. An exception to this is the work of Fathi and Mori [32], where motion templates are used for 2D pose estimation. This approach has closer similarities with that of discriminative methods, where a localised search is performed after initialization. Using richer, generative models of motion remains unexplored.

Discriminative and generative methods are currently two conflicting approaches to pose estimation. Whilst motion has partly been explored in discriminative approaches, e.g. Bissacco *et al.* [9], it is yet to be fully explored using generative approaches. Currently a representation of motion that could be exploited in a 3D generative approach does not exist.

Chapter 3

Learning and Modeling Motion

In this chapter, the wealth of extractable information present in a sparse set of moving features is demonstrated. Initially experiments are performed on individual features using only their motion. The purpose of these initial experiments is to examine whether useful information can be obtained from just the motion of a feature independent of its position in the image plane or relative to other features.

This is achieved by calculating the likelihood that a feature is tracking a specific limb based on its motion through the image plane. This likelihood is calculated by making a comparison between the observed motion and prior learnt motion models. It is also found that through the presented representation of motion further information about the phase (or temporal state) of a motion can also be extracted.

These motion models act as detectors that are akin to the part detectors described in Section 2.5, where initially limbs are detected independently of their position in the image and without knowledge of where previous detections have taken place.

Rather than the part detectors described in Section 2.5 that are dependent on appearance, the presented detectors depend only on motion. A feature tracking a foot is recognised as such because its motion is similar to the motion of a foot when performing a specific gaited action.

The main contribution of this chapter is to demonstrate that it is possible to extract high-level information from just the motion of a sparse set of tracked features. The importance of this, that can not be overstated, is that this is achieved without exploiting any information about the structure or location of the features. This is something that is not possible with the traditional MLD experiments described in Section 2.1 where it is only possible to degrade, as apposed to remove, structural cues.

In this chapter a detailed description of the models used to represent motion is presented. It is described both how models are learnt and how observed motions are compared to them. Results are presented that demonstrate the effectiveness of the presented representation. Findings are summarised at the end of the chapter.

3.1 Modeling Motion

As described in Chapter 2 there are many different techniques to model motion and the type of model used is largely dependent on the target application. These techniques can be broadly split into two groups: the first seeks to learn discriminative models of motion so that given an observation the model can tell us the most likely activity being performed. The second approach seeks to use motion to improve tracking. This can be performed by using *a priori* knowledge of a motion or action to estimate the likely configuration of a person over consecutive frames. Alternatively, observed motions and simple dynamic models can be used to predict an update of the model parameters in future frames. The approach used here is to create generative models of motion; this will allow individually observed motions to be compared against each of the models.

The main advantage of generative models is that if used in the correct probabilistic framework they allow you to ask them a variety of questions and give you meaningful answers. For example a question that could be asked is "What is the likelihood of observing motion v given there is a person walking in the scene?" or "What is the likelihood that the observed motion v is that of a foot given that it is caused by a person walking in the image?". Generative models allow a certain flexibility that

other approaches do not, a single model can be used to answer many questions.

3.1.1 Model Representation

The representation of motion used is designed so that observed features can be compared to each of the models independently. The models are designed to exploit the shape of the trajectory made by each feature across the image plane. A separate motion model is learnt for each of the main limbs and will act as limb detectors. Each model is represented by a discrete set of phases, where the observational model for each phase is represented as a Gaussian distribution over the motion expected to be observed in the image plane between consecutive frames. The mean of each Gaussian represents the average motion between consecutive frames and the covariance matrix represents the uncertainty in each motion. A small covariance implies that little variation is expected in the motion and a large covariance implies a large variation. These covariances define how flexible different parts of the model are.

Each model is defined by $\mathbf{\Theta} = \{\mathbf{R}, \mathbf{\Sigma}\}$ where $\mathbf{R} = \{\mathbf{r}_1, .., \mathbf{r}_m\}$ represents the mean of each Gaussian, $\mathbf{\Sigma} = \{\mathbf{\Sigma}_1, .., \mathbf{\Sigma}_m\}$ represent the corresponding set of covariances and m is the number of phases in the model. By placing the Gaussian distributions end to end in temporal order the shape of the motion model can be seen. This is illustrated in Figure 3.1.

In effect this representation models motion as a deformable shape, where each edge of the shape is the motion expected between consecutive frames. Estimating which limb a feature is tracking is then solved as a matching problem between the observed trajectory of the feature and the shape of the motion model. This representation is similar to that of Coughan *et al.* [18] where a set of Gaussian distributions are used to define the edges of a deformable shape.

Whilst representing the motion model as shown in Figure 3.1 is useful for visualisation, the reader should think of the model as being composed by a set of discrete phases, where the observational model for each phase is represented as a Gaussian distribution, this interpretation will be important in order to understand how observations are compared to the motion models.

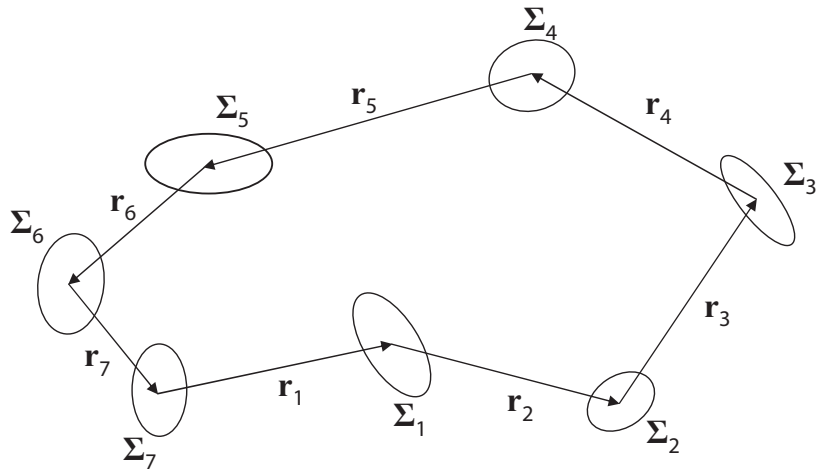


Figure 3.1: *Illustrative diagram of a motion model. Each arrow represents the mean* $\mathbf{r}$ *and each ellipse illustrates the covariance* $\mathbf{\Sigma}$*. The model shown contains 7 phases.*

3.1.2 Model Learning

To learn a model for a specific limb consider a set of n exemplar gait cycles $\{\mathbf{V}^1, .., \mathbf{V}^n\}$ where each gait cycle consists of m temporally ordered vectors $\mathbf{V} = \{\mathbf{v}_1, .., \mathbf{v}_m\}$. Each vector $\mathbf{v}_j$ represents the motion observed in the image plane across the jth frame. The start position of a motion is arbitrary but is assumed to be the same for all exemplars. A model is learnt $\mathbf{\Theta}^{max}$ that maximises:

$$P(\mathbf{V}^1, .., \mathbf{V}^n|\mathbf{\Theta}) = \prod_{i=1}^{n}\prod_{j=1}^{m} p(\mathbf{v}_j^i|\mathbf{\Theta}_j). \tag{3.1}$$

This is a maximisation over all the training examples for every position in the model and is the Maximum Likelihood (ML) estimate of $\mathbf{\Theta}$. If it is assumed that the observed motion at each phase is conditionally independent of any previous or future observations there is effectively a separate set of training examples $\{\mathbf{v}_j^1, .., \mathbf{v}_j^n\}$ for each phase j. Equation 3.1 can then be maximised by solving for each $\mathbf{\Theta}_j$ independently,

$$\mathbf{\Theta}_j^{max} = arg\max_{\mathbf{\Theta_j}} \prod_{i=1}^{n} p(\mathbf{v}_j^i|\mathbf{\Theta}_j). \tag{3.2}$$

This is the ML estimate of Θ_j. Given that $p(\mathbf{v}_j^i|\mathbf{\Theta}_j)$ is modeled as a Gaussian distribution Equation 3.2 is solved by simply estimating the parameters $\{\mathbf{r}_j, \mathbf{\Sigma}_j\}$ directly from the training set $\{\mathbf{v}_j^1, .., \mathbf{v}_j^n\}$ for each phase j. Using this method a separate model is learnt for each of the main limbs.

In this section it has been shown how motion models can be learnt directly from training data provided the training data has two important properties. Firstly, all motion exemplars must start at the same temporal position in the motion being modeled. Secondly, all exemplars must contain the same number of observations i.e. have the same temporal length.

3.1.3 Training Data Acquisition and Preprocessing

Training sequences consist of people being filmed from the sagittal view whilst walking on a treadmill. In each sequence the main joint positions are hand labeled in each frame. It is assumed that opposite limbs will make the same motion so only the nearside of the body is labeled; this means parts that are occluded for large parts of the gait cycle do not have to be labeled. The parts that are labeled are the head, shoulder, elbow, wrist, hip, knee and foot. An example frame showing the scene used and a person with labels overlaid is shown in Figure 3.2.

To learn the models three different people walking on a treadmill for roughly 300 frames each were hand labeled, this is equivalent to about 10 complete gait cycles per person. An example of the data collected for one person is shown in Figures 3.3 and 3.4. The data has been offset on the y-axis so that it does not overlap.

As expected there are much larger motions in the x-axis than the y-axis. It is also possible to see that each joint moves in a cyclic motion. However, for joints such as the shoulder that make a very small motion this feature is not clear. This is because

Figure 3.2: *Example frame from a sequence labeled for training data. The square markers show estimated limb positions.*

the error in manually estimating the position of a joint is of a similar order to the underlying motion, the effect of this is discussed below.

Before motion models can be learnt the data is preprocessed to be in a form as described in Section 3.1.2. Firstly, the data is cut into individual examples of a complete gait cycle. Whilst this process could be performed manually an alternative approach is to find the turning point in the data corresponding to when the foot is at its maximum forward displacement, this point is then used to define the start and end point of each gait cycle. These time slices are shown by the black dashed vertical lines in Figures 3.3 and 3.4. These slices define the start and end of each gait cycle for all parts, this is so that all models are temporally aligned.

Next each individual example is resampled to contain the same number of measurements as the mode of all the examples. This is achieved by using a cubic spline to interpolate between measurements from which new data can be extracted. During this process the start and end positions are fixed. Following these two preprocessing steps the first derivative of the gait cycles is calculated and the motion models can be learnt as described in Section 3.1.2.

The models learnt consist of 32 phases and are shown in Figure 3.5. The dots show the values of the training data for each phase and the ellipses show one standard deviation of the covariance matrices. A feature tracking each model would move

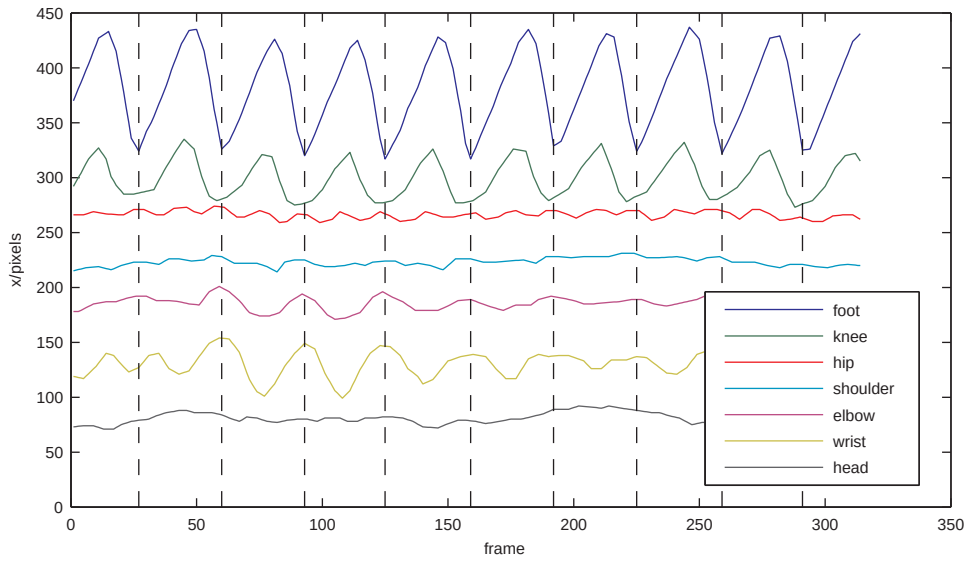


Figure 3.3: *An example of the x-component of hand labeled training data as a function of time. The y-axis position of each part has been offset so that signals do not overlap. The dashed black vertical lines define the start and end of each gait cycle.*

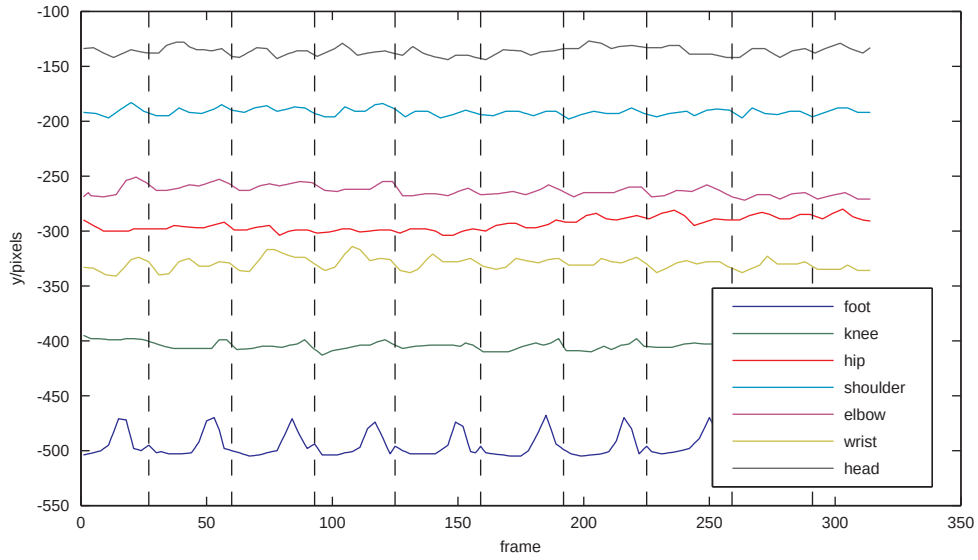


Figure 3.4: *An example of the y-component of hand labeled training data as a function of time. The y-axis position of each part has been offset so that signals do not overlap. The dashed black vertical lines define the start and end of each gait cycle.*

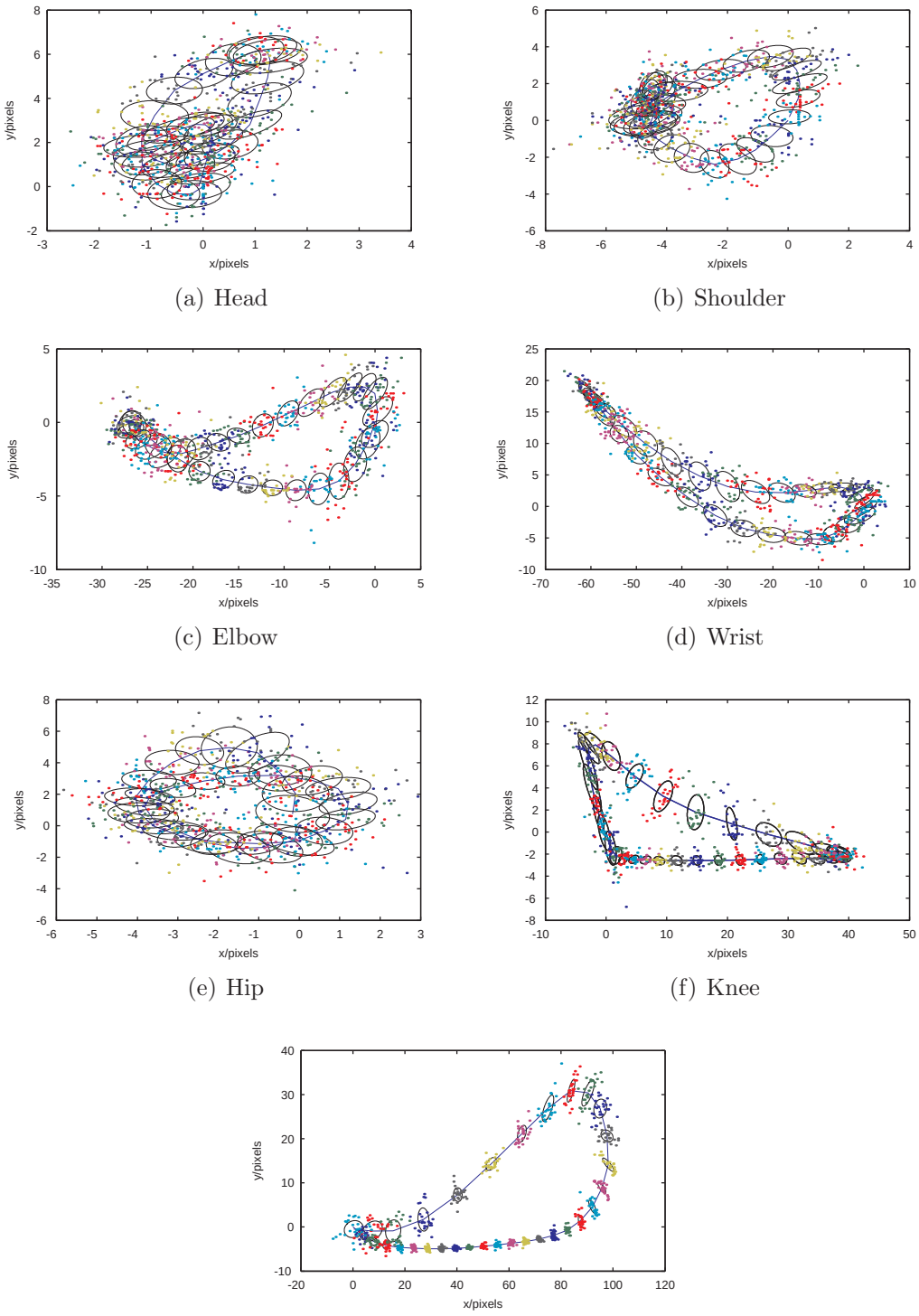
through the phases in an anti-clockwise direction. Whilst difficult to see in Figure 3.5 the models form a roughly closed loop, this is not set as a constraint on the models.

The Gaussian used to represent the expected motion at each phase is expected to model primarily three things. The first is the natural variation in a gait cycle that a single person will exhibit as they walk. The second is the variation expected to be observed from person to person. The third is capturing variation due to representing a motion by a discrete set of phases, in effect each Gaussian models the motion that is expected to be observed at any time in the interval $(j - 0.5)\Delta t < t < (j + 0.5)\Delta t$ where j is a discrete phase, t represents time and is a continuous variable and Δt is the time between frames. However, also being learnt is the noise in the training data due to hand labeling the ground truth. This is particularly clear on models such as the head and shoulder where in some parts of the gait cycle there is very little motion other than noise.

Whilst this noise is undesirable since it represents only how well a person can track a joint and is not characteristic of the motion being performed, it can be speculated that the error of a human tracking a joint may be similar to a feature being tracked by the KLT algorithm. This clearly isn't the case when a KLT point slides along an edge due to the aperture effect, but when tracking a well defined feature the accuracy of the two may be similar. If it is assumed that this is the case then this noise is in fact very useful since it can be used to model the observational noise we would expect from the KLT feature tracker. The similarity in noise between the KLT feature tracker and a human is of course very speculative but is perhaps an interesting point to consider.

3.2 Comparing Observations to Models

In this section a method is presented to compare the observed motion of a single feature tracked over a number of frames to a motion model. The presented approach is set in a probabilistic framework and simple examples are shown to demonstrate the properties and limitations of this method. From an initial approach, modifications are suggested to overcome some of these limitations.



(a) Head

(b) Shoulder

(c) Elbow

(d) Wrist

(e) Hip

(f) Knee

(g) Foot

Figure 3.5: *2D motion models of walking learnt from ground truth data. Ellipses show one standard deviation of the variation expected in this motion. Points show ground truth data.*

Throughout this section most terms will only be defined up to a proportionality. This is because most probabilities will be represented computationally as a logarithm, which are less sensitive to numerical instabilities (i.e. can be used to represent much smaller numbers) and a product of probabilities can be efficiently calculated as a sum of the logarithms. Calculating exact conditionals requires marginalising over variables which is often cumbersome and expensive to calculate. Since the main objective will be to maximise the posterior distribution this will have no adverse effect, since the maximum of a distribution will be the same regardless of how it has been scaled.

Consider that a motion $\mathbf{v}$ is observed across two consecutive frames. The probability of observing this motion given the action being observed is in the jth discrete gait phase is given by

$$p(\mathbf{v}|x=j) \propto \frac{1}{|\mathbf{\Sigma}_j|} e^{-\left(\frac{1}{2}(\mathbf{v}-\mathbf{r}_j)^T \mathbf{\Sigma}_j^{-1}(\mathbf{v}-\mathbf{r}_j)\right)} \tag{3.3}$$

where Equation 3.3 is a Gaussian with mean $\mathbf{r}_j$ and covariance $\mathbf{\Sigma}_j$. For brevity in future $p(\mathbf{v}|x=j)$ will be referred to as $p(\mathbf{v}|x)$. If the motion of a feature is observed over t frames $\mathbf{V}_t = \{\mathbf{v}_t, \mathbf{v}_{t-1}, .., \mathbf{v}_1\}$ the probability of observing these motions given the sequence of phases $\mathbf{X}_t = \{x_t, x_{t-1}, .., x_1\}$ is calculated as

$$p(\mathbf{V}_t|\mathbf{X}_t) \propto \prod_{i=1}^{t} p(\mathbf{v}_i|x_i) \tag{3.4}$$

Where the assumption is made that both the order of phases and the sequence of observations are conditionally independent.

If a feature is being tracked over a number of frames, each time a new measurement is observed it is undesirable to have to evaluate the whole product in Equation 3.4. To calculate this more efficiently Equation 3.4 can be rewritten as

$$p(\mathbf{V}_t|\mathbf{X}_t) \propto p(\mathbf{v}_t|x_t) \prod_{i=1}^{t-1} p(\mathbf{v}_i|x_i) \tag{3.5}$$

from this it is clear that Equation 3.4 can be calculated iteratively as

$$p(\mathbf{V}_t|\mathbf{X}_t) \propto p(\mathbf{v}_t|x_t) p(\mathbf{V}_{t-1}|\mathbf{X_{t-1}}) \tag{3.6}$$

To constrain Equation 3.6 it can be assumed that in each consecutive frame the action being observed is only able to move into the next consecutive phase in the model, so that

$$x_t = \begin{cases} x_{t-1} + 1 & \text{if } \ x_{t-1} \neq m \\ 1 & \text{otherwise} \end{cases} \tag{3.7}$$

where m is the number of phases in the model. The lower condition is used to enforce the cyclic nature of gait. Once the last phase is reached the motion will start at the beginning again.

The presented approach can be seen as being similar to a first order Hidden Markov Model (HMM), where phases represent the hidden states, except that the simplification has been made that only the next consecutive state can be moved into rather than allowing complex state transitions.

This assumption makes Equation 3.4 more efficient to evaluate since the sequence of all previous phases is implicit in the current phase. For example to maximise Equation 3.4 the complexity of the presented approach is $\mathcal{O}(mt)$ whereas a HMM is $\mathcal{O}(m^2t)$, where m is the number of phases and t the number of frames a feature has been tracked over.

The constraint imposed by Equation 3.7 can be interpreted as simply modeling the conditional dependence $p(x_t|x_{t-1})$ of the phases between consecutive frames where

$$p(x_t|x_{t-1}) = \begin{cases} 1 & \begin{cases} \text{if } x_t = x_{t-1} + 1 \\ \text{or } x_t = 1 \text{ and } x_{t-1} = m \end{cases} \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

Including this extra term Equation 3.6 can then be written in a similar form as a first order HMM or more generally as a Recursive Bayesian filter [25]

$$p(\mathbf{V}_t|\mathbf{X}_t) \propto p(\mathbf{v}_t|x_t)p(x_t|x_{t-1})p(\mathbf{V}_{t-1}|\mathbf{X_{t-1}}) \tag{3.9}$$

The only difference between Equations 3.6 and 3.9 is that in the latter the conditional dependence between phases is modeled. As the motion models presented in this section act as low-level part detectors it is preferred that they are simple and therefore computationally cheap. The cost of constraining only consecutive phase transitions can occur is that the resultant model is slightly more 'rigid' than if using a HMM.

3.2.1 Experiments using Exemplar Motions

In this section the presented approach is demonstrated on exemplar motions and it is shown how the probability density function (pdf) changes as new observations become available. The purpose of this section is not to illustrate the robustness of the presented technique, but rather how it behaves given unseen observations.

For testing, an exemplar motion is generated from the model of a foot (Figure 3.5 (g)). Whilst the length and starting phase of the motion is arbitrary, it is chosen to

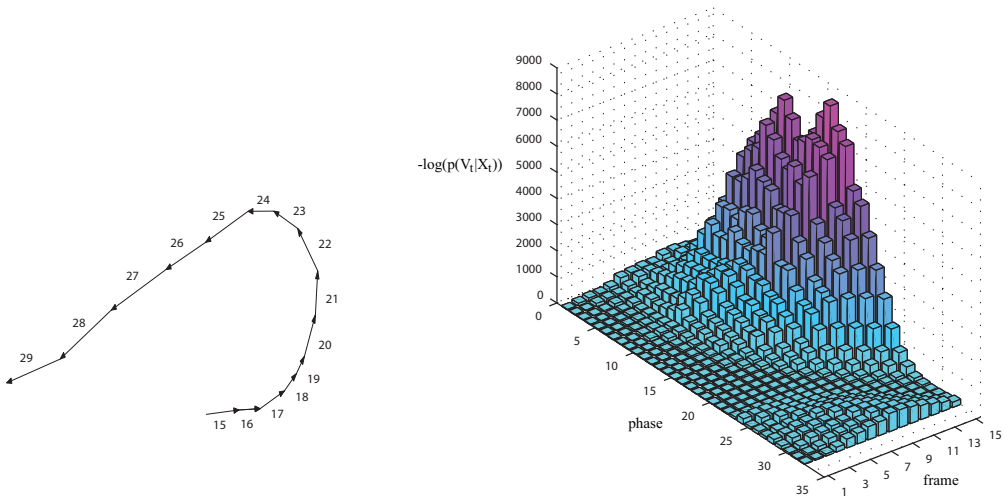
start at the 15th phase for a length of 15 frames. To each observed motion a small amount of noise drawn randomly from a Gaussian distribution ($\sigma = 1.0$) is added. The generated motion is shown in Figure 3.6 (a).

The pdf as a function of time given that the generated motion is observed is shown in Figure 3.6 (b). This has been created by comparing the generated trajectory to the motion model of the foot and assumes that in the first frame only the first part of the motion has been observed, then at each subsequent frame a new part of the motion is observed. This figure shows how the probability of being in each phase changes as new observations become available. The probability is shown as the negative log probability; a low negative log probability represents a high probability. Figure 3.6 shows that initially the first couple of frames of information, corresponding to when the foot is still on the ground, are not particularly discriminative. However, as observations from more distinctive parts of the gait cycle become available the most probable phases become clear as a valley in the pdf.

This effect is also because as new observations become available a wider temporal window, hence more information, is being integrated. For comparison, in Figure 3.7 the probability density function is shown if only the current observations available in each frame are used (i.e. calculating Equation 3.3 for each frame). There are large parts of the pdf with a similar probability making the model less discriminative. To further illustrate this point at each frame the phase with the minimum negative log probability corresponding to the phase that the observed motion was most likely generated by can be found. In Table 3.1 the estimated phase using a single frame of observations compared to multiple-frames is shown. Whilst the model that uses multiple frames incorrectly estimates the phase twice, the single frame model estimates eight phases incorrectly. This illustrates the need to integrate observations over a number of frames.

Frame	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Ground Truth	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Multi Frame	18	13	17	18	19	20	21	22	23	24	25	26	27	28	29
Single Frame	18	9	17	18	22	20	20	22	24	24	26	26	28	27	29

Table 3.1: *Gait phase estimation using observations integrated over multiple frames (middle) and single frames (bottom). Values coloured red are incorrect.*



(a) (b)

Figure 3.6: *The probability density function as a function of time of being in a particular phase given an observed motion trajectory. (a) Shows a hypothetical motion trajectory created from the foot model shown in Figure 3.5 (g), noise has been added from a Gaussian ($\sigma = 1$). The labels show the phase of gait each trajectory was created from. (b) shows the probability density function of observing the given motion. The axis labeled 'frame' describes how much of the trajectory has been observed.*

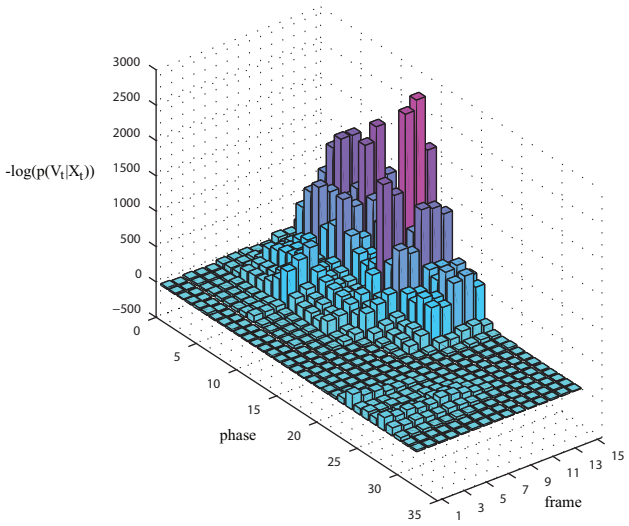


Figure 3.7: *The probability density function using only a single frame of data. Notice that in comparison to Figure 3.6, which uses observations collected over several frames, the current phase in each frame is less clear. This is especially noticeable when more frames of data have been observed.*

A characteristic of Equation 3.9 is that the value of $p(\mathbf{V}_t|\mathbf{X}_t)$ is partly dependent on the number of frames a feature has been tracked over. As a feature is tracked over a larger number of frames the value of $p(\mathbf{V}_t|\mathbf{X}_t)$ will inevitably get smaller as numbers with values less than unity are multiplied. The problem with this is that if comparisons are to be made between the likelihoods of different observed trajectories, only those tracked over the same number of frames can be fairly compared. Otherwise a bias is introduced towards those that have been tracked for a shorter period. To overcome this effect Equation 3.9 can be modified as:

$$p(\mathbf{V}_t|\mathbf{X}_t) \propto \left(p(\mathbf{v}_t|x_t)p(x_t|x_{t-1})p(\mathbf{V}_{t-1}|\mathbf{X_{t-1}})^{t-1}\right)^{\frac{1}{t}} \tag{3.10}$$

Which can be simply rewritten as the average logarithm

$$l(\mathbf{V}_t|\mathbf{X}_t) \propto \frac{1}{t}\left(l(\mathbf{v}_t|x_t) + l(x_t|x_{t-1}) + l(\mathbf{V}_{t-1}|\mathbf{X_{t-1}})(t-1)\right) \tag{3.11}$$

where the notation that $l(A|B) = \log\left(p(A|B)\right)$ is used. To illustrate the advantage of this approach the negative log probability calculated using both Equation 3.9 and Equation 3.10 is shown for the exemplar motion in Figure 3.8. The probabilities shown are for the sequence of phases shown in the middle row of Table 3.1. A comparison of the values from these two approaches shows that using the average there is no biased towards a trajectory tracked over less frames compared to a trajectory tracked over more frames. This will prove critical when a comparison is made of the motion of features that have been tracked over a different numbers of frames.

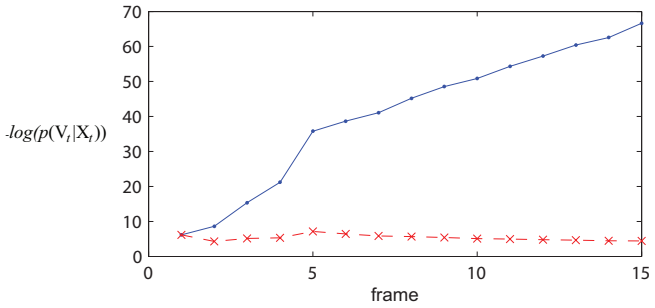


Figure 3.8: *Comparison of calculating $p(\mathbf{V}_t|\mathbf{X}_t)$ using Equation 3.10, shown as red crosses, and Equation 3.9, shown as blue dots. The probabilities are those estimated for the sequence of phases presented in the middle row of Table 3.1.*

The next problem to address is the assumption that only transitions between consecutive phases are allowed over consecutive frames. What will happen if the period of a person walking is different than the period of the model? Again this can be investigated by creating synthetic data from a model and adding a small amount of noise. Two exemplar motions are created from the model of a foot, however, in one example on the 6th frame of the motion a phase is skipped and in the other example the motion stays in the same phase for an extra frame. The purpose of this is to simulate someone walking faster or slower than the model respectively. As before the most likely phase for each frame can be estimated.

First considered is the exemplar motion where a phase is skipped. The ground truth for the sequence of phases the exemplar motion was generated from is shown in Table 3.2. Also shown is the estimated phase using the current method. As can be seen a phase is skipped in the 6th frame, however, it is not until the 15th frame that the presented approach estimates the correct phase again; the model and exemplar motion effectively drift out of phase. Clearly the longer a motion has been observed for the longer it will take the model to correct itself.

Frame	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Ground Truth	15	16	17	18	19	21	22	23	24	25	26	27	28	29	30
Estimated	18	13	17	18	19	20	21	22	23	24	25	26	27	28	30

Table 3.2: *Gait phase estimation given a synthetic trajectory that skips a phase. Incorrect values are shown in red.*

From previous exemplars it can be seen that the correct phase is estimated after a few frames, this suggests there may be very little benefit in integrating over many frames. Taking this into consideration Equation 3.11 can be modified so that the number of frames integrated over can be controlled

$$l(\mathbf{V}_t|\mathbf{X}_t) \propto \frac{1}{\lambda}\left(l(\mathbf{v}_t|x_t) + l(x_t|x_{t-1}) + l(\mathbf{V}_{t-1}|\mathbf{X_{t-1}})(\lambda - 1)\right) \tag{3.12}$$

where λ acts as a decay constant and $l(\mathbf{V}_t|\mathbf{X}_t)$ is now effectively calculated as a weighted mean, where new observations are weighted higher than old observations. It's also interesting to note that Equation 3.12 shares the same form as a low-pass filter. New results using this approach are shown in Table 3.3.

Frame	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Ground Truth	15	16	17	18	19	21	22	23	24	25	26	27	28	29	30
Estimated	18	13	17	18	19	20	21	22	24	25	26	27	28	29	30

Table 3.3: *Gait phase estimation given a synthetic trajectory that skips a phase.*

Now the phase is corrected in the 9th frame as opposed to the 15th. These results were obtained using $\lambda = 3.0$. A comparison of the two approaches on a trajectory where a phase is remained in for two consecutive frames are shown in Table 3.4. Here

the original approach corrected in the 12th frame compared to the new method that corrected in the 7th frame.

Frame	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Ground Truth	15	16	17	18	19	19	20	21	22	23	24	25	26	27	28
Original Method	18	13	17	18	19	20	21	22	23	24	25	25	26	27	28
Proposed Method	18	13	17	18	19	20	20	21	22	23	24	25	26	27	28

Table 3.4: *Gait phase estimation given a synthetic trajectory that remains in the same a phase for two consecutive frames.*

In reality whilst walking a person will not skip or remain in a phase, the observed gait cycle will in effect be a resampled version of the gait model. Whilst resampling the foot model would have made a more realistic exemplar motion it would be difficult to know how a phase on the resampled motion corresponds to a phase in the model. The purpose of this section was not to quantitatively test our approach but to demonstrate its properties and illustrate how it can deal with problems such as observing gait cycles of different lengths. In further sections the presented approach is tested on real data where such in depth analysis is not possible.

3.3 Extracting Information from Real Observations

In this section initial experiments are conducted to test the robustness of the presented method and investigate how best to exploit it. Experiments are conducted on KLT features extracted from sequences of different people walking on a treadmill. The same scene is used to collect data as depicted in Figure 3.2. This scene is used for initial testing as parameters such as the angle of the person walking relative to the camera and the scale of the walker will be similar to those in the training data. This will establish how well the approach is able to contend with tracking noise and person to person variation.

In this section a method to learn a model of the background is described, during which a model of the tracking noise can also be estimated. Following this, how well the approach can discriminate which limb a feature is most likely to be tracking and

how accurately a tracked feature can be used to estimate the phase of a motion is investigated.

3.3.1 Learning a Model of Background Motion

The obvious approach to learning a model of the background motion would be to simply fit a Gaussian to the observed motions in each frame. An example of this method is shown in Figure 3.9. As this Figure shows, the problem with this approach is that features that are tracking the foreground typically have a large motion causing the mean and covariance of the Gaussian to be incorrectly estimated. To accurately estimate the model parameters of the background motion it is necessary to first segment foreground and background features. The correct parameters can then be learnt from the subset of features considered to be tracking the background.

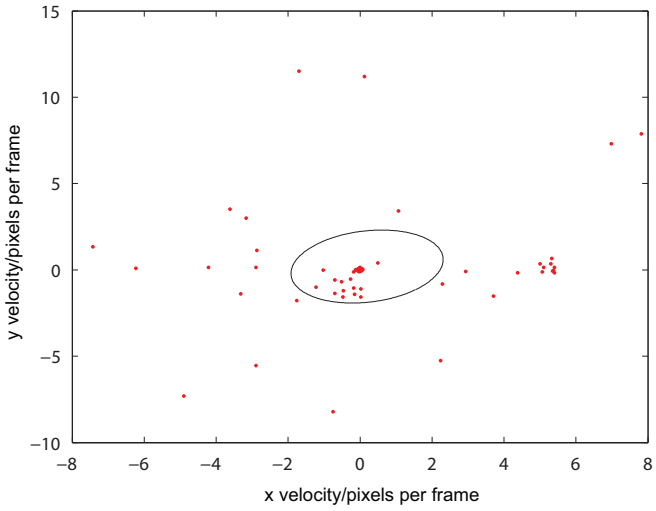


Figure 3.9: *Example of fitting a Gaussian to the observed motions from a single frame. The ellipse shows one standard deviation of the covariance. Each red dot represents the observed motion of a single tracked feature.*

It is assumed that the majority of features track the background and that these features will have a similar motion plus some additional noise. A robust method for estimating a model given observations that include outliers is RANSAC [40] (Random Sample Consensus), this technique has been demonstrated to achieve good results using the motion of tracked features [48].

In the context of this work the algorithm is used to randomly select a feature that has been tracked over consecutive frames. The motion of this feature is treated as a hypothesis for the underlying model, in this case the motion of the background. The motion of all other features are then compared to this hypothesis and are tested to see if their value lies within some tolerance τ of the hypothesis. The number of features that lie within the tolerance is recorded and then another feature is randomly selected as the new hypothesis and the process is repeated again. The algorithm is repeated a predetermined number of times and then the hypothesis with the highest number of inliers is picked as the robustly estimated model. There are several modified versions of the algorithm; however, as RANSAC is not being used to perform the final segmentation of the features, only learn a background model, a modified version is not necessary.

The following equation defines how many times, k, a new hypothesis should be selected to ensure with probability Z that an inlier will be selected as the hypothesis

$$k = \frac{\log(1 - Z)}{\log(1 - w)} \tag{3.13}$$

Where w represents the proportion of points that are inliers. In all the presented experiments k is set to 20, assuming w is a relatively modest 0.5, this ensures that an inlier will be selected with probability > 0.999.

In Figure 3.10 a background/foreground segmentation using RANSAC on the same data as depicted in Figure 3.9 is shown. The background is the dense cluster of red points which appear as a blob due to the compactness of the points. The features tracking the foreground are represented as blue crosses.

Once segmentation has been performed a Gaussian is fitted to the features tracking the background. The centre of the Gaussian represents the motion of the background $\mathbf{v}_{bg}$ and the covariance represents the tracking noise $\mathbf{w}$.

Given these two measurements the probability that a feature is tracking the background given a single observed motion v can be calculated as

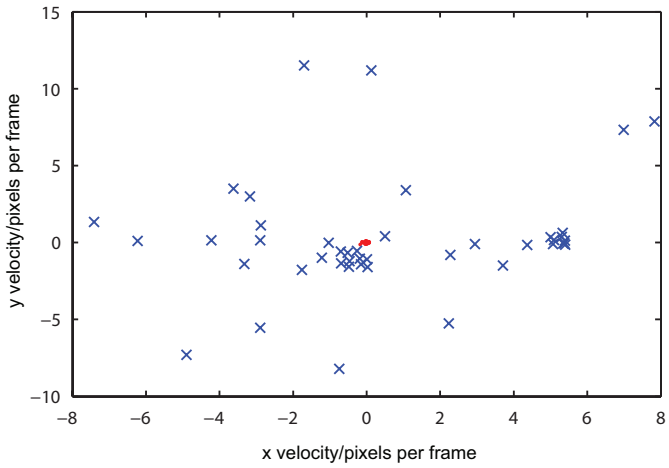


Figure 3.10: *Foreground-background segmentation using RANSAC. The blue crosses show features classed as foreground features and the red cluster of points show features classed as background features.*

$$p(\mathbf{v}) \propto \frac{1}{|\mathbf{w}|} e^{-\left(\frac{1}{2}(\mathbf{v}-\mathbf{v}_{bg})^T \mathbf{w}^{-1}(\mathbf{v}-\mathbf{v}_{bg})\right)} \tag{3.14}$$

Given a sequence of observations $\mathbf{V}_t = \{\mathbf{v}_1, .., \mathbf{v}_t\}$ the likelihood that the feature was tracking the background can be calculated by simplifying Equation 3.12

$$l(\mathbf{V}_t) \propto \frac{1}{\lambda} \left(l(\mathbf{v}_t) + l(\mathbf{V}_{t-1})(\lambda - 1)\right) \tag{3.15}$$

The simplification arises as there are now no underlying states corresponding to different phases of the model.

In this section it has been shown how to robustly estimate a model of the background motion by first segmenting foreground and background features. From this process a model of the tracking noise given by $\mathbf{w}$ has also been estimated which will be used in subsequent sections to add observational noise to the models.

3.4 Experiments

In this section initial experiments are conducted to demonstrate the performance of the presented approach. Initially three aspects are investigated. Firstly, how well the presented approach can distinguish between gaited motion and non-gaited motion. Secondly, how well the presented method can discriminate between the motions of different limbs. Lastly, how well the presented approach can estimate the gait phase of a motion. All of the following experiments are performed on automatically extracted KLT features.

3.4.1 Gait Detection

The aim of this section is to both qualitatively and quantitatively illustrate how well the presented approach can detect gaited motion. In particular it will be shown that this approach does not simply conclude that gaited motion is anything that moves and non-gaited motion is anything that doesn't. This is illustrated by making a comparison to segmentation performed by RANSAC.

Detection is in effect a two choice classification problem. Does an observed object appear more like your target object or more like everything else? What constitutes "everything else" is very hard to define and data driven approaches model everything else by using large quantities of negative examples when learning classification surfaces. The ratio of negative to positive examples used for training is often $\approx 10:1$. It makes sense to use many more negative examples as "everything else" covers a very large space. A more suitable ratio of negative to positive examples would probably be several orders higher than those used; however, this is in practice unrealistic.

In the presented framework the expected appearance of the target object is represented by the motion models learnt for each limb and "everything else" is represented by the background model learnt online. Since the presented approach uses generative models the decision boundary is already well defined.

A class is defined by $\mathcal{C}_k$, where $k = 1$ constitutes the class of gaited motion and $k = 2$

the class of non-gaited motion. Classification is performed by assigning the observed motion $\mathbf{V}_t$ to the class that has the highest posterior distribution $p(\mathcal{C}_k|\mathbf{V}_t)$. Whilst the posterior distribution for the non-gaited class is defined by Equation 3.15, for the gaited class the distribution is less obvious.

Equation 3.12 defines the posterior distribution for a given gait phase assuming only a single motion model. In reality there is a motion model learnt for each limb and the gait phase is not of interest, only of interest is the likelihood of a motion being gaited. The introduction of multiple limbs can be accommodated by introducing an extra parameter into the argument of $p(\mathbf{v}_t|x_t)$ as $x_t = \{j, l\}$, as before j represents the phase of which there are J and now l represents the limb of which there are L. The probability of a motion being biological can then be calculated by marginalising over j and l.

$$p(\mathcal{C}_1|\mathbf{V}_t) = \sum_{l=1}^{L}\sum_{j=1}^{J} p(\mathbf{V}_t|x_t = \{j, l\}, \mathbf{X}_{t-1}) \tag{3.16}$$

where the arguments at time t are shown for clarity. This is just a summation of the probabilities over all limbs and all gait phases at time t.

When the likelihood is calculated for an observed motion tracked over a single frame $p(\mathbf{v}|x = \{j, l\})$ given by Equation 3.3 it is important that observational noise is included in the calculation. This is defined as the parameter $\mathbf{w}$ and is the same as the covariance of the background motion, which was introduced in Section 3.3.1. The model for an observed motion $\mathbf{v}$ is then

$$\mathbf{v} = \mathcal{N}(\mathbf{r}_x, \mathbf{\Sigma}_x) + \mathcal{N}(0, \mathbf{w}) \tag{3.17}$$

Where the first term on the right hand side is the uncertainty learnt from hand labeled training data due to inter-gait variation. However, the second term is observational noise which can not be learnt from hand labeled training data. The effect

of including this noise is equivalent to the convolution of two Gaussian distributions. Equation 3.3 can then be calculated as

$$p(\mathbf{v}|x=\{j,l\}) \propto \frac{1}{|\mathbf{\Sigma}_x + \mathbf{w}|} e^{-\left(\frac{1}{2}(\mathbf{v}-\mathbf{r}_x)^T(\mathbf{\Sigma}_x+\mathbf{w})^{-1}(\mathbf{v}-\mathbf{r}_x)\right)} \tag{3.18}$$

The performance of RANSAC is in part determined by the threshold τ this parameter defines the tolerance on deciding whether an observation is an inlier or outlier. If τ is set very large then the corresponding covariance of the background model will also be large since there will be a large variety of motions classed as inliers. Likewise if this parameter is set very small the covariance of the background will also be very small. Whilst observational noise should always be included in a model, here it is particularly important since it will help to alleviate any biases introduced through the choice of τ.

As an illustrative example, Figure 3.11 shows two Gaussian distributions where $\mathcal{C}_1$ represents gaited motion and $\mathcal{C}_2$ represents non-gait. The standard deviation of $\mathcal{C}_2$ is much larger than that of $\mathcal{C}_1$ and the means of both distributions have a similar value. By not including observational noise in this example an observed motion would be classed as gaited even if the motion was the same as the mean of $\mathcal{C}_2$. The result of this would be a high misclassification rate. However, the inclusion of noise through Equation 3.18 prevents this from occurring. A choice of τ is picked that ensures all background features can be included in a model.

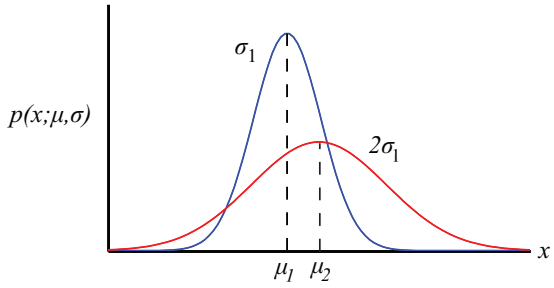


Figure 3.11: *Illustration of two Gaussian distributions used to represent $\mathcal{C}_1$ and $\mathcal{C}_2$ with means and standard deviation μ_1, μ_2 and σ_1, $2\sigma_1$ respectively. Notice that an observation $x_t = \mu_2$ would be classified to class $\mathcal{C}_1$.*

Example frames showing segmentation using RANSAC compared to the proposed gait detector are shown in Figure 3.12. This shows that the proposed method clearly outperforms RANSAC. This is as features such as those tracking the head and shoulders that have little motion are often misclassified as background features by RANSAC.

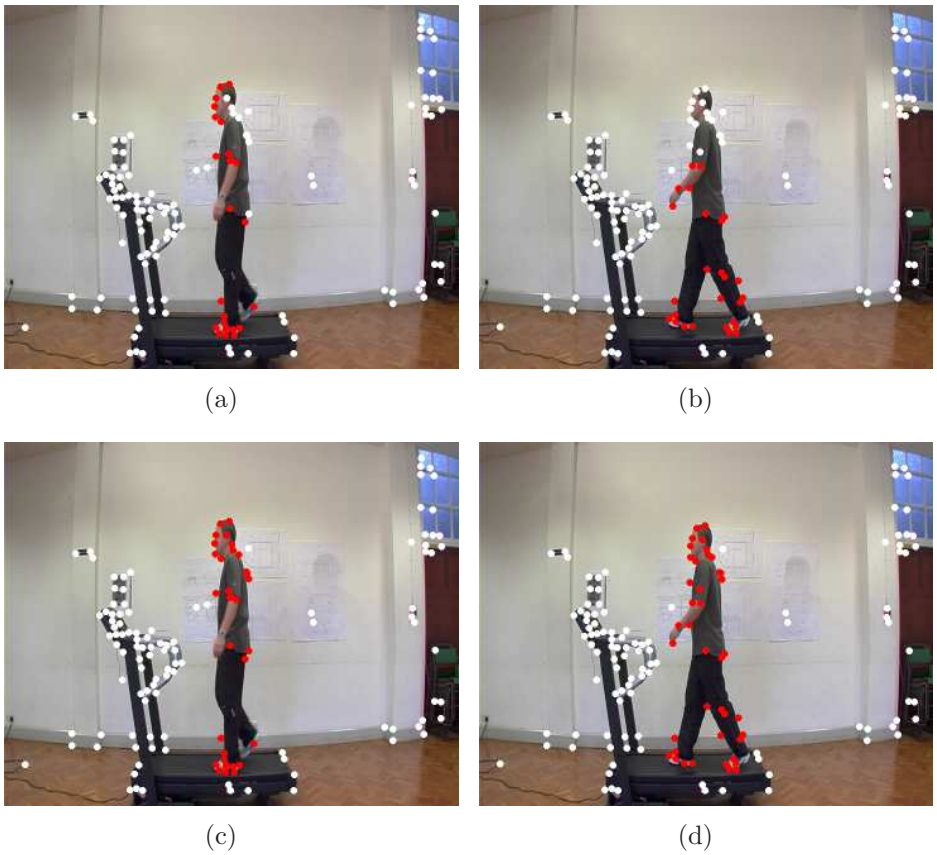


Figure 3.12: *Example frames of different segmentation techniques. a) and b) show exemplar frames using RANSAC to segment foreground/background features using only motion. c) and d) show the same frames using gait detectors. Features coloured red have been classified as foreground features, those that are white background features.*

To quantify the accuracy of the presented gait detector ten different people were filmed at a resolution of 720 x 576 pixels walking on the treadmill shown in Figure 3.2 for approximately 130 frames each. These ten people are different from those used to train the motion models. The KLT algorithm was then used to generate 150 features which were tracked across consecutive frames; any features that were lost were replaced with new features in each frame. For each sequence the features were

hand labeled as either foreground or background to form a ground truth set.

As a baseline RANSAC is used. This will allow a comparison to be made between the performance of the proposed method and a segmentation performed by RANSAC. RANSAC will be performing a segmentation based purely on whether a feature is moving or not. It is simply assumed here that if a feature is moving it's gait and if it's not moving it's non-gait. The purpose of this is to demonstrate that the proposed approach does not simply reach the same conclusion.

The results of the segmentation via RANSAC shown in Figure 3.12 (a) and (b) could be improved via fine tuning of τ. As discussed previously the performance of the proposed method is also dependent on τ since an initial segmentation via RANSAC is used to learn a model of the background and observational noise. If τ is set too small the background model learnt will have a very small covariance and it would be expected that the majority of features would be classed as foreground features. This would result in a high true-positive rate but low true-negative rate. Conversely, if τ is set too large, the covariance would be large and most features would be classed as background features resulting in high true-negative rates but low true-positive rates. What would be expected is that the proposed gait detector's probabilistic approach would perform better than RANSAC over a variety of different values of τ.

A comparison of gait detection using RANSAC and the proposed gait detector are shown in Figure 3.13. These values were found by first hand labelling each tracked KLT feature as gait or non-gait in every frame in every sequence to use as a groundtruth, this corresponded to hand labelling approximately 150000 features. In Figure 3.13 the True-positive rate and the True-negative rate is shown for RANSAC and the proposed method. As expected when τ is small there is a high true-positive rate and low true-negative rate. When τ is large the converse is true. However, in general the gait detector performs better. Particularly noticeable is that as τ gets larger, whilst RANSAC's True-positive rate rapidly deteriorates the proposed gait detector is less affected implying the proposed gait detector is less sensitive to the value of τ particularly at larger values of τ.

To further illustrate this, the accuracy of RANSAC and the Gait detector are shown

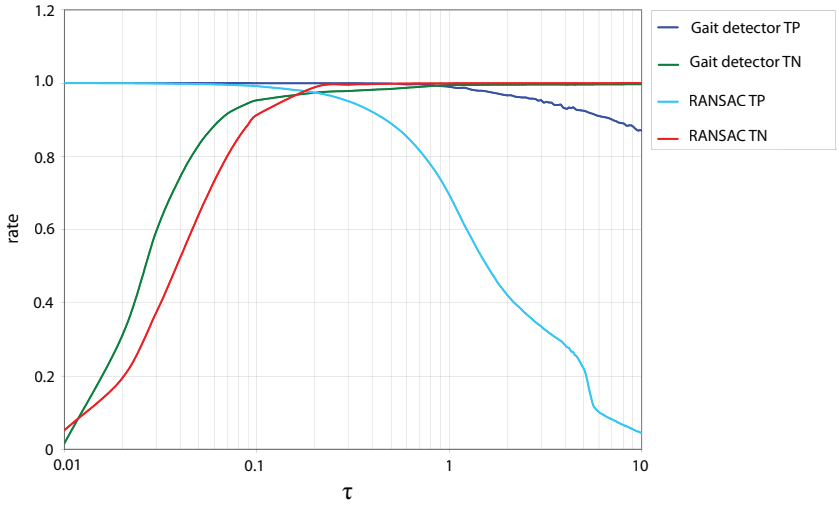


Figure 3.13: *Gait detector performance using RANSAC and proposed gait detector. TP represents True-positive rate. TN represents True-negative rate.*

in Figure 3.14 as a function of τ. The accuracy as a measure of performance is often biased if there are not equal positive examples and negative examples in the test set. To overcome this bias each measurement is weighted by the inverse of the total number of examples in that class. This is equivalent to the average of the true-positive and true-negative rates.

Figure 3.14 shows that the gait detector has better performance over just RANSAC and is less affected by changes in τ. When $\tau = 0.2$ the accuracy of the two methods are very similar. However, this level of accuracy is achieved over a small range of τ by RANSAC.

In this subsection the accuracy of the proposed method has been quantitatively evaluated and it has been demonstrated that the presented approach does more than simply classifying a feature as gait if it's moving or non-gait if it's not. Furthermore the relatively flat accuracy curve between $\tau = 0.1$ and 10 shown in Figure 3.14 implies that the proposed method does not require careful parameter 'tweaking' to ensure robust performance.

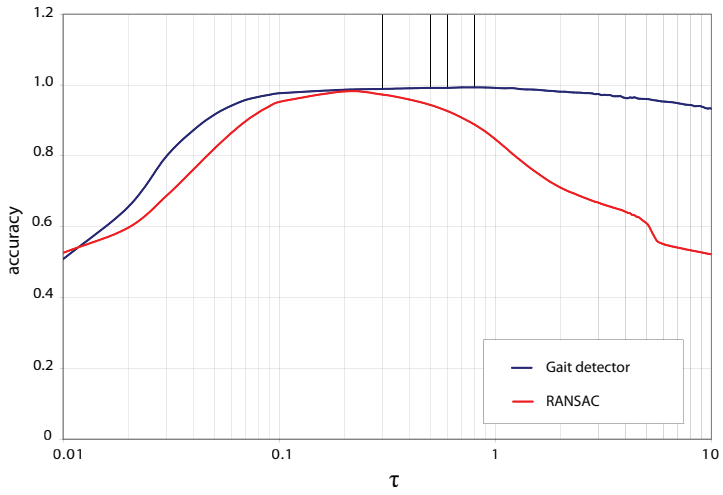


Figure 3.14: *Gait detector accuracy using RANSAC and proposed gait detector.*

3.4.2 Limb Classification

In this section quantitative results are presented that describe how well the proposed method can discriminate between the motions of different limbs. The process of classifying an observed motion as being that of a particular limb occurs after it has been classified as gait or non-gait, where non-gait features are eliminated from further use.

Classifying a feature based on its motion can be achieved by maximising Equation 3.12 over $x = \{j, l\}$, where l corresponds to the limb and j the phase. This method extracts both the phase and the limb that a feature is most likely tracking. The maximisation of Equation 3.12 is performed via an exhaustive search of all phases and limbs. A value of $\lambda = 5$ is used for all experiments in this section.

A ground truth is created by hand labelling the limb that each foreground feature is best described as tracking for each frame in all ten sequences. Often this classification is not trivial to perform, for example a feature tracking the mid point between the knee and hip could equally be labeled as either tracking the hip or the knee. Whilst this ground truth will inevitably contain errors it will still provide a good estimation of how well a feature can be classified as tracking a particular limb based only on its motion.

The confusion matrix for limb classification is shown in Figure 3.15. The highlighted diagonal shows the total number of correctly classified features. This shows that for most limbs the highest fraction of features are classified to the correct limb. There are however many misclassifications. The motion of parts such as the knees and elbows may be very similar as is the motion of the hip and the shoulder meaning that motion alone can not always discriminate between the two.

Actual

Predicted	Foot	Knee	Hip	Shoulder	Elbow	Wrist	Head
Foot	10526	363	185	32	170	354	6
Knee	1286	522	329	74	340	657	15
Hip	76	186	3192	2324	626	587	1322
Shoulder	25	51	991	2872	750	273	3683
Elbow	270	337	712	430	1916	932	110
Wrist	4084	1331	774	196	437	2040	73
Head	75	38	861	6249	621	250	10284
Total	16342	2828	7044	12177	4860	5093	15493

Figure 3.15: *Confusion matrix for limb classification. The bottom row shows the total number of features that tracked each part. The limb that was tracked the least was the knee.*

These results are shown more clearly in Figure 3.16 where the confusion matrix shown has been normalised down each column. The diagonal values show the true-positive rate for each limb. It would be expected that just chance would achieve a true-positive rate of about 14%. From Figure 3.16 it is clear that the true-positive rate for all limbs is higher than this. However, for the knee this value is very close to chance. The main reason is that features tracking the knees are particularly poorly tracked due to edge effects and frequent occlusion as one leg moves past the other. As a result, fewer features are tracked on the knee as shown in the bottom row of Figure 3.15 and those that are tracked are likely to contain more noise than those features tracking parts of the body elsewhere.

In Figure 3.16 many features tracking the knee are shown as being classified as tracking the wrist. This is as the motion model for the wrist has large covariances compared to other limbs as different people show more variation in arm movement whilst walking than any other limbs. The consequence of this is that any tracked

features that contain large amounts of noise are likely to be classified as a wrist feature as this model is the most tolerant of noisy observations. This can be seen by the comparatively high values across the predicted wrist row of the confusion matrix shown in Figure 3.16.

A further anomaly in Figure 3.16 is the high number of features tracking the shoulder that are misclassified as tracking the head. There are two contributing factors to this result. The first is that the motion of the shoulder and the head are similar. The second is a result of inaccuracies in hand labeling a ground truth data set for testing. As described previously, during hand labeling the features were manually classified to the limb that they were nearest, for example features tracking the back of the neck, which is frequently tracked, were labeled as shoulder features, when their motion is probably more similar to that of the head.

Actual

Predicted	Foot	Knee	Hip	Shoulder	Elbow	Wrist	Head
Foot	0.64	0.13	0.03	0.00	0.03	0.07	0.00
Knee	0.08	0.18	0.05	0.01	0.07	0.13	0.00
Hip	0.00	0.07	0.45	0.19	0.13	0.12	0.09
Shoulder	0.00	0.02	0.14	0.24	0.15	0.05	0.24
Elbow	0.02	0.12	0.10	0.04	0.39	0.18	0.01
Wrist	0.25	0.47	0.11	0.02	0.09	0.40	0.00
Head	0.00	0.01	0.12	0.51	0.13	0.05	0.66

Figure 3.16: *Normalised confusion matrix for limb classification - chance* ≈ 0.14.

However, this raises a problem, how do we expect a feature to move that is located between two limbs? Furthermore, how would we hope it would be classified? This is a particularly important question as most features will not be located at the exact position of a main limb. The answer is that the motion will most likely have similarities with both limbs. Therefore, it would be expected to have a high likelihood of being classified as either limb. This suggests that using the presented motion models as hard classifiers is not suitable as they are not reliable enough to accurately discriminate which limb a feature is tracking using motion only. Instead they are best used as soft classifiers, where the likelihood that a feature belongs

to a particular limb is calculated rather than just a hard decision about feature classification.

3.4.3 Phase Classification

In this section the presented method's capability of estimating phase is investigated. Phase is estimated for each feature through the same minimization as used in the previous section. Whilst each of the motion models contains J discrete phases, features tracking opposing limbs will be out of phase by π radians. To overcome this, the possible set of phases are forced between 1 and $J/2$. This is achieved by taking the modulus of the phase the feature is classified as and $J/2$, then adding 1.

A ground truth is generated by labelling the frames where the toe is at maximum forward swing, at these frames the gaited action being observed is assumed to be at phase 1, the frame prior to this is assumed to be in phase $J/2$. A straight line is then fitted between each first and consecutive last phase, this line represents gait phase as a function of frame number. The phase in each frame is than defined by the value taken from this function rounded to the nearest integer. This method is used as it is not clear how to determine manually the phase in each frame independently, which makes creating a ground truth very difficult. Whilst this method will only create a ground truth with accuracy of $\approx \pm 1$ phase, there is no obvious alternative. This version of a ground truth will allow the accuracy of the presented method to be explored, but it should be remembered that the ground truth is itself noisy and has a limited accuracy.

The resultant confusion matrix for all features, independent of which limb they were classified to be tracking, is shown in Figure 3.17. This shows a graphical representation of a confusion matrix where lighter parts have a higher occurrence of features classified to that cell. An accuracy of 1.0 would appear as the diagonal being white and all remaining squares being black. Whilst individual values in each of the cells can not be read, this figure shows the majority of features are classified to the correct phase or close to the correct phase. Notice the overlap present in the top right and bottom left corners; this is as gait is cyclic.

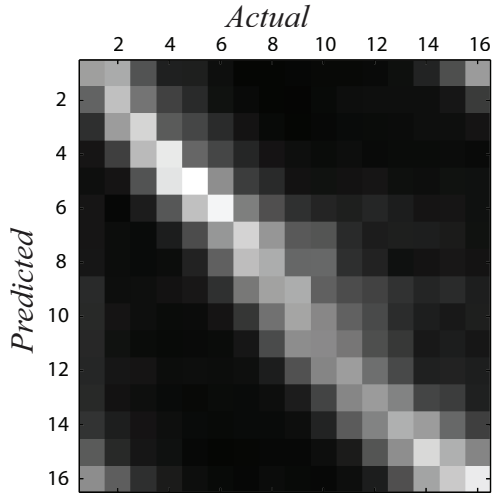


Figure 3.17: *Phase estimation confusion matrix - lighter areas show cells with higher occurrences ($\lambda = 5.0$).*

In Figure 3.18 the accuracy is shown plotted against λ, introduced in Section 3.2.1. This constant effectively defines the size of the temporal window used to integrate a feature's observations over. A value of λ that is too small will mean only observations from a very small number of frames will be used. However, a value that is too large will produce a very rigid model that will easily drift out of phase with the person being observed, if the frequencies of the two are not the same. This can be seen in Figure 3.18 where a too small or too large value of λ results in a drop in performance. The error function goes relatively flat beyond $\lambda = 13$ this is as there will be less features tracked for more than this number of frames, so integrating each feature for a longer period will not effect performance. The best performance is achieved when $\lambda \approx 6.5$. However, a value between 3 and 10 will still provide a similar accuracy showing this method is not too dependent on the exact value of λ.

A maximum accuracy of 0.19 is achieved, this is greater than chance (≈ 0.06). Furthermore, features that are misclassified are frequently classified to a phase in very close proximity to that of the correct phase. This is shown in Figure 3.17 as the gradual change to black as you move to a location further from the diagonal.

In this section it has been shown that the motion of an individual feature can be used to accurately estimate gait phase. These experiments demonstrate that the

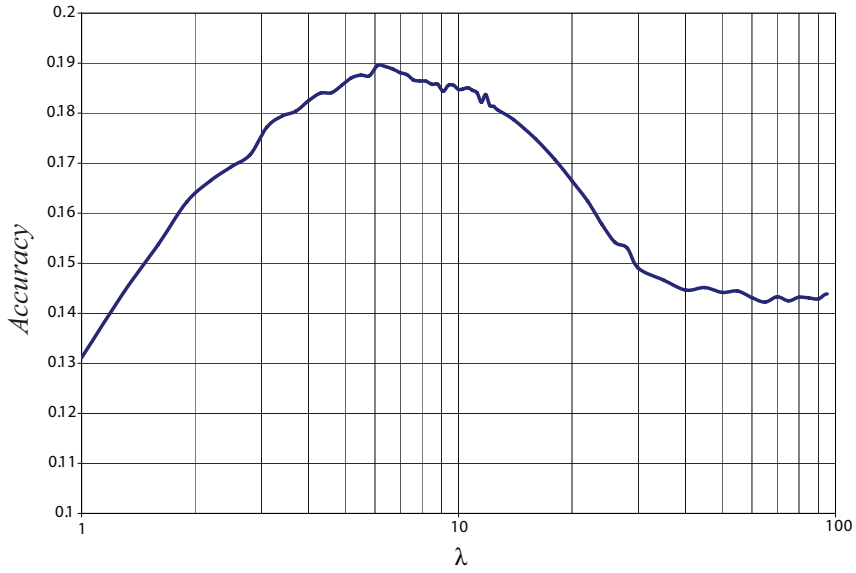


Figure 3.18: *Phase estimation accuracy as a function of* λ.

presented approach is capable of extracting information from the motion of the features and that this information is not masked by noise as a result of tracking errors.

3.4.4 Estimating Global Gait Phase

The ability to estimate gait phase for a single tracked feature has no obvious benefit. However, if some consensus could be reached about the most likely phase of the motion given all the tracked features in each frame, this information would have many uses. For example the gait phase provides a coarse estimate of pose, since a person performing a gaited action will often have a similar pose for each phase.

This consensus can be formed by each observed feature voting for the most likely phase of the motion. The current phase can then be taken as that with the most votes. This approach is illustrated in Figure 3.19 where both the extracted sequence of phases and the ground truth are shown. There is generally good agreement between the ground truth and extracted sequences.

Considering the sequence of the extracted phases there are some unlikely phase

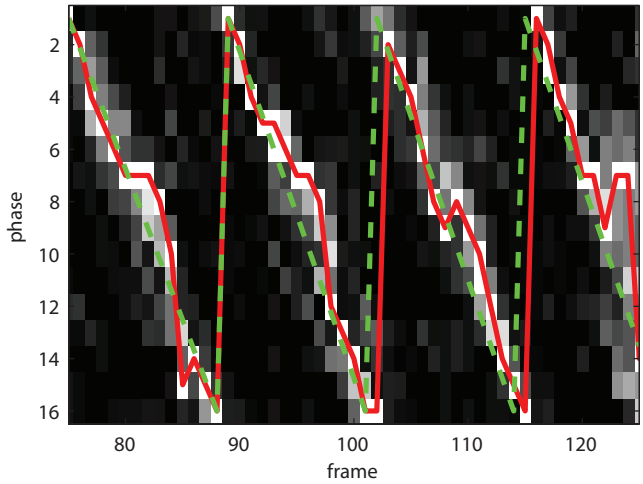


Figure 3.19: *State probability matrix. The lighter a cell the more votes that cell received. Green dashed line shows ground truth. Red solid line shows extracted sequence of phases. The sawtooth appearance is due to the cyclic nature of gait, once the last phase of gait is reached the model returns back to the first phase again.*

transitions, for example between frames 86 and 87 or frames 124 and 125. This is as a negative phase change occurs. It would be expected that at each new frame the estimated phase would increase, not decrease, with the exception of when the model reaches the last phase. The problem with using the phase with the most votes is that this method does not consider the phase of its neighbours. The phase in a given frame is extracted independent to the extracted phase in all other frames.

This problem can be overcome by using a Hidden Markov Model (HMM) to estimate the phase in each frame. A HMM can be used to find the sequence of hidden states, which in this case represent phases, given a set of observations and a prior model. A state transition matrix represents the probability of moving from one state to another across consecutive frames and captures temporal trends in expected observations. For example a state probability matrix could be defined so that only forward state transitions are possible, so that the spurious transitions in Figure 3.19 would not be possible.

To describe a HMM the same notation is used as Rabiner *et al.* [77]. The set of states is defined by $S = \{s_1, s_2, .., s_N\}$ these correspond to phases in our model. Given a set of observations $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, .., \mathcal{O}_T\}$ a HMM can be used to estimate

the most likely sequence of states $Q = \{q_1, q_2, .., q_T\}$ by maximising the probability $P(Q, \mathcal{O}|M)$, where M defines the parameters of the HMM.

The model parameters consist of the transition matrix A, where the entry a_{ij} represents the probability of a transition occurring between the ith and jth state across consecutive frames, $P(q_t = s_i, q_{t+1} = s_j)$. A function B that defines the likelihood of the current observation $\mathcal{O}_t$ given the model is in the ith state $P(\mathcal{O}_t|q_t = S_i)$. A prior model π that defines the likelihood of the model being in the ith state in the first frame $P(q_1 = S_i)$.

Finding the optimal sequence of states can then be found via the Viterbi algorithm. A naive attempt to maximise $P(Q, \mathcal{O}|M)$ results in an algorithm that is NP-complete since every possible combination of the states must be considered. However, the Viterbi algorithm exploits the assumption that only the states at directly neighboring positions in the HMM $\{q_t, q_{t+1}\}$ are conditionally dependent. This assumption represents the Markovian property of a HMM.

The Viterbi algorithm is computed by first calculating at each time step and for each state

$$\delta_t(i) = \max_{q_1, q_2, .., q_{t-1}} P(q_1, q_2, .., q_t = i, \mathcal{O}_1, \mathcal{O}_2, ..\mathcal{O}_t)|\lambda) \tag{3.19}$$

where $\delta_t(i)$ is an intermediate function which represents the best path to the ith state at the current time. This can be calculated recursively as

$$\delta_{t+1}(j) = \max_i \delta_t(i) a_{ij} b_j(\mathcal{O}_{t+1}) \tag{3.20}$$

where $b_j(\mathcal{O}_{t+1}) = P(\mathcal{O}_t|q_t = S_j)$ and Equation 3.20 again highlights the Markovian property of a HMM , as $\delta_{t+1}(j)$ can be calculated as a function of only the current observations and the previous value of $\delta_t(i)$.

At each frame and for every state $\delta_{t+1}(j)$ is computed from the available observations. At each time frame, the index i that maximised $\delta_{t+1}(j)$ is recorded. When the last frame is reached the index is found that maximises $\delta_T(j)$ which corresponds to the optimal state $q_T^* = s_j$. After which, the optimal sequence can be traced back through the set of maximum indices recorded at each frame. $\delta_t(i)$ is initialised in the first frame through the prior distribution as $\delta_1(i) = \pi_i b_i(\mathcal{O}_1)$.

There are several methods to estimate the parameters of a HMM, the most popular being the Baum-Welch algorithm which is an Expectation Maximisation (EM) approach. This approach is iterative and is guaranteed to converge to a local maxima. However, the Baum-Welch algorithm is unsuitable for training a HMM for the presented technique largely because the presented approach uses very little training data, just three sequences. For example the prior function π would have only three entries, therefore the model would assume any new sequences would also begin in any of these three phases. A solution would be to train the HMM using individual complete gait cycles as used to learn the motion models, however, then a method would be required to estimating the start and end of each gait cycle in an unseen sequence, it is this problem that the HMM is partly required for in the first instance.

Instead the HMM used for this particular problem is manually engineered. As there is no expectation that a sequence should begin in a particular phase a flat prior π is used, so each phase is equally as likely. The transition matrix is defined so that from a particular state only three types of transition can occur: To remain in the current state, move into the next consecutive state or skip a state, the exception being when the last state is reached the state can move back to the beginning again. The same transition probabilities are used for every state. This makes the model quite rigid but given the model has N phases a minimum gait length of $N/2$ frames can be achieved if a state is skipped in every frame. There is no upper limit on the maximum length of gait cycle since the model could remain in the same state indefinitely. This transition matrix is shown below in Equation 3.21.

$$A = \begin{pmatrix} a_{ii} & a_{ii+1} & a_{ii+2} & ... & 0 & 0 & 0 \\ 0 & a_{ii} & a_{ii+1} & ... & 0 & 0 & 0 \\ 0 & 0 & a_{ii} & ... & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & ... & a_{ii} & a_{ii+1} & a_{ii+2} \\ a_{ii+2} & 0 & 0 & ... & 0 & a_{ii} & a_{ii+1} \\ a_{ii+1} & a_{ii+2} & 0 & ... & 0 & 0 & a_{ii} \end{pmatrix} \tag{3.21}$$

The values a_{ii}, a_{ii+1} and a_{ii+2} are set as 0.1, 0.8 and 0.1 respectively. So it is most probable that the next phase will be moved into in the next frame, but equally probable that a phase will be remained in over consecutive frames or be skipped. All other transitions have a probability of zero. Note that the bottom two rows of the transition matrix allow phases to loop back to the first state again.

The observation function B could just be defined as the number of votes a phase gets normalised by the total number of votes for each frame. The obvious problem with this is that any phases with zero votes will have a zero probability. This will reduce the set of possible state sequences in the HMM. Consider for example a frame that has just one feature which votes for the jth phase. As this is the only phase with positive likelihood all sequences must pass through this phase to avoid having a zero likelihood. This is overcome by defining a likelihood function as

$$b_i(\mathcal{O}_t) = e^{-\left(\frac{1-N_i/N_{tot}}{\sigma}\right)^2} \tag{3.22}$$

where N_i is the number of votes for the ith phase and N_{tot} is the total number of votes in the current frame. The constant σ defines how narrow the function is. Notice that even if a phase has zero votes it will still have a positive likelihood.

In Figure 3.20 the same state probability matrix is shown as in Figure 3.19 but with

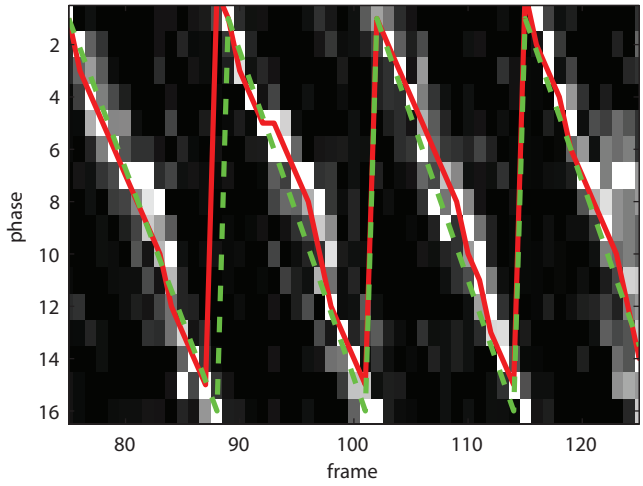


Figure 3.20: *State probability matrix. The lighter a cell the more votes that cell received. Green dashed line shows ground truth. Red solid line shows extracted sequence of phases using HMM ($\sigma = 0.03$).*

the optimal extracted path calculated using a HMM. Notice only forward transitions now occur and the extracted sequence better matches the ground truth sequence.

The constant σ present in Equation 3.22 effectively defines whether the model, represented by A, or the observations, represented by B dominates. So in effect, the exact values of a_{ii}, a_{ii+1} and a_{ii+2} are not important, as the parameter σ will be the deciding factor in whether they have a significant effect on calculating the optimal sequence of phases. In Figure 3.21 the accuracy of the HMM at estimating phase compared to the groundtruth of all ten sequences are shown. This accuracy is calculated as the average difference between the ground truth and the extracted sequence of phases (the red and green dashed line in Figure 3.20).

A small value of sigma will mean that the observations will dominate and a large value the model. As can be seen a large value of sigma results in a larger error than a smaller value, this indicates that for these sequences the observations are more reliable than the model. However, the best average error is achieved when $\sigma \approx 0.3$, where both the model and observations are exploited.

Figure 3.22 shows the error for each of the subjects compared against their average gait length. The red squares show the error from a HMM using a high value of σ

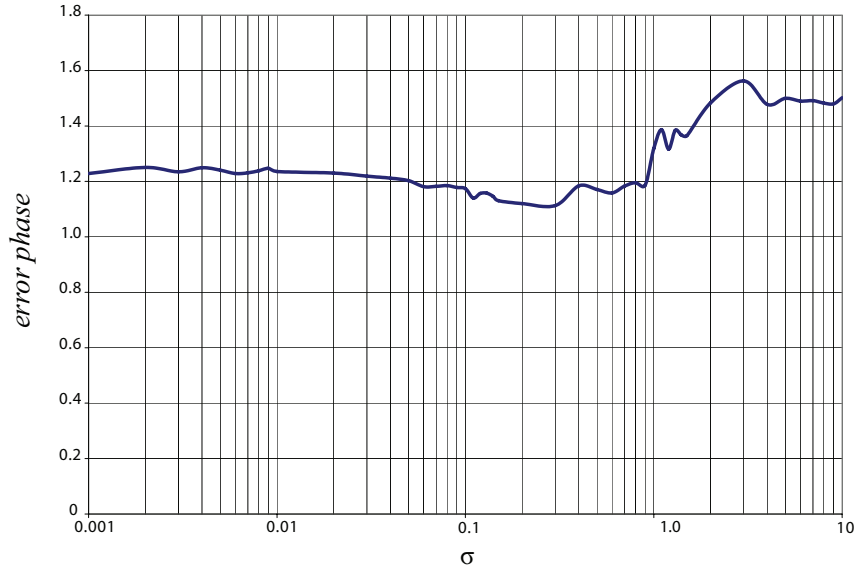


Figure 3.21: *HMM average phase estimation error as a function of σ calculated over ten sequences of people walking on a treadmill.*

so that the model dominates, whereas the blue triangles show the error for a HMM where both model and observations are used. This shows that for subjects that have a similar gait to the model the error is unaffected if the model dominates, since it is accurate for that subject's gait. However, for subjects with a significantly different gait length compared to the model it is important that the model does not dominate since it will drift out of phase with the observations.

The accuracy of the presented phase estimation technique can be demonstrated qualitatively by using a representative pose for each phase and plotting these on each frame of the sequences. The representative pose is the average pose learnt from the same training data used to learn the motion models. Sample frames are shown in Figure 3.23, notice the close agreement between the representative poses and the actual pose of the subject in each frame. This represents a very coarse estimate of the subject's pose, notice also this has been achieved using only the motion of the features, no information about the features' positions have been exploited.

In this section it has been shown that a HMM can be used to estimate the global gait phase of a subject being observed. Using this information a coarse estimate of pose can be extracted by learning a representative pose for each phase of gait. All of this has been achieved without exploiting the structure of the features, only a

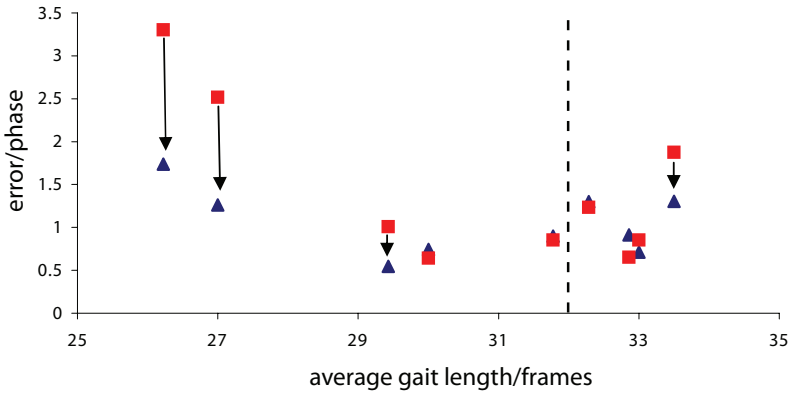


Figure 3.22: *HMM Phase estimation errors for each subject used for testing compared against their average gait cycle length. The red squares show the error using $\sigma = 1.20$ and the blue triangles using $\sigma = 0.03$. The arrows show the change in each error. The dashed straight line shows how many phases were contained in the motion models.*

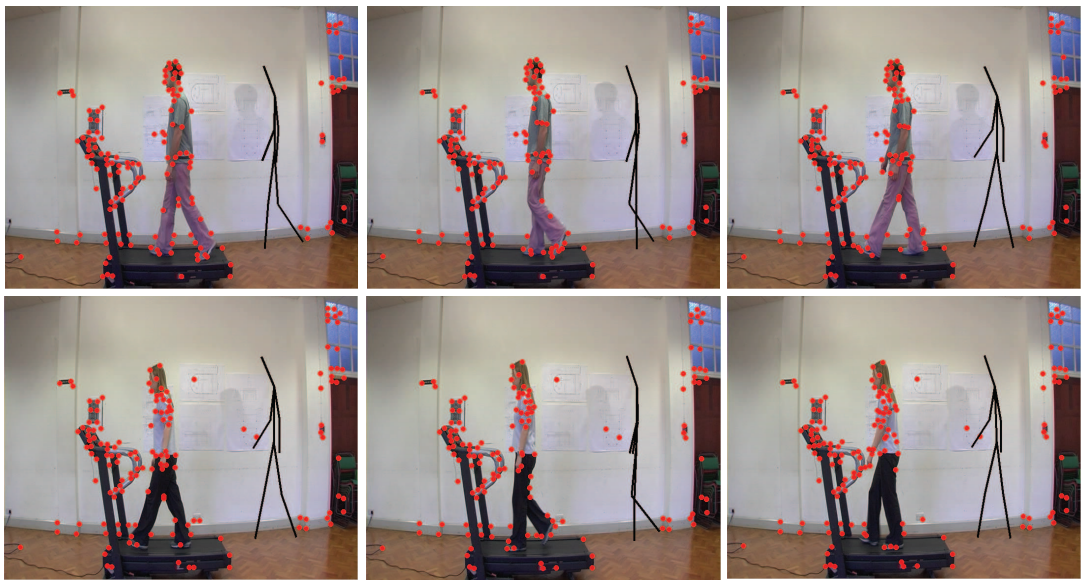


Figure 3.23: *Qualitative results of phase estimation. Sample frames are shown for two different subjects. On each frame a pose is plotted which is representative of the phase estimated for that frame. The set of tracked features are also plotted as red circles.*

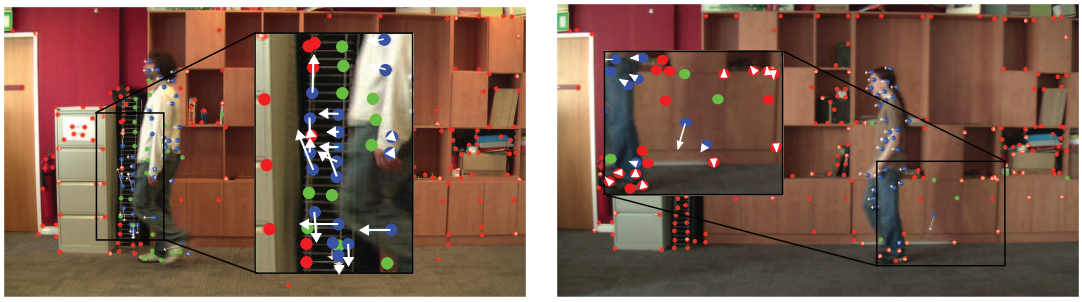
feature's motion is used and not the position.

3.5 Estimating Phase for Moving People

In the previous experiments motion models have been learnt and applied to people walking on a treadmill, in this section it is investigated how the same models can be applied to people walking normally. The current models can not be directly applied to people walking in real scenes as the person's net translational motion will be present in each observation. Therefore, this net translation must first be estimated in each frame so that it can be compensated for.

As in the previous sections 10 sequences were filmed. Each contains a different person walking from the right to the left of the scene with a static camera, nine of the subjects used are different to those from the previous section. The sequences are filmed in high definition with a resolution of 1920 x 1080 and then down sampled to a resolution of 1024 x 576. This method is used so that the subjects will have approximately the same height as in the previous section, but will remain in shot for longer than if filmed at a resolution of 720 x 576. As the scene is larger and more cluttered 250 KLT features are tracked and a minimum distance between features is set as 15 pixels, this is to ensure some features will track the foreground object rather than just image features belonging to the background.

This scene will challenge the algorithm in two ways. Firstly, as described it will test the algorithm on people that are not stationary and secondly, the tracking will contain much more noise than previously used. Example frames are shown in Figure 3.24 where the scene used is shown and two types of tracking error are illustrated. In Figure 3.24 (a) errors are shown as the subject occludes a set of features attached to the CD rack in the background, as these features are occluded the KLT feature tracker finds similar features elsewhere in the image, this results in spurious motion estimation. In Figure 3.24 (b) a feature is illustrated that is tracking the shadow of the person walking; also notice the change in lighting across the sequences. Another problem is that often features that were tracking the back edge of the subject will start to track a feature on the background. The problem is



(a) (b)

Figure 3.24: *Example frames showing the scene used to create sequences of moving people. Two causes of spurious tracking are illustrated. Features becoming occluded (a) and features tracking the subject's shadow (b). Blue circles represent features classed as tracking the foreground, red the background and green features that could not be tracked across consecutive frames.*

these features will initially be classified as foreground features, a number of frames are then required before these features will be reclassified as tracking the background.

Initially the velocity of each subject is manually extracted by labeling the position of the hip in the first and last frame, then assuming the subject walked at constant velocity through the sequence, as it is assumed the ground plane will lie horizontally along the x-axis of the image it is assumed only motion along this axis will require compensating for. This will test the approach's ability to overcome noisy features rather than inaccurate motion estimation. Further to this, initially features are manually segmented into foreground and background features. This is as it is first desirable to determine the accuracy of the presented method at estimating phase assuming perfect segmentation can be achieved. Including spurious motions such as those shown in Figure 3.24 will do little but to distort the resulting confusion matrix.

The confusion matrix for phase estimation assuming perfect foreground/background segmentation is shown in Figure 3.25. Whilst the cells around the diagonal still appear the lightest this is not as clear or well defined as that shown in Figure 3.17. This shows that the accuracy has dropped given the increased noise in the observations.

This is further illustrated in Figure 3.26 where the accuracy is shown as a function

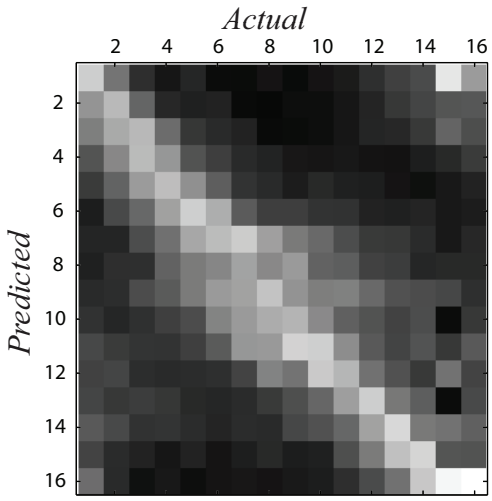


Figure 3.25: *Phase estimation confusion matrix - lighter areas show positions that had a higher frequency ($\lambda = 10.0$).*

of λ. The accuracy is now much lower than that shown in Figure 3.18 and much closer to chance ≈ 0.06. However, Figure 3.25 shows that most features lie near the diagonal, on Figure 3.26 the fraction of features correctly classified to ± 1 phase of the ground truth are also shown. This shows a much higher fraction of features are correctly classified than chance (≈ 0.19). The curve shown is much flatter than that for people walking on a treadmill, this is partly as features will be on average tracked for a much shorter period when someone is walking and moving across a scene compared to when they are walking on a treadmill. Therefore, increasing the value of λ will have little significance. The accuracy does increase with λ, though the effect is not as noticeable as in Figure 3.18.

The phase estimation error using a HMM is shown as a function of σ in Figure 3.27. The shape of the graph is the reverse of that shown for the stationary case in Figure 3.21. Whilst Figure 3.21 showed it was better to rely on the observations than the model, in Figure 3.27 the opposite becomes true, the model is more reliable than the observations. This is a direct result of the tracking being less accurate. There is however, a small minima at $\sigma = 0.25$ which is a similar value to where the minima was found in Figure 3.21 suggesting this region is where both model and observations contribute. One of the reasons the error does not increase at high values of σ as in the treadmill case is that the sequences of people walking across

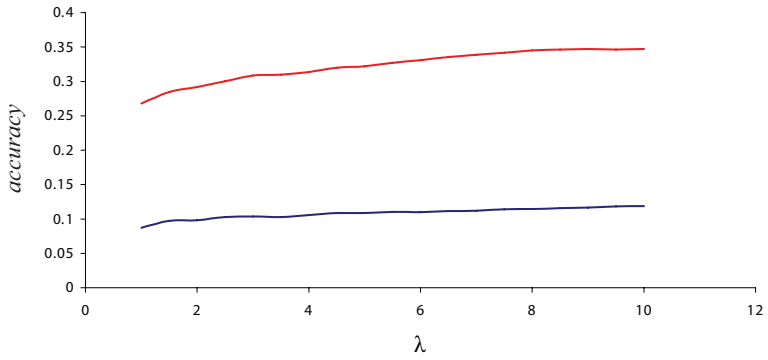


Figure 3.26: *Phase estimation accuracy as a function of λ. The blue line shows the accuracy of features classified to the correct phase. The red lines shows the accuracy of features classified to ± 1 of the correct phase.*

the scene are about half the length of those used when a person was walking on a treadmill. The consequence is that if the model has a slightly different temporal length to that of the person being observed, over a short period the phase and model won't become significantly out of phase, whereas over a longer period a large error would be expected to accumulate.

However, despite the tracking errors Figure 3.27 shows that the presented method is still capable of accurately estimating gait phase, though the minimum error achieved is slightly higher than when each subject was walking on a treadmill. The loss in accuracy is caused by two effects. The first is that the features are not tracked as accurately and the second is that each feature is tracked over less frames before being lost.

3.5.1 Automatically Estimating Foreground Velocity

In the previous section the velocity of the subject walking was estimated manually for each sequence, in this section a method is presented to perform this task automatically. This is achieved by tracking the foreground object, however, this is not trivial, whilst RANSAC can be used to initially segment the features into foreground and background features this segmentation will contain many errors. Furthermore, even given a perfect segmentation it is still not trivial to determine where the foreground object is as the features are sparse and are randomly distributed across the

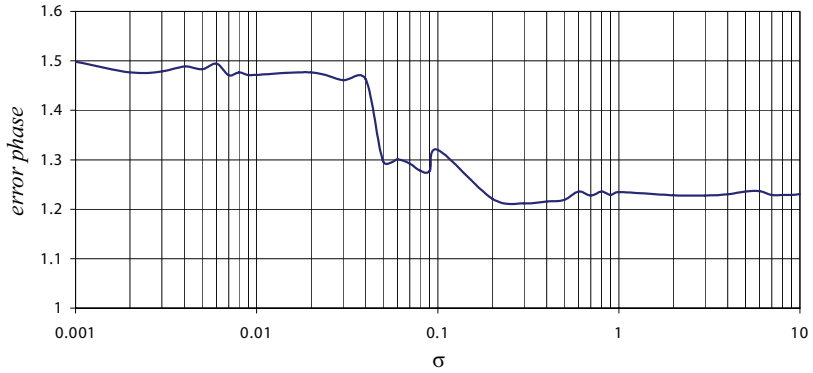


Figure 3.27: *HMM average phase estimation error as a function of σ.*

body. Techniques such as using the centre of mass of the features will be inaccurate since often the distribution of features are not uniform and more features may be present on the upper rather than lower part of the body. Finding the outer limits of the foreground cluster is also inaccurate since there will often be outliers that may be located a large distance from the foreground object. To overcome these difficulties a particle filter, which is briefly described below, is used to propagate a bounding box.

As first presented in Section 2.5 a Bayesian recursive filter can be described by the following equation

$$\underbrace{p(X_t|\mathcal{O}_t)}_{posterior} = \underbrace{p(\mathcal{O}_t|X_t)}_{observational} \underbrace{\int_{X_{t-1}} p(X_t|X_{t-1})p(X_{t-1}|\mathcal{O}_{t-1})}_{prior} \tag{3.23}$$

where X_t is the state at time t and $\mathcal{O}_t$ is an observation made at time t. The result of evaluating the integral in Equation 3.23 results in a prediction or prior over the variable X_t, this allows Equation 3.23 to be written as

$$p(X_t|\mathcal{O}_t) = p(\mathcal{O}_t|X_t)p(X_t|\mathcal{O}_{t-1}) \tag{3.24}$$

It would be desirable to calculate a property of the posterior distribution in the above equation, such as the expectation value $\hat{X}$. This can be achieved by integrating Equation 3.24 over X_t, however, in practice this integration may be very difficult to perform. A particle filter overcomes this by performing this integration using Monte Carlo methods. A standard Monte Carlo approach would achieve this by directly drawing a set of random samples $\{X_t^1, .., X_t^N\}$ from the posterior distribution $p(X_t|\mathcal{O}_t)$ and then performing the integration as a summation over the samples such that

$$\hat{X} = \int_{X_t} p(X_t|\mathcal{O}_t) X_t dX_t = \frac{1}{N} \sum_{n=1}^{N} X_t^n \tag{3.25}$$

However, if the posterior distribution is unknown it is very difficult to directly draw samples from. The particle filter makes use of Importance Sampling which negates this problem. This is achieved by sampling from a proposal function $q(X_t)$, which can be selected such that it is easier to directly draw samples from and then assigning each sample a weight according to

$$w^n \propto \frac{p(X_t^n|\mathcal{O}_t)}{q(X_t^n)} \tag{3.26}$$

Using this method it is only required that the posterior $p(X_t^n|\mathcal{O}_t)$ is known up to a multiplicative constant, which can easily be calculated. An obvious choice for the proposal function is the prior distribution $p(X_t^n|\mathcal{O}_{t-1})$, substituting this and Equation 3.24 into Equation 3.26 the weight of each particle is simply given by

$$w^n \propto p(\mathcal{O}_t|X_t^n) \tag{3.27}$$

A particle filter that uses this choice of proposal function is called a Sampling-

Importance-Resampling (SIR) particle filter [5]. From this set of particles a further set can be drawn where the likelihood of a sample being retained is proportional to w^n, once this further set has been selected it can be used to evaluate the posterior using Equation 3.25.

Particles are propagated across consecutive frames through the state evolution function $X_t^n = f(X_{t-1}^n, v_{t-1})$, where v_{t-1} is process noise. Passing samples through this function effectively generates samples from the prior distribution $p(X_t|\mathcal{O}_{t-1})$. This prior is of course the same distribution used as the proposal function $q(X_t)$ and the samples generated from it can be used for Importance Sampling. It is assumed that it is easy to generate the process noise v_{t-1}, often this is assumed to be Gaussian.

As with a HMM, a particle filter has three principal components. The first, as described above, is the state evolution function $X_t = f(X_{t-1}, v_{t-1})$, the second is the observation likelihood function $p(\mathcal{O}_t|X_t)$ and the final component is a prior distribution $\pi(X_1)$ used to initialise the particle filter in the first frame.

The purpose of the particle filter is to propagate a bounding box that can be used to track the foreground object. From this the translational motion of the subject can be estimated and compensated for. It is therefore intuitive to define the function $p(\mathcal{O}_t|X_t)$ as being higher if a higher proportion of features classified as tracking the foreground are encompassed by the bounding box. A function of the same form used to convert a state's votes to a likelihood in Section 3.4.4 can be used

$$p(\mathcal{O}_t|X_t^n) \propto e^{-\left(\frac{1-N(X_t^n)/N}{\kappa}\right)^2} \tag{3.28}$$

where $N(X_t^n)$ is the number of foreground features encompassed by the bounding box if its centre is located at the position X_t^n. For all experiments a value of $\kappa = 1.0$ was used, this value was determined empirically.

The evolution function $X_t = f(X_{t-1}, v_{t-1})$ is defined as a random walk where each step is generated randomly from a spherical Gaussian distribution with standard deviation of 50 pixels. This assumes that there is no prior expectation about how a

subject will walk through the scene.

It is assumed that the subject's starting location is not known in the first frame so a flat prior is used. Each particle is drawn from a uniform distribution across the image dimensions.

To achieve a good initialisation seven iterations of the particle filter are applied to the first frame of each sequence. RANSAC was used to segment the features into foreground and background features using the features' motion. To improve accuracy the motion of each feature was averaged over a ten frame sliding window [47].

In Figure 3.28 the set of particles are shown and how they are propagated across two consecutive frames. The expected position of the bounding box is also shown in each frame. A fixed size bounding box is used for all subjects (250 by 450 pixels). This is set so that the box is larger than any person that is expected to be observed.

In Figure 3.28 it can be observed that more particles are located on the back half compared to the front half of the subject being tracked. This is as there will be a slight bias towards where the distribution was located in the previous frame. This bias is the prior; whilst the particles are propagated through a random walk the centre of the distribution will remain unchanged. Therefore, it is expected that there will a bias in the position of the particles towards where the centre of the distribution was located in the previous frame. Further examples frames are also shown in Figure 3.29.

Quantitative results using automatic motion estimation are shown in Figure 3.30 using three different methods of velocity estimation. In all three a particle filter is initially used to track the foreground object. The first method calculated the motion for each frame directly from this signal. The second smoothed the signal with a Gaussian filter before calculating the motion for each frame. The third calculated the average motion across the entire sequence and assumed the subject would walk with constant velocity. The errors shown are greater than those presented in the previous section, this is not only because the motion of the subject is not as accurately estimated, but also because unlike in the previous section the foreground segmentation of the object is performed automatically. So these errors represent the

Figure 3.28: *The propagation of particles. The top row shows the same frame but with the initial set of particles (left) and the set of particles after they have been allowed to perform a random walk (right) before being applied to the next frame. The bottom row shows the next consecutive frame, with the initial distribution (right) and the distribution after importance sampling (left). The resultant estimated position of the bounding box is also shown for each frame. The red circles show the set of foreground features being used and each white point shows the location of each particle that defines the centre of a bounding box.*

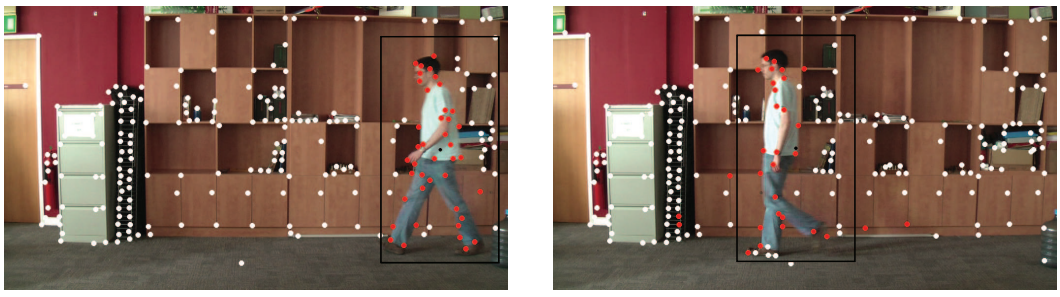


Figure 3.29: *Qualitative results of tracking the foreground object using a particle filter. The white circles show features classed as tracking background and the red circles the foreground.*

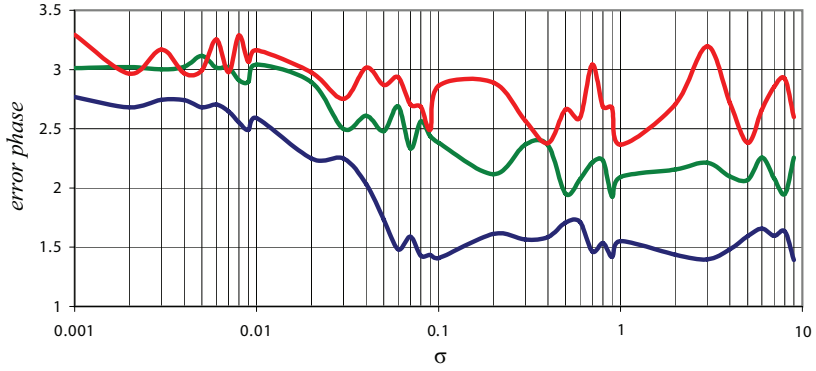


Figure 3.30: *HMM average phase estimation error as a function of σ using automatic foreground velocity estimation. The red line shows the error through estimating motion using only a particle filter, the green line shows the error after smoothing using a Gaussian filter (standard deviation = 25 frames) and the blue line shows the error after calculating the average velocity from the entire sequence.*

error that the system could be expected to achieve.

The phase estimation error for the method that uses the raw output of the particle filter has the largest error and this error is only slightly effected by the value of σ compared to the other methods, suggesting the raw output of the particle filter contains too much noise to accurately estimate phase. The filtered signal performs slightly better, however, assuming constant velocity across the entire sequence provides the smallest errors.

In Figure 3.31 the effect of inaccurate motion estimation is shown on the accuracy of estimating gait phase. These results were obtained by adding a constant velocity to the manually extracted motion for each subject. This shows that the accuracy quickly drops as even small amounts of noise are added to the observations. For comparison the average velocity of the subjects in the x-axis is 11.5 pixels per frame. This graph shows that the velocity must be calculated accurately if the performance presented in the previous section is to be achieved.

In Figure 3.32 some example frames are presented showing the extracted phase. Again there is close agreement between the model and the person's pose shown in each frame. However, notice that the stride length of the subject shown is signif-

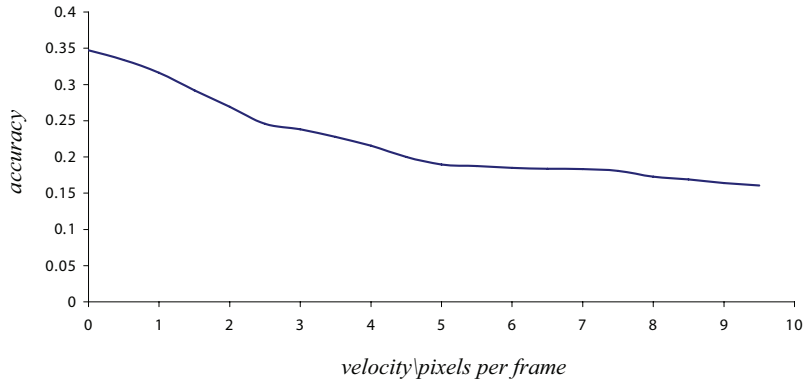


Figure 3.31: *The effect of inaccurately estimating velocity on phase estimation, the accuracy rate shown is for those features correctly classified within* ± 1 *of the hand labeled groundtruth phase.*

icantly larger than that of the model, this highlights that whilst estimating phase presents a very coarse estimation of pose, this estimation will clearly never be able to extract the individual characteristics of a person's walk. For this the location of the features must be explored.

3.6 Summary

In this chapter a method has been presented to represent motion based on learning a deformable shape of the trajectory of motion across the image plane. The motion models learnt can be used to extract information from each observed trajectory independently. It was shown that whilst the models provide above chance accuracy at estimating which limb a motion was most likely caused by, this accuracy is not high enough to allow the models to be used as hard classifiers. Instead they are more suitably used as soft classifiers, to provide only the likelihood of a feature tracking a particular limb.

The accuracy of using the models to estimate phase was also explored and it was found by integrating the data extracted by all foreground features that a HMM could be used to accurately extract phase for each frame in the sequence. This technique has been designed to exploit the features' motion by trying to extract all present information; there have been no attempts to use methods such as selecting the best

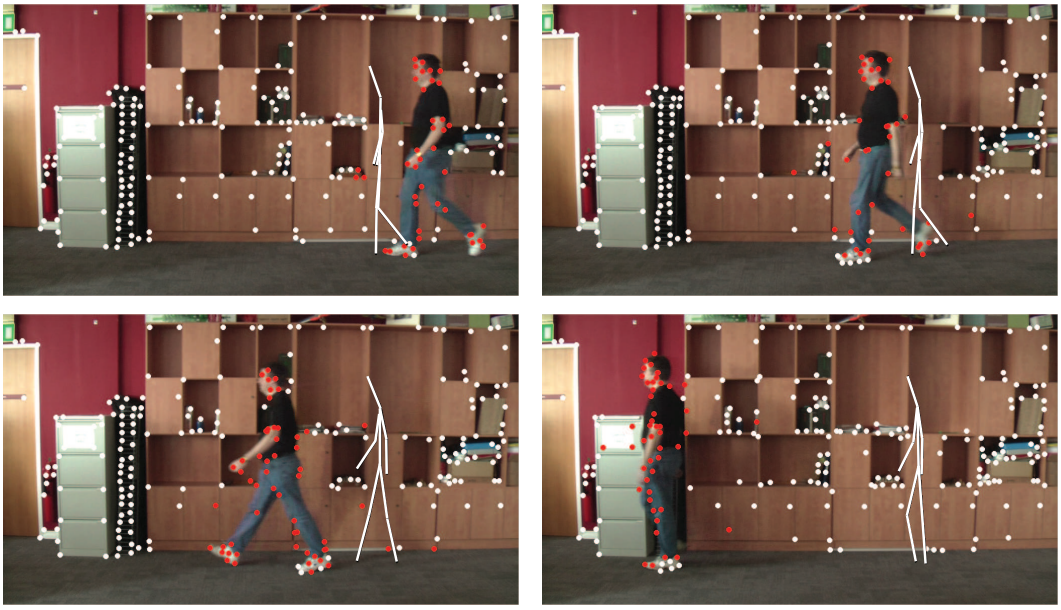


Figure 3.32: *Qualitative results of phase estimation. Sample frames are shown from two different subjects. On each frame a pose is plotted which is representative of the phase estimated for that frame. The set of tracked features are also plotted as red circles.*

features or using heuristics to solve this difficult problem.

These methods were extended to tracking people walking through a scene by compensating for each subjects' translational motion. This was achieved by using a particle filter to propagate a bounding box to track the foreground object from which their motion could be estimated. It was found that using a prior model, in this case constant velocity, significantly improved performance. The scene used for these experiments were particularly challenging, due to the cluttered scene and poor lighting which resulted in many large tracking errors.

All results were achieved without the need to scale models used and were tested on subjects with differing cadences demonstrating the robustness of the presented approach.

The contribution made by this chapter is to demonstrate that information can be extracted from the motion of a sparse set of features and the noise present in these features is not large enough to completely mask all information contained in them. Even without exploiting the position of any of the features a coarse estimation of

pose has been provided using motion alone suggesting that accurate pose estimation using only a sparse set of features is possible.

Chapter 4

Estimating Pose in the Image Plane

In this chapter methods to exploit the spatial location of the tracked features are explored. This is achieved using articulated models combined with techniques that allow efficient searches to be performed across the entire image plane. Experimental results are provided that demonstrate both the accuracy and robustness of the presented technique.

The motion models presented in the previous chapter allowed two pieces of information to be extracted for each tracked feature, gait phase and the most likely limb the features is tracking. Whilst each feature's phase was exploited in the previous chapter to estimate the global phase of the motion being performed, the likelihood of a feature tracking a particular limb was not. It is this information that will be utilized in this chapter.

Probability maps are created that describe the likelihood of a limb being at a specific image location in each frame based on the motion observed at that location. Efficient searches that exploit the graphical structure of the model used to represent a human are used to estimate pose in each frame. To improve the efficiency of the search a prior model will be used that is dependent on the subject's gait phase estimated in the previous chapter. This will allow the search space to be reduced and improve the accuracy of the presented method.

The purpose of this chapter is to demonstrate that despite the sparsity and unpredictability of what features on the body will be tracked, 2D pose can still be estimated. This in turn implies that 3D pose estimation could be achievable.

4.1 Pictorial Structures

The Pictorial Structure was first introduced by Fischler and Elschlager [41] as a way of searching for instances of a known object in an image by decomposing the object into a set of principal parts, where each part is modeled separately and the relationship between each part is represented as a spring like connection. Efficient methods to search for these Pictorial Structure in an image were presented using Dynamic Programming (DP). This representation was further explored and developed by Felzenswalb and Huttenlocher [35] and applied to articulated objects such as people, faces and cars [37]. It was also shown how the matching problem could be presented in a Bayesian framework rather than that of energy minimization as presented in [41]. In the presented description of Pictorial Structures the notation from [35] is followed.

A Pictorial Structure is defined as a collection of parts and connections that define how one part should be placed relative to another. This structure can be represented by the graph $G = (V, E)$ where $V = \{v_1, .., v_n\}$ is the set of n vertices of the graph and $\{v_i, v_j\} \in E$ are the set of edges that join the graph's vertices, not all vertices are necessarily joined. The graph's vertices represent each individual part of the model; there is one vertex for each part. The edges represent the spring like connections between each of the parts. An instance of an object can be described as $L = \{l_1, .., l_n\}$, where vertex v_i is placed at location l_i, which represents a location in the image plane. The function $m_i(l_i)$ describes the goodness of fit when part v_i is placed at location l_i in terms of how well the observation at this location agrees with the model. A low value corresponds to when the model and observations agree and a high value corresponds to when there is a large mismatch between the two.

The function $d_{ij}(l_i, l_j)$ represents the deformation cost of placing vertex v_i at location l_i and vertex v_j at location l_j, this deformation cost can be thought of as the

stretching of the spring like connections between parts. A large stretch will result in a high deformation cost.

The optimal match between image and model can then be defined as

$$L^* = arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right) \tag{4.1}$$

To solve Equation 4.1 two sets of terms must be minimized, the deformation terms and the observation terms, it is important to note that these can not be minimized independently since they are both dependent on the same parameters. The minimum of Equation 4.1 represents a compromise between the expected configuration of the Pictorial Structure and the most likely configuration given only the observations.

By considering Bayes' theorem it can be shown that Equation 4.1 is equivalent to maximising the posterior distribution given by

$$p(L|\mathcal{O}, \theta) \propto p(\mathcal{O}|L, \theta)p(L|\theta) \tag{4.2}$$

through the minimization of the negative log-likelihood, where $\theta = \{U, C\}$ are the model parameters and $\mathcal{O}$ represents the current observations. The model parameters consist of the observation parameters $U = \{u_1, .., u_n\}$, which models the expected appearance of each part and $c_{ij} \in E$ which models the connection between each of the parts.

If each part can be modeled independently such that $p(\mathcal{O}|l_i, u_i)$ is the probability of observing $\mathcal{O}$ given the part v_i is placed at location l_i and the observation parameter u_i, the term $p(\mathcal{O}|L, \theta)$ can be rewritten as

$$p(\mathcal{O}|L,\theta) = \prod_{i=1}^{n} p(\mathcal{O}|l_i, u_i) \tag{4.3}$$

Furthermore if each deformation is conditionally independent then the 2nd term on the right hand side of Equation 4.2 can be written as

$$p(L|\theta) = \prod_{(v_i,v_j)\in E} p(l_i, l_j|c_{ij}) \tag{4.4}$$

where $p(l_i, l_j|c_{ij})$ is the probability of placing part v_i at location l_i and part v_j at location l_j given the connection parameter c_{ij}. Equation 4.2 can then be reformulated as

$$p(L|\mathcal{O},\theta) \propto \left(\prod_{i=1}^{n} p(\mathcal{O}|l_i, u_i) \prod_{(v_i,v_j)\in E} p(l_i, l_j|c_{ij}) \right) \tag{4.5}$$

Taking the negative log-likelihood of Equation 4.5 then results in an equation with exactly the same form as that of Equation 4.1 where $m_i(l_i) = -\log p(\mathcal{O}|l_i, u_i)$ and $d_{ij}(l_i, l_j) = -\log p(l_i, l_j|c_{ij})$. So it can be observed that the minimization in Equation 4.1 is equivalent to maximising the posterior distribution. Furthermore it becomes clear that when put in terms of Bayesian statistics the deformation terms of Equation 4.1 represent a prior on the expected model configuration.

4.2 Efficient Searches

The graph G that represents the structure of the human body is assumed to be a tree and contains no closed loops. This graph is depicted in Figure 4.1. To exhaustively

search for the set of locations to minimize Equation 4.1 would require complexity h^n, where h is the number of possible locations a part can be placed and n is the number of parts. However, since the graph used is acyclic the location of each node in the graph is dependent only on its children and the search can be performed more efficiently. This dependence on only the child nodes represents the same Markovian property that was assumed in the HMM and Particle Filter presented in the previous Chapter. Furthermore, the search that is used to efficiently minimize Equation 4.1 is akin to the Viterbi algorithm. The exception being that the algorithm used here is modified to allow paths in the graph to converge, for example in Figure 4.1 where the head and elbows join to the same node (the shoulder). The method used is Dynamic Programming and will be shown to have complexity h^2n.

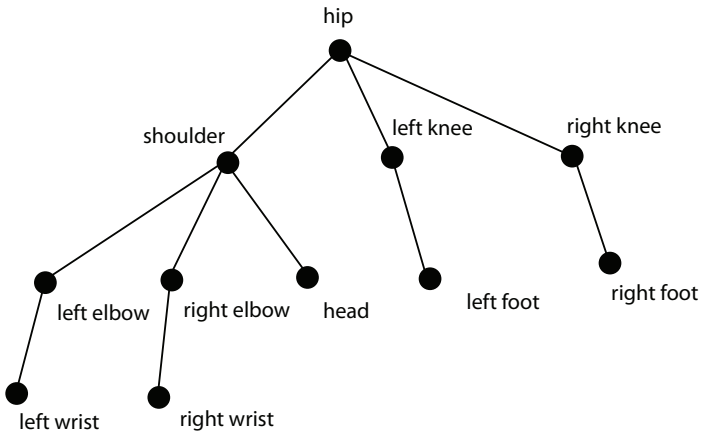


Figure 4.1: *Graph structure used to represent Pictorial Structure, the hip is shown as the root node. Notice there are no loops present in the graph.*

The value of L^* determined from maximising Equation 4.2 is often referred to as the Maximum a Posterior (MAP) estimate and represents the mode of the posterior distribution.

The algorithm starts at the leaf nodes of the tree (those that have no children) and works towards the root node, effectively passing information up the tree. As in the Viterbi algorithm an intermediate function is computed at each iteration. These are denoted as $B_j(l_i)$, these describe the cost to locate part v_j at the optimal location l_j^* and can be written as a function of only the parent node's location l_i. These functions can be calculated for any leaf nodes in the graph as

$$B_j(l_i) = \min_{l_j} \left(m_j(l_j) + d_{ij}(l_i, l_j) \right) \tag{4.6}$$

A function $D_j^*(l_i)$ is defined for each node to record the best location for part v_j as a function of the location of its parent l_i and can be calculated as

$$D_j^*(l_i) = arg\min_{l_j} \left(m_j(l_j) + d_{ij}(l_i, l_j) \right) \tag{4.7}$$

For any nodes that have children excluding the root node the intermediate functions are calculated as

$$B_j(l_i) = \min_{l_j} \left(m_j(l_j) + d_{ij}(l_i, l_j) + \sum_{v_c \in C_j} B_c(l_j) \right) \tag{4.8}$$

Where v_c are the children of v_j and it is assumed the functions $B_c(l_j)$ have already been computed. The functions $D_j^*(l_i)$ can be calculated by replacing the min with $arg\min$ in Equation 4.8. The cost function for the root node is calculated as

$$B_r(l_r) = (m_r(l_r) + \sum_{v_c \in C_r} B_c(l_r)) \tag{4.9}$$

The MAP estimate can then be calculated by first finding the position l_r^* that minimizes Equation 4.9, the optimal location for each child node can then be found through the function $D_j^*(l_i)$ so that $l_j^* = D_j^*(l_r^*)$. This process is then repeated for each node moving down the tree until the optimal position of all nodes is known.

The algorithm operates by first gathering information up the tree towards the root

node, then once the position of the root node has been estimated, the algorithm traces back down the tree extracting the optimal position of each node it passes. Typically to minimise Equation 4.6-4.9 for each location l_i requires a search through all possible locations l_j of which there are h. This has to be repeated for all l_i of which there are also h. Therefore, to calculate each $B_j(l_i)$ requires complexity h^2. This has to be repeated for each node in the graph of which there are n, giving total complexity to minimise Equation 4.1 of h^2n.

The DP algorithm used in this work is modified to be more efficient by defining the deformation cost as

$$d_{ij}(l_i, l_j) = \begin{cases} -\log p(l_i, l_j|c_{ij}) & \text{if } \; l_j \in l_i + T_{ij}(\theta) \\ \infty & \text{otherwise} \end{cases} \tag{4.10}$$

where $\theta = \{\theta_1, .., \theta_k\}$ represents a set of k angles and $T_{ij}(\theta)$ is a function that calculates possible positions of l_j given the angle set θ. This function is defined since limb lengths are fixed. Given a location l_i there are only a small number of possible locations for l_j. In practice Equation 4.8 is minimised over l_j through the parameter θ. In general k is much smaller than the number of locations in the image, the result is that the complexity of finding the minimum of Equation 4.1 increases linearly with the number of grid locations, since the complexity is now hkn, where h is the number of possible locations for a parent limb (the number of pixels in the image), k is the number of positions a child node can be located conditioned on the location of the parent node and n is the number of nodes in the model.

4.3 Model Representation

The spatial model is represented as a set of joints, where the position of a joint with respect to its parent is defined by an angle measured relative to the horizontal $\phi(l_i, l_j)$ and a fixed distance L_{ij}. In current approaches the relative angle between two limbs (three joints) is typically used, this allows the conditional dependence

between them to be modeled and stops unlikely poses being inferred. However, in the presented approach a different spatial model is used for each gait phase, this means the model is well enough constrained so that unlikely poses do not occur. This assumption allows the search space to be further reduced as the orientation of a parent joint is not of importance, only its position. This assumption implies that people are expected to be upright whilst walking and that scenes are filmed with the ground plane at the bottom of the image.

The fixed distance L_{ij} is taken to be the mean length between the ith and jth parts across all training data. This is assumed to be constant and can not deform. The prior for the angle on each joint $p(l_i, l_j|c_{ij})$ is defined by a Von-Mises distribution:

$$\mathcal{M}(\phi(l_i, l_j), \mu_{ij}, \kappa_{ij}) \propto e^{\kappa_{ij}\cos(\phi(l_i,l_j)-\mu_{ij})} \tag{4.11}$$

where μ_{ij} represents the mean angle of the distribution and κ_{ij} defines how constrained the joint is. Learning a different prior for each phase consists of estimating different values for the parameters μ_{ij} and κ_{ij} for each limb.

The mean angle μ_{ij} can be estimated from training data by first calculating the mean sine of the angles and the mean cosine of the angles then taking the arctan of the two. This is equivalent to representing each angle as a directed vector with length $1/m$, where m is the number of data points, then placing them all end to end and taking μ_{ij} to be the angle of the resultant vector. The parameter κ_{ij} can be calculated using approximations described in [39].

The models can be learnt from exactly the same training data as used to learn motion models in the previous chapter. A different model is learnt for each phase of gait, using the subset of training data for each phase. This involves learning a different set of connection parameters $c_{ij} = \{\mu_{ij}, \kappa_{ij}\}$ for each phase. It is expected that the limb lengths will remain constant for all phases, so these lengths can be learnt using all training data. Some example models are shown in Figure 4.2, these show the prior pose expected for each phase shown.

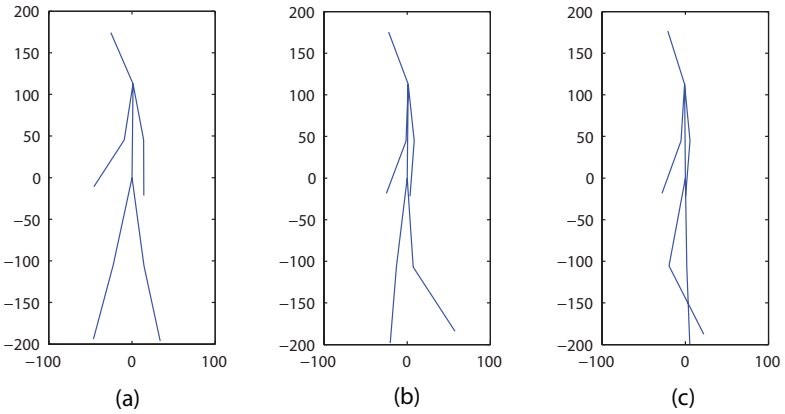


Figure 4.2: *Example prior pose for phases 1 (a), 5 (b) and 10 (c).*

4.4 Creating Dense Probability Maps

Before the optimal location can be found in each frame the term $p(\mathcal{O}|l_i, u_i)$ must first be defined. This term is the probability of making the observations $\mathcal{O}$ given v_i is located at position l_i and the model parameters u_i. So this probability describes how well the observations can be explained by assuming a limb is located at a particular location.

In the previous Chapter motion models were introduced that allow the likelihood of a particular motion being observed given it was tracking a particular limb to be estimated. It is these likelihoods that will be used to create the probability distributions $p(\mathcal{O}|l_i, u_i)$. These distributions are referred to as probability maps since they describe at every pixel in the image the likelihood that the observed motion was caused by a particular limb moving at that location.

As the position of each tracked feature is known this provides the likelihood of a specific joint being at that position in the image, this allows a probability map to be created for each joint. However, the problem is that the tracked features are very sparse meaning that the likelihoods at most of the pixels in the image are missing. To avoid limiting the search over image locations where only features are present, the likelihoods between features must be inferred so that dense probability maps can be constructed.

Simply using standard interpolation techniques to perform this inference suffer with two major problems. Firstly, due to the sparse data the resultant probability maps lack detail, meaning large regions have similar likelihoods and parts such as individual legs can not be distinguished. Secondly, if a feature is particularly noisy and is assigned a very low likelihood, this feature will corrupt neighbouring features that may have a high likelihood.

The dense probability maps should have two properties: detail should be maintained and noisy features with very low likelihoods should be replaced with an alternative likelihood from a neighbouring feature.

To achieve this consider a set of locations on a grid $x \in \mathcal{G}$, where the grid represents the image pixels. The observed features classified as tracking the foreground lie on a subset of the grid $\mathcal{B} \subset \mathcal{G}$, at these locations the negative log likelihood of each feature tracking the joint in which we are interested $m_i(x)$ is known, as calculated in the previous chapter using the feature's motion. At a location x_m further away from an observed feature $x_n \in \mathcal{B}$ the probability should decrease to reflect the increased uncertainty in that observation at the current location. This is represented by a zero mean Gaussian $p(x_m, x_n) = \mathcal{N}(x_m - x_n, 0, \sigma)$. The inferred likelihood at each location x_m of the grid is calculated as

$$m_i(x_m) = \min_{x_n \in \mathcal{B}} (m_i(x_n) - \log(p(x_m, x_n))) \tag{4.12}$$

Since $p(x_m, x_n)$ is defined as a zero mean Gaussian, Equation 4.12 can be efficiently calculated as a distance transform using the techniques described in [34]. An example of a calculated probability map with the features overlaid is shown in Figure 4.3. Regions with a higher likelihood are represented by darker colours; this is as the probability maps actually represent the negative log of the probability. There is a local minimum around every feature point; this makes it preferable for a limb to be located in the neighbourhood of a pixel containing a feature. In Figure 4.3 it appears as though the hip could be located at the position of any tracked feature with equal likelihood, this is not the case and is because the figure shown does not have enough dynamic range to show the full variation in probability. What would

be expected is that the likelihood would be greater at locations where small hip like motions were observed. The converse of this can be seen on the back leg where the likelihood is clearly lower.

The constant σ can be thought of representing a measure of how close to a limb a feature is expected to be located, i.e. if there is a foot at position x it would be expected there is a feature tracking the foreground within $\pm\sigma$ pixels of this position. A probability map can be constructed for each limb over which the spatial search can be performed.

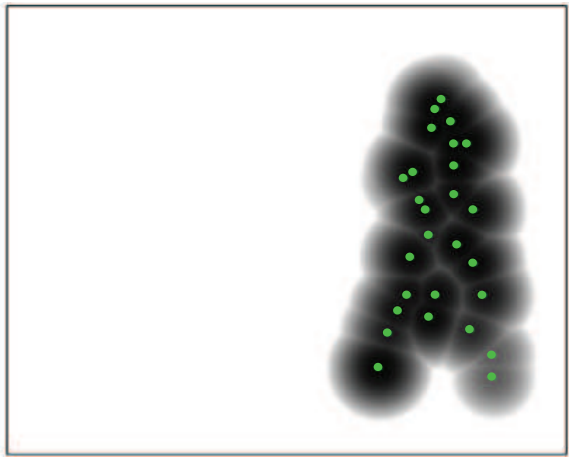


Figure 4.3: *Example of a likelihood map for the hip location with KLT features overlaid. Darker regions represent areas with a higher likelihood.*

Three consecutive frames with the extracted pose plotted are presented in Figure 4.4 these demonstrate the effect of changing the value of σ. When σ is large the prior will dominate Equation 4.1 and when σ is small the observations will dominate. This figure shows that when a low value of σ is used the model is over fitted to the observations; each limb is forced to be located in close proximity of a feature. However, when using a large value of σ the original prior is recovered. What would be expected is that at some intermediate value the observations and priors would both have a similar importance, such that the model could deform to the observations whilst still maintaining a likely pose.

It can also be seen in Figure 4.4 that the estimated position of the root node (the hip) moves a significant amount across consecutive frames. It would be expected that if the position of the root node can be accurately estimated it is likely the estimated

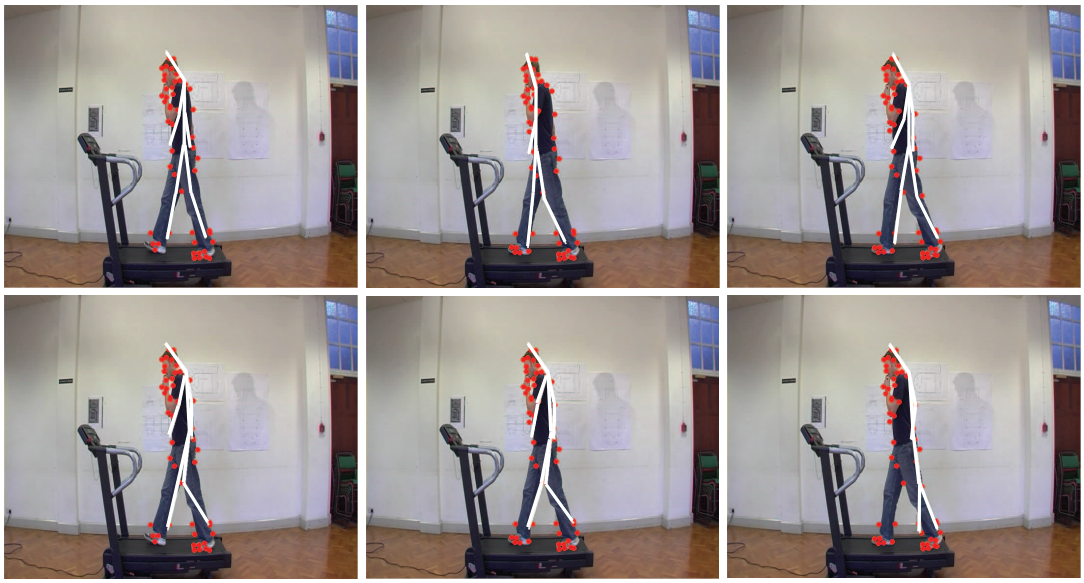


Figure 4.4: *Estimating pose using the MAP estimate and the effect of differing values of σ, three consecutive frames are shown for each value of σ. (top row) $\sigma = 100$ prior has large influence. (bottom row) $\sigma = 1$ prior has little influence, each joint is forced to be located in close proximity to a foreground feature. Foreground features are plotted as red circles.*

location of the other limbs will also be more accurate. A more robust method to evaluate the posterior distribution than using the MAP estimate is the expectation value. It would be expected that the posterior distribution for the root node would be unimodal, however, as the observations are both noisy and sparse the resultant distribution is likely to have many local maxima and minima. Whilst a DP solution is particularly well suited to evaluating noisy probability density functions since it effectively performs an exhaustive search, the global maxima found will most likely not coincide with the centre of the distribution, this is illustrated in Figure 4.5.

To calculate the expectation value the full posterior distribution $p(L|\mathcal{O})$ needs to be calculated. Through the DP solution presented the posterior distribution was maximised but this was achieved without having to actually calculate it. However, the approximation $B_r(l_r) \approx -\log(p(l_r|\mathcal{O}))$ can be used to calculate the expectation value for the root location $\langle l_r \rangle$. The location of the other nodes in the graph can then be extracted using the MAP estimate conditioned on the root position $\langle l_r \rangle$.

One of the difficulties with calculating an expectation value is that if the posterior

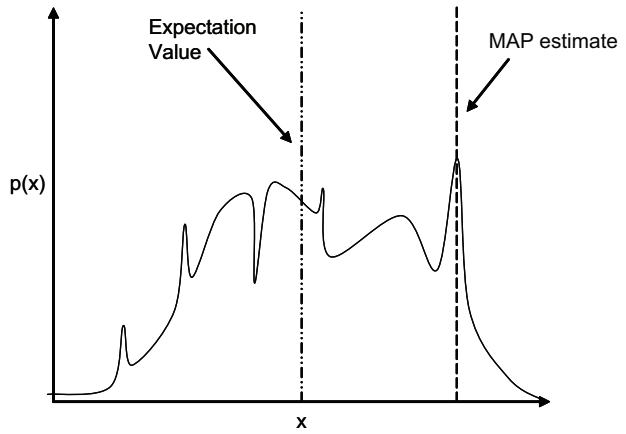


Figure 4.5: *Difference in evaluating a posterior by maximising it or by calculating the expectation value. As the distribution shown contains noise the expectation would be a far more robust measure than the MAP.*

distribution is sharply peaked then the expectation value will be the same as the MAP estimate. A solution to this is to smooth the posterior using techniques from simulated annealing [24] where

$$p'(l_r|\mathcal{O}) = p(l_r|\mathcal{O})^{\frac{1}{\gamma}} \tag{4.13}$$

The value of γ affects how smooth the resultant posterior will be. Whilst γ would normally be set to a constant here it is defined as

$$\gamma = \frac{\log(p(l_r|\mathcal{O})_{max}) - \log(p(l_r|\mathcal{O})_{min})}{\rho} \tag{4.14}$$

where ρ is a constant that specifies the order of magnitude between the lowest probability and the highest, in all the presented experiments this is set to 100. This makes the approach more robust since the degree of smoothing γ is calculated for each frame depending on the quality of the current observational data. The resultant probability distribution for the hip location is shown in Fig. 4.6 (a). The

distribution is very broad; this is expected as there is a large uncertainty in the exact position of the root node because the observational data was very sparse. The two horizontal lines in Fig. 4.6 (a) are because if the root node was located on either of these lines the outermost joints of the object could not be placed in the image, this has a zero probability.

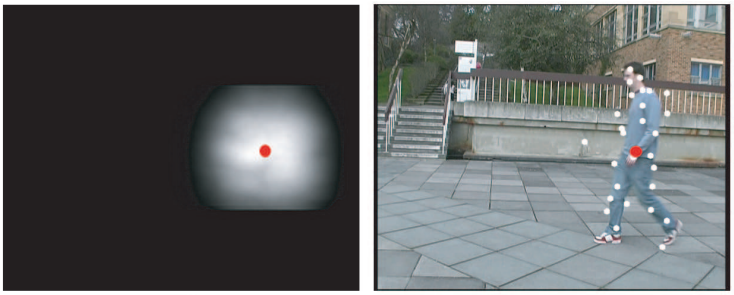


Figure 4.6: *The expectation value is shown as a red dot. (a) Resultant probability distribution for the root node (hip), lighter regions have a higher probability. (b) Corresponding image with KLT features classified as foreground features overlaid.*

Results using the expectation value and the MAP estimate to infer the location of the root node in the sequence shown in Figure 4.4 are presented in Figure 4.7. Notice the MAP estimate contains a large error (25.2 pixels) however the expectation value has a much smaller error (11.1 pixels). In the y-axis the frequency of the gait cycle can also be observed in the signal.

It can also be seen in Figure 4.7 (b), which shows the y-axis, that the signal extracted using the expectation value appears to be offset from the ground truth signal by ≈ -10 pixels. This is as tracked features are generally more densely distributed on the top half of the body so the solution tends to pull the estimated pose upwards so more limbs are located in closer proximity to other features. This results in the head be located too high and can be seen in Figure 4.4. This can be overcome by adding a weighting factor to the head so that it is forced to be located closer to a tracked feature. This can be achieved by setting $\sigma_{head} = \sigma/4$ so that when the dense probability maps are constructed the map for the head expects a feature to be located closer to where it will be placed. In a sense this is exploiting knowledge of the tracker, since features are more consistent on the top of the head estimating the head's position can be accurately achieved by being much more reliant on observations

compared to other limbs. The result of this method is also shown in Figure 4.7 where an accuracy of 5.1 pixels is achieved.

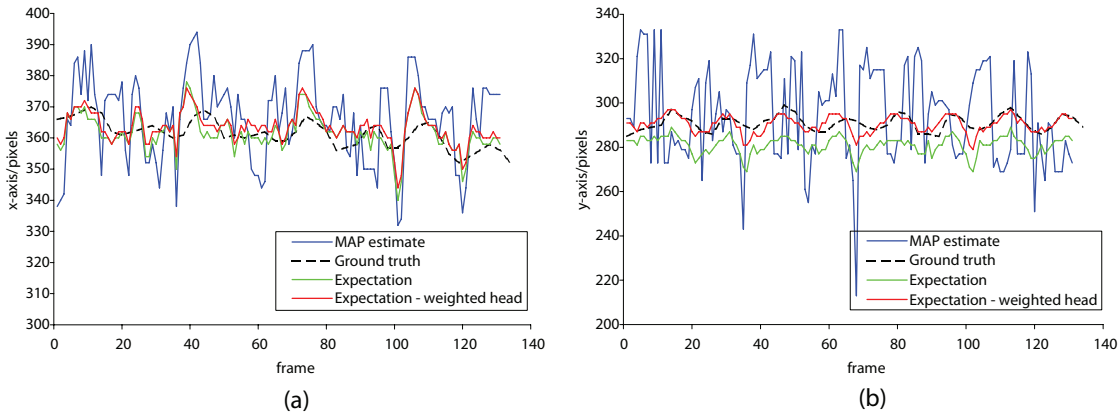


Figure 4.7: *Accuracy in estimating the position of the hip in the x-axis (a) and y-axis (b) for the sequence shown in Figure 4.4 using three different methods; MAP estimate (blue line), expectation value (green line), expectation value with weighting for head ($\sigma_{head} = \sigma/4$)(red line). The ground truth is shown as the black dashed line. The average error is 25.2 pixels, 11.1 pixels and 5.1 pixels for each method respectively.*

4.4.1 Initial Experiments

In this section initial results are presented using the methods described in the previous sections. For these experiments the data set of ten people walking indoors are used (as shown in Figure 4.8), this data set will be referred to as the moving data set (as apposed to the treadmill data set). This data set is used rather than the treadmill data set as people's gait tend to be significantly different when walking normally across a stationary surface. Since the models used were learnt from people walking on a treadmill this will be a greater test for the presented approach. Furthermore, this scene results in considerably noisier and more unreliable feature tracking so will act as a good indicator of the robustness of the presented technique.

As the model used is rigid, the size of the model will not deform to match a person being observed if they are taller or smaller than the model. To overcome this, the size of the person in each sequence is provided so that the prior model can be correctly scaled.

A low-pass filter is also applied to the root location in each frame to keep its movement temporally coherent. All of the results presented were performed on a grid of 256×144 rather than the original image size of 1024×576 pixels. A reduced grid size was used for efficiency, calculating pose using this resolution required just 0.15 seconds per frame of processing on a 2.6 GHz processor. The angular range searched over for each joint was set as $\pi/2$ radians. This search was represented as 15 discrete angles centered on the average angle for that joint in the given phase.

In Figure 4.8 example frames are shown illustrating the effect of using different values of σ, as the person shown has a much larger stride than that of the prior model (top row) it becomes more important that observations are exploited compared to those example frames shown in Figure 4.4. A value of $\sigma = 10$ is used as this allows a good compromise between prior and observations without over fitting. In Figure 4.8 features incorrectly estimated as being foreground features can also be seen. Despite this the location of the person is still accurately estimated.

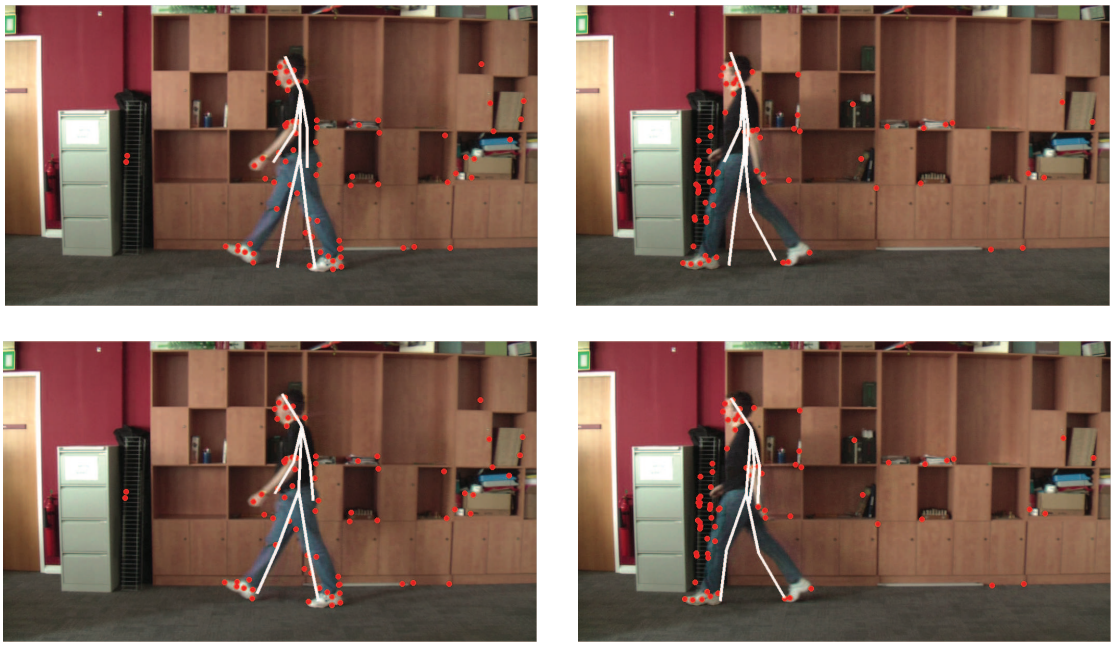


Figure 4.8: *Estimating pose using the MAP estimate and the effect of differing values of σ, two sample frames are shown using different value of sigma. $\sigma = 100$ (top row) prior has large influence. $\sigma = 10$ (bottom row) prior has less influence and each joint is forced to be located in close proximity to a foreground feature. Foreground features are plotted as red circles.*

To quantify the presented method for each sequence the location of each of the limbs was hand labeled to form a ground truth. As the right wrist and elbow could not be seen for most of the sequence these limbs were omitted from the ground truth. The error was calculated as the root mean square (rms) difference between the ground truth and the extracted limb positions averaged over all frames measured using the original image resolution of 1024×576 pixels.

The average error calculated over all sequences as a function of σ is shown in Figure 4.9. This shows that the error initially decreases as σ gets larger, this is because as σ becomes larger the prior becomes more dominant and is generally more reliant than the observations, however, there is a minima when $\sigma = 40$ and then the error gets larger with σ. At this value the balance is struck between the prior and the observations, the prior will deform but is not allowed to over fit the observations.

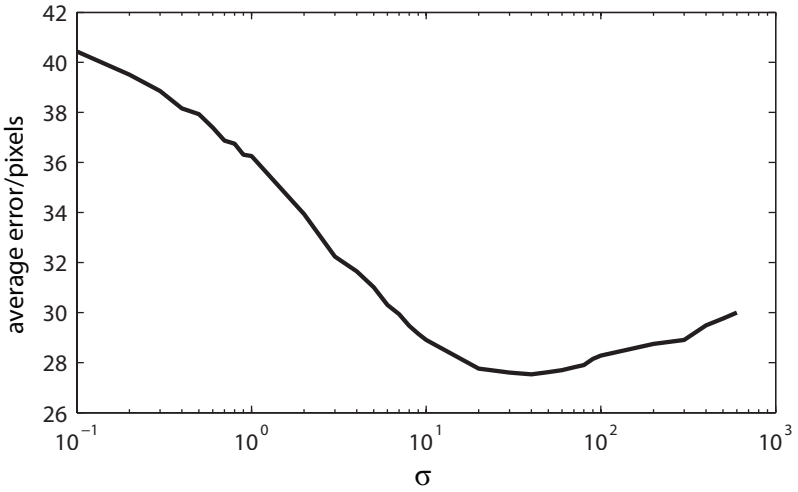


Figure 4.9: *Accuracy in estimating pose as a function of σ. The error is estimated over all ten sequences and all limbs (excluding right elbow, right arm) and is calculated as the root mean square difference between the extracted limb position and the hand labeled ground truth. A minimum error of 27.5 pixels is achieved when $\sigma = 40$.*

Quantitative results are presented in Table 4.1 for each subject and for each limb. These results show that for parts located nearer the root the error is generally lower than for those parts located further away. This is not just because errors will accumulate as you move away from the root node, but that in general limbs located further from the root would be expected to show more variation (e.g. the wrists and ankles). One of the problems with the presented approach is that there is no

temporal consistency across consecutive frames on any limbs other than the low-pass filter applied to the hip. It is this that is addressed in the next section.

subject	hip	shoulder	elbow	wrist	knee	ankle	head	mean
1	14.5	18.6	21.2	27.3	29.5	41.4	28.9	25.9
2	17.8	33.3	29.6	22.9	23.9	39.7	26.7	27.7
3	12.1	26.5	34.3	34.4	22.1	33.3	24.4	26.7
4	22.5	18.8	15.2	15.7	20.3	30.5	17.7	20.1
5	14.3	25.4	32.5	22.1	23.3	34.3	16.6	24.1
6	18.4	26.0	28.6	30.7	24.1	35.6	27.0	27.2
7	23.2	29.4	36.7	42.0	25.4	40.2	23.6	31.5
8	18.7	30.9	27.1	27.1	23.2	32.7	22.5	26.0
9	17.2	29.3	23.5	31.4	25.9	38.6	27.7	27.7
10	27.5	35.0	30.1	38.3	25.3	35.1	24.2	30.8
mean	18.6	27.3	27.9	29.2	24.3	36.1	23.9	26.8

Table 4.1: *Pose estimation rms errors for walking measured in pixels for each subject. The error presented for the knee and ankle is the average of both the left and right limb.*

4.5 Enforcing Temporal Coherence of Limbs

In this section high-level motion models are introduced to enforce the motion of each limb to be temporally coherent. A high-level motion model is represented by the change in angle between adjacent joints' position. As this is measured relative to the parent node's position, temporal searches can be performed separately for each joint. The purpose of the temporal search is to refine pose estimates from the previous section by making limb movements temporally coherent over the sequence of frames. The position of the root node is not of interest here since this was robustly estimated in the previous section. A low-pass filter is adequate to make the motion of the root node temporally coherent.

A high-level motion model is created for each joint except the root joint. This describes how a joint will move relative to it's parent as a function of phase, each model is defined by a set of angles that represent the expected motion between frames $\phi = (\phi_1, .., \phi_m)$, where m is the number of phases in the model.

The temporal search is also performed via Dynamic Programming, the method used

has many similarities with the HMM used in the previous chapter, except the probability density functions used to describe the state transitions are continuous and parameterized by a Von-Mises distribution over the change in angle expected to be observed across consecutive frames. The graph used for the temporal search consists of n vertices, where each vertex represents a frame of the sequence. The possible locations for a vertex now correspond to different angles. The observational data used for a joint $\mathcal{O} = \{\mathcal{O}_1, .., \mathcal{O}_n\}$ is the angle of that joint estimated in the previous section for each frame. The gait phase estimated using methods described in the previous chapter is also exploited, this sequence is defined as $S = \{S_1, ..S_n\}$. The temporal search is performed over the entire sequence using Equations 4.6-4.9, after first defining

$$p(l_i, l_j|c) = \mathcal{M}(l_i, l_j + \phi_{s_j}, \kappa_{s_j}) \tag{4.15}$$

and

$$p(\mathcal{O}_j|l_j) = \mathcal{M}(l_j, \mathcal{O}_j, \alpha\kappa_{s_j}) \tag{4.16}$$

The deformation term described by Equation 4.15 makes it most probable to move through the angle ϕ_{s_j} across consecutive frames. The observational likelihood is defined so that the probability of a particular location l_j is lower the further it is away from the observed angle $\mathcal{O}_j$. α is a constant that defines the weighting between observations and model. For a low value of α the model will dominate and a high value the observations. An example of the effect of using different values of α for estimating the angle of the knee joint are shown in Figure 4.10. When $\alpha = 1.0$ the motion model acts as a template which is deformed to fit the observations, notice in particular that the amplitude of the observed gait is maintained, this would not be achieved if for example a low-pass filter was used.

The temporal search is performed for each limb over every frame. To conduct the

temporal search a space spanning 2π radians is used, represented as 180 equally spaced discrete values. Whilst assuming an independent search can be performed for each limb may seem a somewhat crude approximation, our model is well enough constrained such that unlikely poses will not occur.

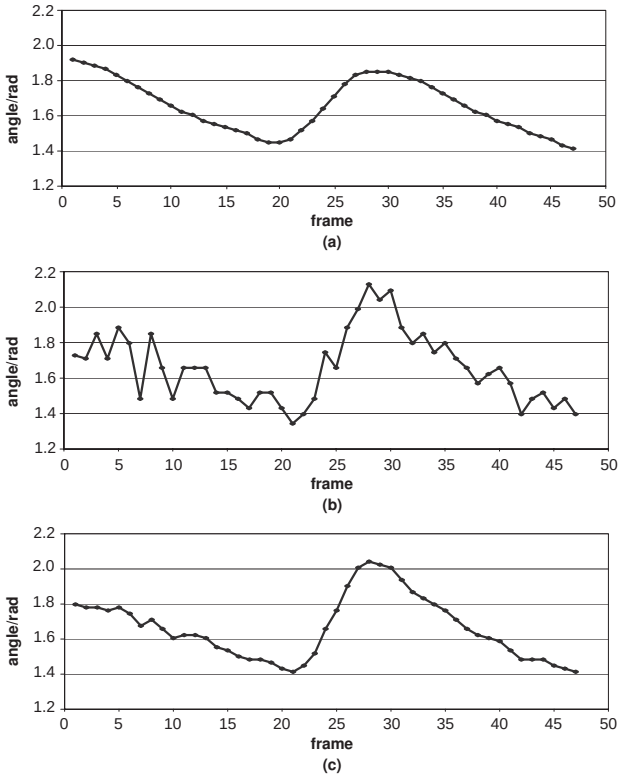


Figure 4.10: *Results of temporal search for knee joint using different values of α. (a) $\alpha = 0.001$, motion model dominates. (b) $\alpha = 1000$, observations dominate. (c) $\alpha = 1.0$, the model is deformed to the observations.*

4.6 Results

In Figure 4.11 the average error as a function of both α and σ is presented, these results have been constructed using the moving data set. As this graph shows a minimum error of 22.7 pixels is achieved when $\alpha = 0.05$ and $\sigma = 20$. In comparison to Figure 4.9 when a minimum of 27.5 pixels was achieved when $\sigma = 40$, this shows that when a motion model is being used to enforce temporal coherence the best results are achieved by first allowing the spatial model greater freedom to

first deform to the observations, then the high-level motion model will attempt to combine the observations in a way that temporally fits expectation.

Given that the smallest error is achieved with differing values of σ depending on whether a motion model is used indicates that the temporal and spatial search are not independent of one another. This is to be expected since the temporal search operates on the results of the spatial search, however, it is interesting that σ can not be optimized independent of α and vice versa to find the global minima.

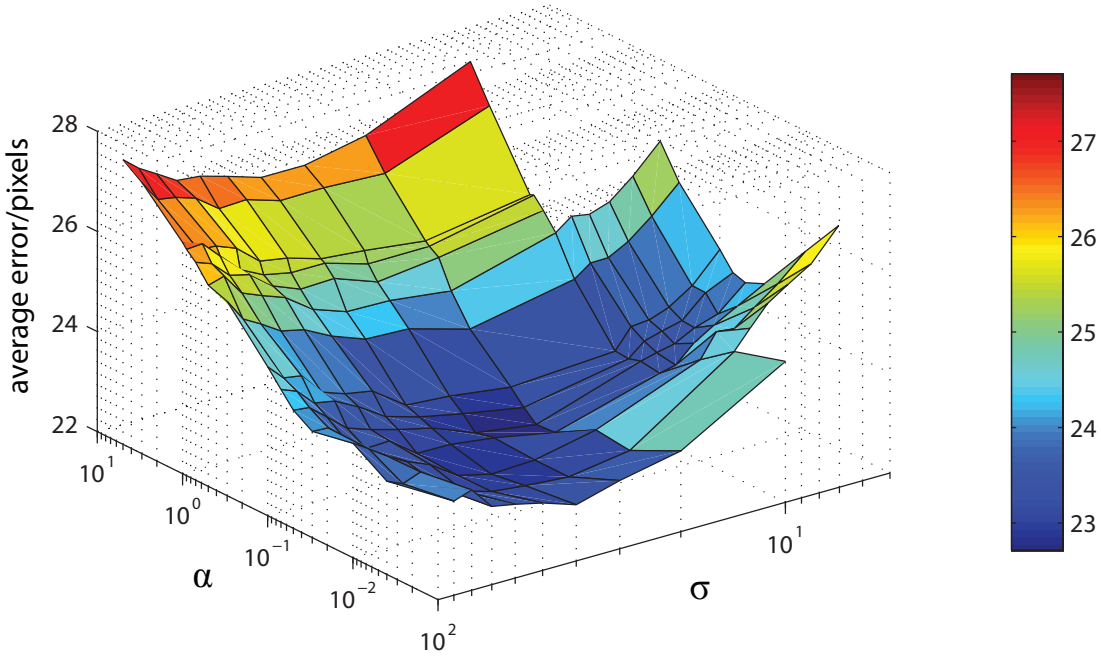


Figure 4.11: *Error surface as a function of α and σ using the walking data set. The minimum error of 22.7 pixels is achieved when $\alpha = 0.05$ and $\sigma = 20$.*

In Table 4.2 results are presented for the treadmill sequences and the moving sequences using the values of $\sigma = 0.05$ and $\alpha = 20$. These results show that a smaller error is achieved on the treadmill sequences compared to the moving sequences. This is most likely a result of the poorer feature tracking present in these sequences due to poor lighting, compression artefacts and background clutter.

The error on the root node acts as a good indicator as to the minimum error that can be expected to be achieved on any other parts, since if this measurement has a

large error it is then likely that other parts further down the tree will have at least as bad an error. The exception to this is in the estimation of the position of the head. Table 4.2 shows that on both the treadmill and the moving sequences the average error in estimating the head position is lower than that of the hip. There are two principal reasons for this. The first is that features tracking the head tend to be more consistent than on any other part of the body, since the head doesn't suffer any self occlusion and changes appearance little over an entire sequence. The second is that this part of the body is easier to create a ground truth for, it is very difficult to infer the position of the hip when creating a ground truth as it is hidden from view meaning that a proportion of tracking errors reported are likely to be errors in hand labelling, these errors are likely to be worse for parts that are hidden from view under clothes, e.g. the hip and knees.

	hip	shoulder	elbow	wrist	knee	ankle	head	mean
moving	18.3	23.0	20.1	21.4	20.5	31.8	17.0	22.7
treadmill	10.9	12.8	13.9	23.8	15.2	22.6	9.2	16.2
Fathi *et al.* [32]	15.1	17.5	23.1	30.0	13.2	15.0	N/A	19.0

Table 4.2: *Average pose estimation rms errors for walking measured in pixels for each set of sequences. The error presented for the knee and ankle is the average of both the left and right limb. The bottom row shows the average error presented by Fathi et al. [32] using Motion Exemplars, achieved on sequences of people walking on a treadmill.*

In Table 4.2 the results are also compared to those using motion exemplars from [32]. This work attempts to match learnt motion templates to sequences of images, by finding the best matching template for the observed motion. The pose of the person from which the template was learnt is known and can be used as an estimate of the pose in the frame being observed, a further localised search can then be performed to refine each estimate. A different template is used for the upper and lower half of the body. Whilst direct comparison is difficult as the two sets of results were obtained using different data sets, both consisted of a treadmill viewed from the side-on and the walkers are a similar height in pixels. For a similar sequence to those used in [32] the method presented here achieves more accurate pose estimation, however, for the moving walking sequences the error is slightly more than that presented in [32].

Comparing the results of the presented technique and those of [32] also shows another

interesting difference in the performance of each method. As the error for the method presented by [32] is fairly constant across the lower limbs this suggests the error is largely caused by a misalignment between the template and the sequence, this is different to our method presented where the error generally increases at limbs further from the root location.

Exemplar frames from the treadmill sequences with the extracted pose overlaid are shown in Figures 4.12 and 4.13. As can be seen there is generally good agreement between the pose of the person being observed and the extracted pose. In the bottom of Figure 4.12 it can be seen that often there are no features tracking the knees, however, despite this the approach still accurately infers their location.

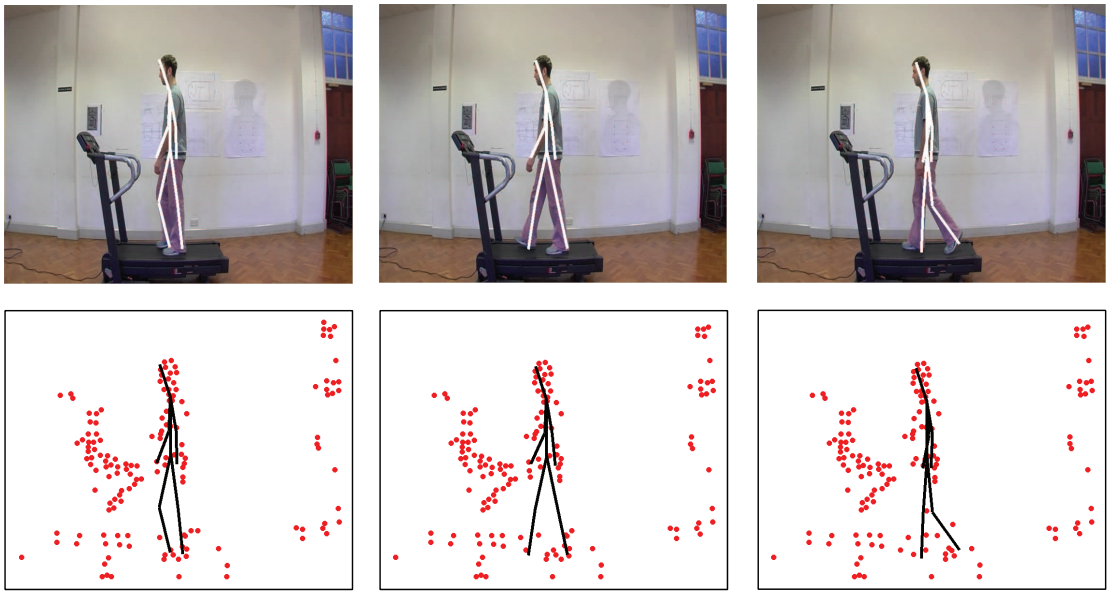


Figure 4.12: *Resultant estimated pose. (Top) sample frames from sequence with pose plotted. (Bottom) the corresponding observational data from which pose was estimated.*

In Figures 4.14 and 4.15 some example frames are shown for the moving sequences. In general the extracted poses and location of each of the limbs closely match that of the person shown in each frame. However, it can also be seen in the top left of Figure 4.14 that the extracted pose of the left leg is quite different to the true pose. This is perhaps a problem with using such sparse data, attempting pose extraction from noisy and sparse data will inevitably have some limitations.

In the top left of Figure 4.15 the pose shown is significantly different to that of the

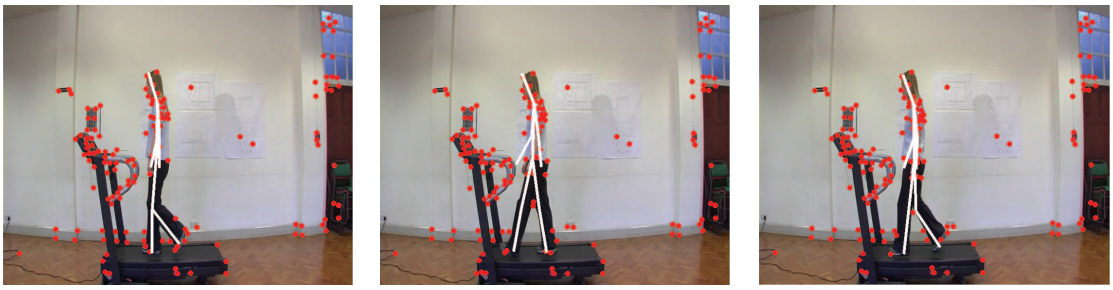


Figure 4.13: *Sample frames showing the resultant estimated pose. In each frame the resultant pose is plotted as are the positions of the tracked features.*

prior (for comparison the prior pose with maximum leg separation is depicted in Figure 4.2 (a)), in particular the legs of the subject are further apart. The more the prior is deformed the greater the cost, so as can be seen the prior finds features located on the inside edges of the subjects legs as this incurs a minimal deformation cost yet still achieves a good observational cost. Whilst the poses extracted are not exactly the same as the subject, this is largely as the model of a person walking on a treadmill does not well represent a person walking normally. However, the treadmill

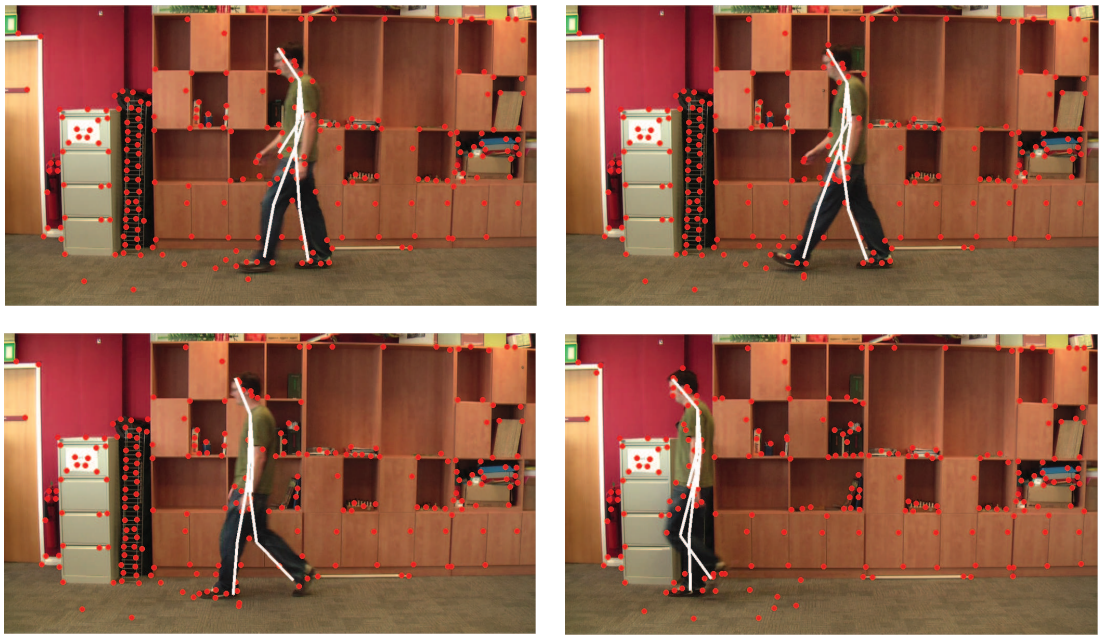


Figure 4.14: *Sample frames showing the resultant estimated pose. In each frame the resultant pose is plotted as are the positions of the tracked features.*

model has been used rather than learning a new model as it provides greater insight into how the approach operates and its limitations. The important point is that the poses extracted are unique, even though the data is both sparse and noisy the presented approach still attempts to extract new information.

To further demonstrate the presented approach a model of a quadruped was learnt from 6 complete gait cycles of a cheetah walking side on as shown in Figure 4.16. The model was then applied to a lion, without any further learning or tuning of parameters, as shown in Fig. 4.17. The lion's appearance and shape is significantly different to that of a cheetah. This sequence is also challenging since there is also a lot of clutter present, such as moving grass, and the colour of the lion is also similar to that of the background. It is unlikely that a binary silhouette could be extracted for this sequence and any approach that requires silhouette extraction would fail. However, the presented method is able to overcome all of these problems.

This example illustrates the strength of using only sparse motion features. Firstly the appearance is not modeled which allows similarly structured objects to be tracked using the same model. Secondly, the appearance of the object being tracked can be very similar to that of the background provided just a few sparse regions are distinguishable, these few sparse regions are sufficient to allow the position and pose of the entire object to be estimated.

4.7 Summary

In this chapter it has been shown how the structure of a sparse set of features can be exploited to estimate pose. The estimates of phase and likelihoods of a feature tracking a specific limb calculated using methods described in the previous chapter were used so that efficient searches could be performed via Dynamic Programming.

Given initial results a method was introduced to make pose estimates coherent across the entire sequence. This approach also used a DP solution and a separate search could be performed for each limb independently.

One of the limitations with the work presented in this chapter is that the models

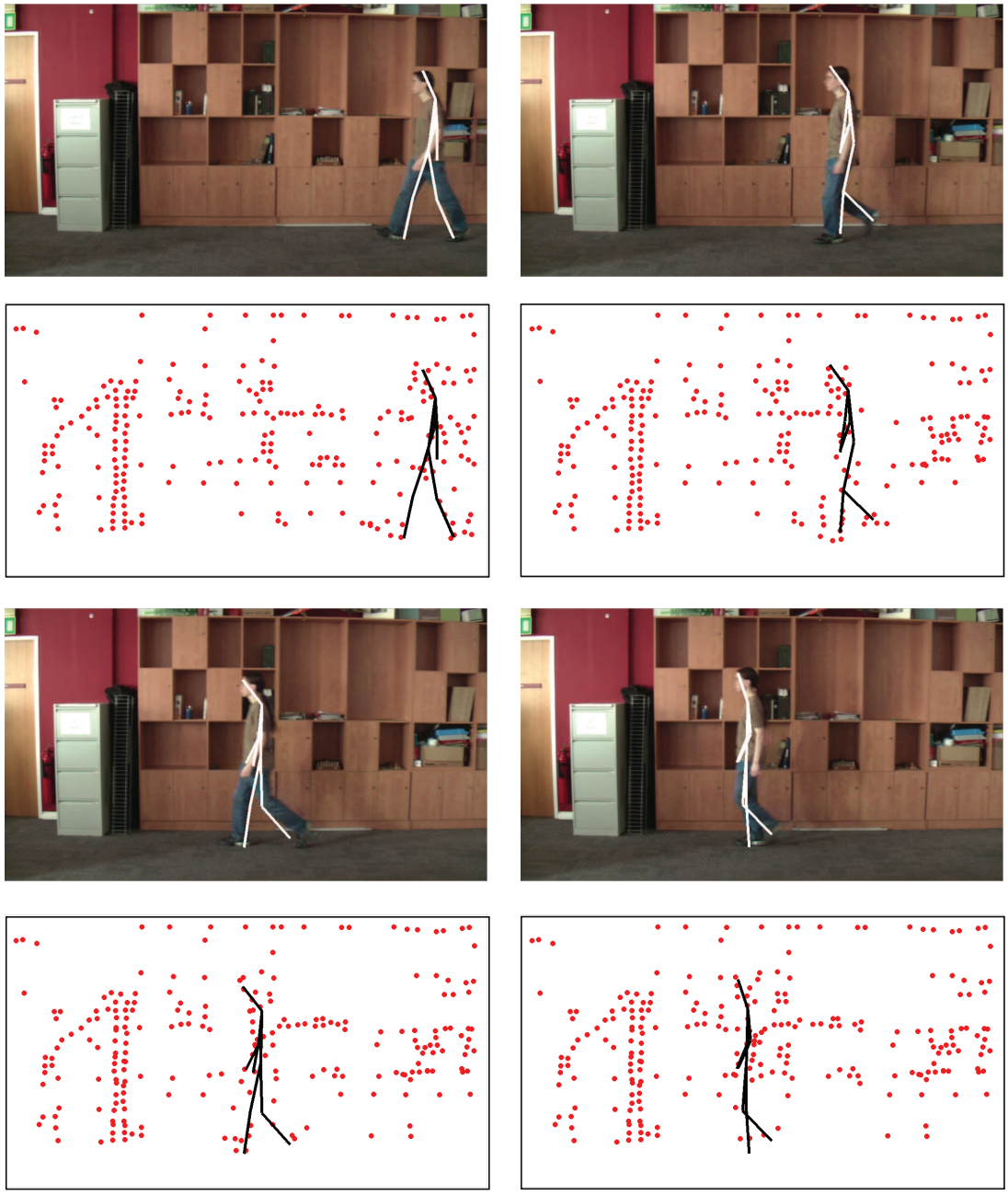


Figure 4.15: *Resultant estimated pose. (Top rows) sample frames from sequence with pose plotted. (Bottom rows) the corresponding observational data from which pose was estimated.*

were constrained to one viewpoint. It was expected that the subject being observed would be walking across the scene from right to left. This meant the dimensionality of the problem was significantly reduced and the problem simplified. Whilst in this

Figure 4.16: *Example frames from the sequence of a cheetah used to learn a model of a quadruped.*

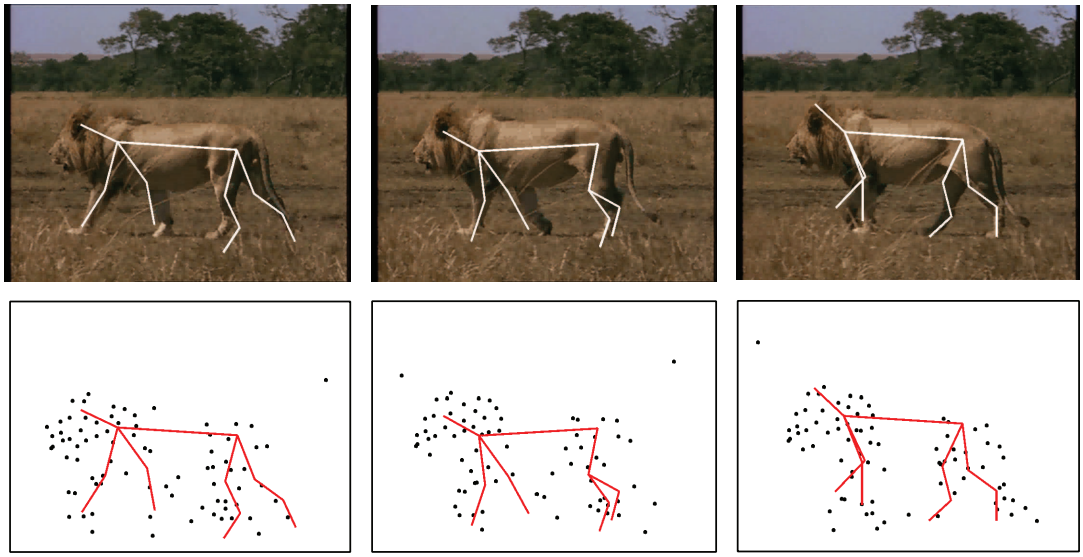


Figure 4.17: *Sample frames with estimated pose plotted. The top row shows the original images. The bottom row shows the set of sparse foreground features used.*

chapter it has been shown that the sparse set of features contain enough information to extract pose given the viewpoint and expected trajectory of the subject is known *a priori*, in the next chapter this assumption is relaxed methods are developed to extract 3D pose for people walking at unknown orientations to the camera.

Chapter 5

3D Pose Estimation

In this chapter the ideas presented in Chapters 3 and 4 are extended to three dimensions. This will not only allow pose to be estimated in 3D, but also to be extracted for people walking at arbitrary directions relative to the camera rather than just perpendicular to it. The extracted pose will also be measured in real world co-ordinates (mm) rather than image pixels. This will allow quantitative analysis to be performed using the HumanEva data set. This data set contains both image data and synchronized Motion Capture of three subjects filmed from seven different camera views.

The main difficulties with using three dimensions is that the search space is infinite. Whilst in the 2D case it seemed obvious that this should be limited within the image boundaries, in three dimensions assumptions must be made to constrain the search space. This is achieved by first estimating the trajectory that the person being observed moves across the ground plane. This will allow the search space to be reduced as the location of the person is approximately known in each frame. If it is assumed the person faces in the direction of travel, knowing their trajectory across the ground plane also provides their orientation.

Further problems are that all observations made in the image plane will have to be mapped into $\mathbb{R}^3$, whilst the geometry of cameras is well understood it will be found that this problem is often under constrained. Two main approaches will be used to

overcome this. Either heuristics will be used so that an exact solution can be found or the entire set of possible solutions will be exploited.

The main contribution of this chapter is to demonstrate that 3D pose can be extracted using just a sparse set of moving features as proposed by the thesis. The remains of this chapter are set out as follows. Firstly, the mathematical framework used for projective geometry is described, following which, a method is presented to estimate the location of the subject being tracked in each frame, using this information an approach is described to estimate 3D motions given 2D image observations. Finally, experimental results are presented.

5.1 Projective Geometry

In this section the mathematical framework used to describe projective geometry is introduced. Firstly, the intrinsic and extrinsic camera parameters are described and it is shown how these can be combined to create a single matrix that performs the projection of a point in $\mathbb{R}^3$ to a point in $\mathbb{R}^2$.

The intrinsic camera parameters describe the properties of a camera, these properties are measured independent of the camera's position or orientation in the real world. These can be understood in terms of a basic pinhole camera shown in Figure 5.1. The camera centre is the point that all incident rays will converge to and is the origin of the camera's coordinate system. The z-axis is normal to the image plane and is called the principal axis. Where this axis intercepts the image plane is called the principal point and represents the origin of the image in the camera's coordinate system. Only the focal length f determines where a point $\mathbf{X} = (X, Y, Z)$ is projected to in the image plane. However, often a new coordinate system will be used so that the origin of the image plane is not located in the middle of the image. The position of the principal point measured in the new coordinate system is $\mathbf{p} = (p_x, p_y)$. These parameters can be constructed into the camera calibration matrix K as

$$K = \begin{bmatrix} fm_x & 0 & m_x p_x \\ 0 & fm_y & m_y p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.1}$$

so that an inhomogeneous point in the camera's coordinate frame $\tilde{\mathbf{X}} = (X, Y, Z)$ can be projected onto the image plane through the transformation $\mathbf{x} = K\tilde{\mathbf{X}}$, where $\mathbf{x}$ is a homogenous vector $\mathbf{x} = (x/z, y/z, 1)$. The factors (m_x, m_y) are parameters to convert measurements from units of mm to units of pixels.

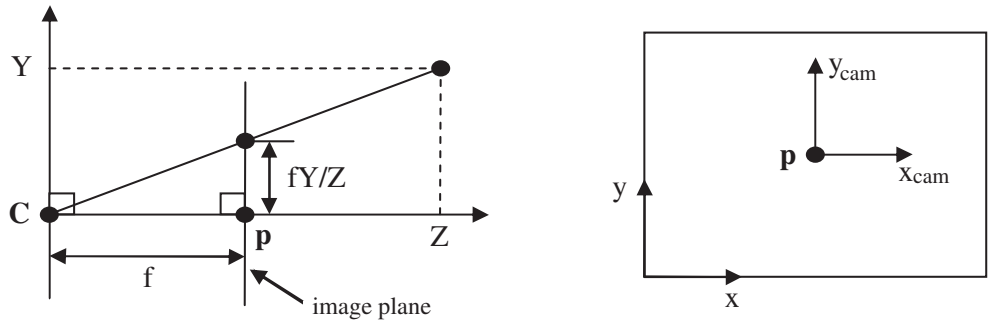


Figure 5.1: *Camera geometry. Cross section of a pinhole camera (left)* **C** *is the camera centre,* **p** *is the principal point where the z-axis (the principal ray) passes through the image plane. The image plane (right) showing the principal point in the camera's coordinate system and the image's. Reproduced from [48]*

In general the coordinate frame of the camera will not be the same as the coordinate frame of the real world so an alignment must be performed. In the experiments carried out in this chapter the real world coordinate frame will be that of the motion capture suite. A point measured in the real world must be mapped into the camera coordinate frame. Firstly, through a translation $-\tilde{\mathbf{C}}$, where $\tilde{\mathbf{C}}$ is the position of the camera centre measured in the real world coordinate frame. Following this a rotation $\mathbf{R}$ is performed to align the axis of the real world and camera. This translation and rotation represent the extrinsic camera parameters. Where $\tilde{\mathbf{C}}$ is an inhomogeneous 3-vector and $\mathbf{R}$ is a 3 by 3 rotation matrix. From this, a homogeneous point in the real world $\mathbf{X} = (X, Y, Z, 1)$ can be projected into the image plane through the equation

$$\mathbf{x} = KR[I| - \tilde{\mathbf{C}}]\mathbf{X} \tag{5.2}$$

where I is the 3 by 3 identity matrix. Defining the projection matrix as $P = KR[I| - \tilde{\mathbf{C}}]$ Equation 5.2 can be written as

$$\mathbf{x} = P\mathbf{X} \tag{5.3}$$

Whilst Equation 5.3 could have been introduced without explanation of its origins it will subsequently prove useful to know how the projection matrix P can be decomposed.

The translation $\tilde{\mathbf{C}}$ represents the position of the camera centre in real world coordinates, in homogeneous coordinates, this can be written as $\mathbf{C} = (\tilde{\mathbf{C}}, 1)$. By inspection of Equation 5.2 it can be seen that $\mathbf{C}$ is the right null space of P. Writing the camera projection matrix as $P = [M|\mathbf{p}_4]$ where M is a 3 by 3 non-singular matrix and $\mathbf{p}_4$ is the 4th column vector of P. The camera centre can then be calculated from the projection matrix as $\mathbf{C} = (-M^{-1}\mathbf{p}_4, 1)$.

A point observed in the image plane $\mathbf{x} = (x, y, 1)$ back projects as a ray passing through the camera centre in $\mathbb{R}^3$. This ray can be represented as the join between two points through which it is known to pass. One point is the camera centre $\mathbf{C}$ and the other known point is where the image point projects to the plane at infinity defined as $\mathbf{D} = (M^{-1}\mathbf{x}, 0)$. Given these two points a join between them can be written as

$$\mathbf{X}(\mu) = (1 - \mu)\begin{pmatrix} -M^{-1}\mathbf{p}_4 \\ 1 \end{pmatrix} + \mu\begin{pmatrix} M^{-1}\mathbf{x} \\ 0 \end{pmatrix} \tag{5.4}$$

To verify all points defined by Equation 5.4 project to $\mathbf{x}$ as expected, Equation 5.4 can be multiplied by P, hence

$$P\mathbf{X}(\mu) = (1-\mu)P\mathbf{C} + \mu P\mathbf{D} = \mu\mathbf{x} \tag{5.5}$$

where $\mu\mathbf{x} = \mathbf{x}$ in homogeneous coordinates. Since $P\mathbf{C} = 0$ the term $(1-\mu)$ can be disregarded, therefore the join can be written as $\mathbf{X}(\mu) = C + \mu D$ where D represents the gradient of the line. Equation 5.5 will frequently be used to define the space in $\mathbb{R}^3$ that a feature observed in the image plane could be located.

5.2 Ground Plane Trajectory Estimation

Estimating the trajectory a person walks across the ground plane is of importance for two reasons. Firstly, this information can be used to constrain the search space by providing an approximate location and orientation of the person in each frame. Secondly, this can be used to estimate the motion in $\mathbb{R}^3$ from an observed 2D motion in the image plane.

It is assumed that the ground plane is known and is defined as the xy plane $z = 0$. The unit vector normal to the ground plane is therefore $\hat{\mathbf{r}}_{gp} = (0, 0, 1)$.

The centre of a person is taken to be represented by the pelvis and this is used to define their position on the ground plane. Given the pelvis position $\mathbf{X}_{pel} = (X, Y, Z)$ their location on the ground plane is the X and Y component of $\mathbf{X}_{pel}$.

This can be calculated by first estimating the position of the pelvis in the image $\mathbf{x}_{pel}$. From this, the location in the ground plane can then be calculated by

$$\begin{pmatrix} X_{pel} \\ Y_{pel} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{p}_1 & \mathbf{p}_2 & Z_{pel}\mathbf{p}_3 + \mathbf{p}_4 \end{pmatrix}^{-1} \mathbf{x}_{pel} \tag{5.6}$$

where $\mathbf{p}_i$ represents the ith column vector of P. Equation 5.6 requires the Z coordinate of $\mathbf{X}_{pel}$ to be known, this is assumed to be half the walking height of the person being observed.

$\mathbf{x}_{pel}$ can be estimated in each frame by using a particle filter to track the foreground object with a bounding box as in Chapter 3. RANSAC is first applied to the motion of the features to segment them into those tracking the foreground and those tracking the background. Features tracking the background are discarded from further use.

The pelvis location is taken to be the mean of the distribution of particles, where each particle defines the centre of a bounding box measured in the image plane. The dimensions of the bounding box are set to be constant. Despite this the position of the pelvis is estimated robustly even though the bounding box could be significantly bigger than the object being tracked.

An example frame showing the segmented foreground and background features is shown in Figure 5.2. Also plotted is the bounding box defined by each of the particles. Whilst it can be seen that each individual particle would provide a very inaccurate estimation of the pelvis location, the average position of all particles is far more accurate.

As required by Equation 5.6 the walking height must be estimated in each frame. This is calculated by estimating the position in $\mathbb{R}^3$ of the foreground feature $\mathbf{x}_{max}$ with the largest y component measured in the image plane. Then, by performing the mapping $\mathbf{x}_{max} \rightarrow \mathbf{X}_{max}$, the height is assumed to be the Z component of $\mathbf{X}_{max}$.

However, this problem is under constrained since no component of $\mathbf{X}_{max}$ is known. To overcome this the foreground feature $\mathbf{x}_{min}$ with the minimum y component measured in the image plane can also be used. It is assumed that the lowest feature is in contact with the ground plane. Its position $\mathbf{X}_{min}$ can be estimated using Equa-

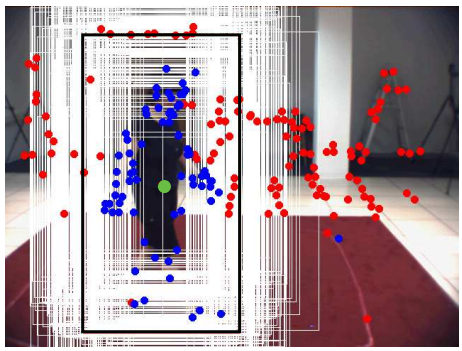


Figure 5.2: *Estimating the position of the hip. The red circles represent features classed as tracking the background and blue circles the foreground. The green circle shows the estimated pelvis position by taking the average of all the particles and the black box shows the bounding box position for this average. Each particle has also been plotted as a white box.*

tion 5.6 and setting $Z = 0$.

Given $\mathbf{X}_{min}$, $\mathbf{X}_{max}$ can be estimated by defining two lines. The first $\mathbf{X}(\mu) = \mathbf{X}_{min} + \mu\hat{\mathbf{r}}_{gp}$ defines a line perpendicular to the ground plane passing through $\mathbf{X}_{min}$ and the second $\mathbf{Y}(\lambda)$ defines the ray that projects to the feature $\mathbf{x}_{max}$ as defined by Equation 5.4. $\mathbf{X}_{max}$ can then be calculated as the location where the two lines are closest (i.e. $\min ||\mathbf{X}(\mu) - \mathbf{Y}(\lambda)||$).

Figure 5.3 shows an example of the extracted height compared against the ground truth. Notice as it is not the standing height being extracted the height is not constant. From this graph it is clear that there is good agreement between the extracted height and ground truth, particularly noticeable is that the frequency of both signals is clearly visible. However, the extracted height does occasionally drift and there are large spikes where no features were extracted on the lower limbs. To overcome these problems the average height is calculated over a complete sequence and used as the height in all frames. Tested on all subjects and camera views this was found to estimate the height with an error of $\pm 26.5mm$.

After the height of the subject had been estimated their position on the ground plane could be estimated in each frame. The resultant trajectory contained noise so that it wasn't smooth as would be expected. The position of the person changed significantly across consecutive frames, as a result the orientation, which is assumed to be the direction of motion, would also change significantly.

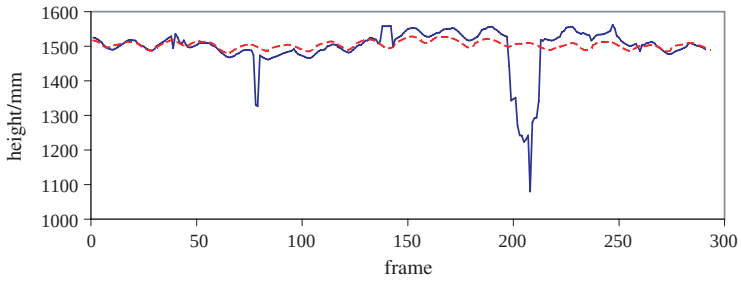


Figure 5.3: *Estimating walking height. The walking height estimated using the highest and lowest foreground feature (solid blue line) compared to the ground truth height (red dashed line).*

The accuracy of the estimated position is worse if the person being observed is at a greater depth. This is as the area encompassed by a single pixel in the image plane maps to a larger area on the ground plane as the ray projecting through the pixel becomes parallel to the surface of the ground plane. This effect is equivalent to mapping an error in the image plane onto the ground plane and is illustrated in Figure 5.4, where it is shown that an equal uncertainty in the image plane for two different positions can result in vastly different errors when mapped onto the ground plane.

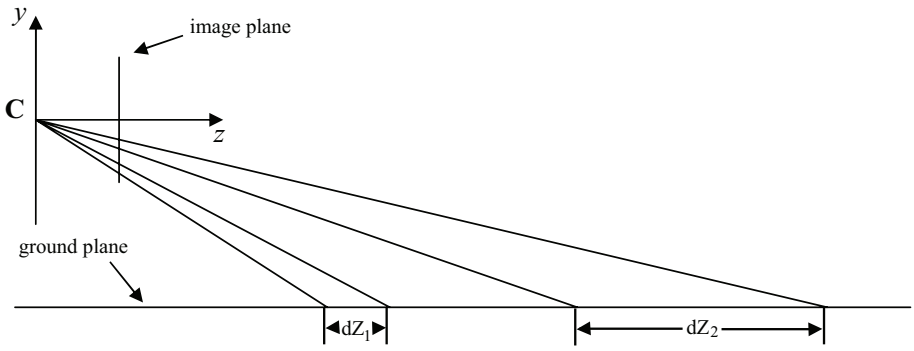


Figure 5.4: *The effect of projecting two errors that have the same magnitude in the image plane onto the ground plane. The resultant errors dZ_1 and dZ_2 are not equal.*

The uncertainty in the pelvis location measured in the image plane is assumed to be Gaussian $\mathcal{N}(\mathbf{x}_{pel}, \Sigma_{im})$. Where Σ_{im} is a 2 by 2 diagonal covariance matrix. It would be desirable to perform the transformation $\mathcal{N}(\mathbf{x}_{pel}, \Sigma_{im}) \rightarrow \mathcal{N}(\mathbf{X}_{gp}, \Sigma_{gp})$ so that errors in estimating the position of the pelvis in the image plane could be propagated into the ground plane. This can be achieved using the Unscented Transform [55] which provides a simple method to map a Gaussian distribution through a nonlinear

transformation, this is described below for a two dimensional Gaussian.

A set of 4 sigma points σ_i is calculated from the columns of the matrices $\sqrt{2\Sigma_{im}}$ and $-\sqrt{2\Sigma_{im}}$. These are then translated to have the same mean $\mathbf{x}_{pel}$ as the original distribution. Following this each of the sigma points are transformed onto the ground plane through Equation 5.6. From this set of transformed points the mean and covariance of the distribution $\mathcal{N}(\mathbf{X}_{gp}, \Sigma_{gp})$ can be calculated.

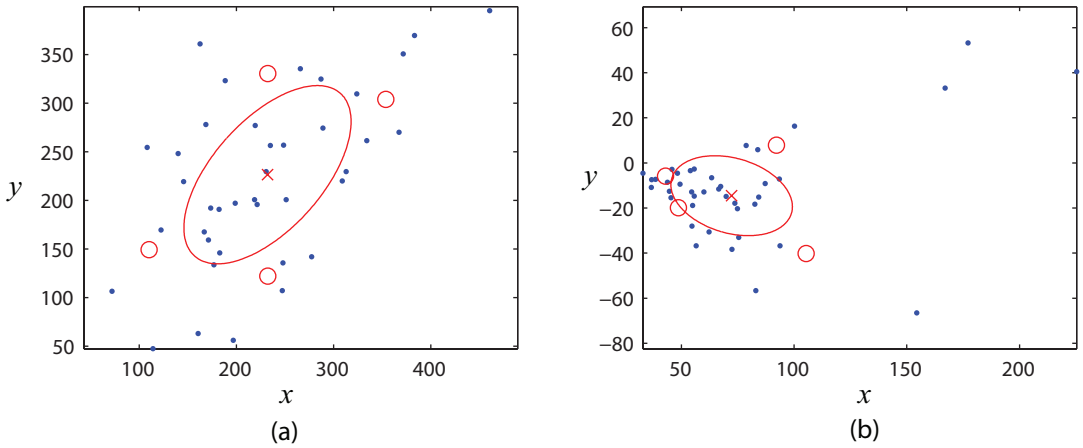


Figure 5.5: *Example of the Unscented Transform being applied to a random distribution of measurements (blue dots) transformed through a non-linear function. The small red circles show the sigma points. The ellipse in each figure shows the probability contour at σ of the Gaussian distribution and the red cross shows the mean. The original distribution is depicted in (a) and the transformed distribution in (b).*

An example showing the effect of the unscented transform is shown in Figure 5.5. Points drawn from a random distribution are shown in Figure 5.5 (a) along with the uncertainty contour 1σ and the mean. The sigma points are also shown as red circles. In Figure 5.5 (b) the transformed points are shown and the Normal distribution calculated from the transformed sigma points. Notice the transformed sigma points are no longer symmetric as in the original distribution.

Also Figure 5.5 illustrates an alternative to the Unscented Transform, which is to draw a random set of features from the original distribution, transform them through the non-linear function and then re-estimate the distribution from the transformed features. The problem with this is that it is very computationally expensive, many features must be drawn so that the mean and covariance of the random features is

the same as the distribution being approximated. All of these features must then be transformed and new parameters estimated. However, the Unscented Transform provides a method to deterministically select a set of features that exactly captures the covariance of the distribution that is being modeled. This produces a much more efficient solution.

In Figure 5.6 the use of the Unscented Transform is shown to estimate the error in the extracted ground plane trajectory. The covariance has been plotted in each frame and is shown on both the ground plane and projected into the image plane. It is assumed that the covariance matrix Σ_{im} is diagonal, this negates the need to use matrix decomposition to estimate the roots of $\sqrt{2\Sigma_{im}}$. Whilst in the image plane the error function appears to be similar when the person is close to the camera compared to when they are further away, the error as seen in the ground plane is far greater when they are further away than when they are closer.

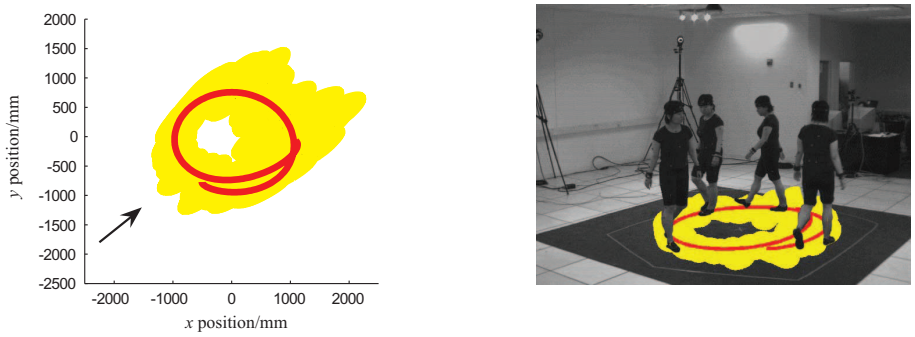


Figure 5.6: *Ground plane trajectory estimation. The trajectory extracted by fitting a polynomial to the original data is shown as the red solid line. The error function is shown as yellow. This error has been created by plotting, for each measurement, an ellipse, representative of one standard deviation of the covariance matrix. These have been shown in the ground plane (a) and the image plane (b). The direction that the ground plane is viewed from is shown in (a) by the arrow. In (b) the person being observed is shown from four different frames so that agreement between the extracted trajectory and actual trajectory can be seen.*

Once all the errors have been propagated into the ground plane they can be used for model fitting. Whilst an online tracker such as a Kalman filter could be used to estimate the trajectory across the ground plane, the approach used here is to fit a polynomial to the observed data. This is akin to the simple dynamic model used

in Chapter 3, however, whilst a first order model was suitable for estimating the position of a person walking across the scene side on to the camera, a higher order model must be used to capture their motion across the ground plane.

A polynomial can be fitted to the ground plane data using least squares. Least squares attempts to minimize the sum of the squares of the residuals

$$||r||^2 = \sum_{i=1}^{m} r_i^2 \tag{5.7}$$

where a residual is the difference between an observation and the model and m is the number of observations being used. Weighted least squares can be used to minimize a modified version of Equation 5.7 where each residual is weighted according to the accuracy in the observation

$$||r||^2 = \sum_{i=1}^{m} \frac{r_i^2}{\sigma_i} \tag{5.8}$$

Where σ_i is a measure of the accuracy of the ith observation [69]. This is set as the standard deviation of each measurement in the ground plane.

In Figure 5.7 some samples have been taken from the blue dashed line. Each sample has had random noise added, drawn from a Gaussian with standard deviation σ_i. Though the exact noise added is not known the standard deviation of each Gaussian used is, as shown by the error bars. The red line depicts the line of best fit using standard least squares and the green line shows the result using weighted least squares. As this graph shows the agreement is much better between the true line and the weighted best fit compared to the difference between the true line and the standard best fit. This is as the weighted best fit is unaffected by outlying points if they are known to have a large uncertainty.

An example of the extracted ground plane position along with the error function for

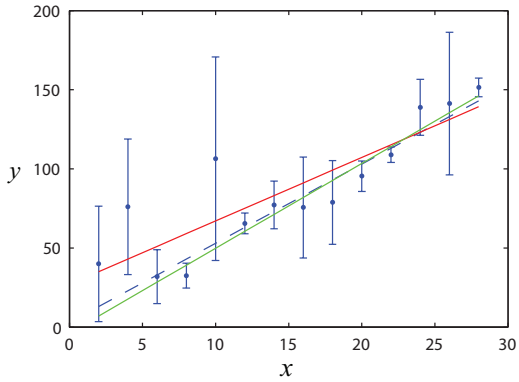


Figure 5.7: *Fitting a straight line to a set of samples drawn from a straight line (shown as the blue dashed line) with random noise added. The random noise is drawn from a Gaussian with standard deviation $\sigma(x)$ shown by the error bars on each point. The red line shows the best fit model using standard least squares and the green line shows the line of best fit using weighted least squares.*

each axis is shown in Figure 5.8. This shows a polynomial used with order 5. An example of an extracted trajectory is also shown in Figure 5.6.

In this Section the trajectory that the person being observed moves across the ground plane has been estimated. This not only provides the location of the person in each frame but also their orientation. Furthermore their height has also been extracted which will be used to scale any models used in subsequent sections. In the next section it will be shown that this information can be used to estimate the motion that has occurred in 3-space to produce the motion observed in the image plane.

5.3 Estimating 3D Motion from 2D Image Trajectories

In Chapter 3 the motion models represented the motion that would be observed if a feature was tracking a specific limb measured in the image plane. These models were two dimensional and had units of image pixels. However, in this chapter 3D motion models are used that have real world units (mm). The problem is that to use 3D motion models 3D motion observations are required. A motion observed in

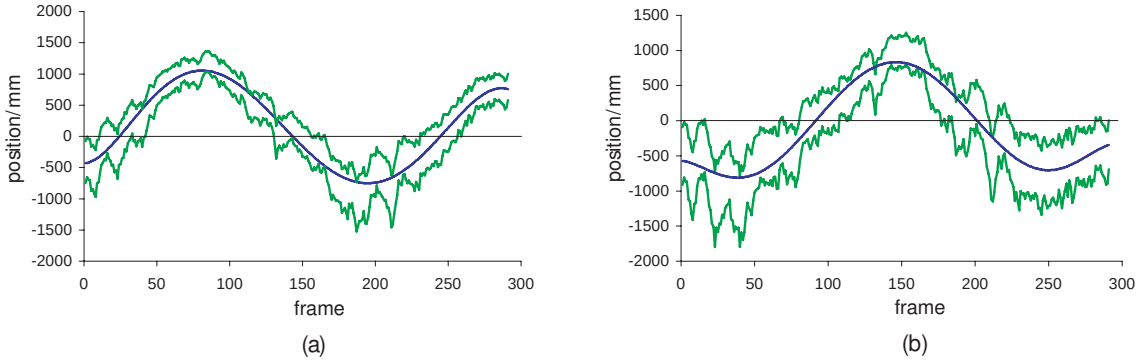


Figure 5.8: *Fitting a 5th order polynomial to the estimated ground plane trajectory in the x-axis (a) and y-axis (b). The resultant trajectory is shown by the blue line and the green lines represent the error function, corresponding to one standard deviation of the covariance.*

the image plane could be caused by an infinite number of possible motions in $\mathbb{R}^3$.

A further problem is that given a 3D motion observation it is not immediately clear how this should be orientated before being compared to a motion model. This is as the coordinate system of the motion model will be different to that of the motion capture system and the transformation between the two is unknown. A possible solution would be to compare the motion at all orientations, however, this would be expensive and all available observations would have to be used to form a consensus about the transformation between the models and real world coordinate system. Another solution would be to simply use the magnitude of the motion rather than the direction, this would greatly simplify the problem but at an expense that the model is likely to be less discriminative.

However, as the motion models use only motion, the absolute position of a feature is not of interest. Therefore only a rotation is needed so that the axes of the model and the real world are aligned. This rotation is set to align the axis of the models with the orientation of the person as estimated in the previous section. The person's estimated position and orientation can be used to define a different coordinate system. This coordinate system is defined so that the x-axis is aligned with the direction the person is facing, the y-axis runs across the shoulders and the z-axis is the same as in the real world coordinate system, the height off the ground plane. This coordinate system will be referred to as the subject's frame of reference. An observed motion can then be rotated into this frame of reference and be compared

directly to the motion models used.

Estimating the 3D motion that caused an observed 2D motion in the image plane can be achieved by assuming that the dominant motion as a person walks occurs in the direction of travel. Given the known location of a person on the ground plane $\mathbf{A}$ and their velocity $\mathbf{V}$, a plane can be defined as

$$\pi(\mu, \lambda) = \mathbf{A} + \mu\mathbf{V} + \lambda\hat{\mathbf{r}}_{gp} \tag{5.9}$$

where $\hat{\mathbf{r}}_{gp}$ is the normal to the ground plane. Given two observations of the same feature across consecutive frames $\mathbf{x}_t$ and $\mathbf{x}_{t+1} = \mathbf{x}_t + d\mathbf{x}_t$. Projecting a ray through each of these points, the position in 3-space of each feature $\mathbf{X}_t$ and $\mathbf{X}_{t+1}$, can be estimated by finding the intersection of the ray and the plane defined in Equation 5.9. The motion in $\mathbb{R}^3$ of these two features can then be calculated. This approach is illustrated in Figure 5.9. Notice that any motion extracted will have no y component in the subject's coordinate system.

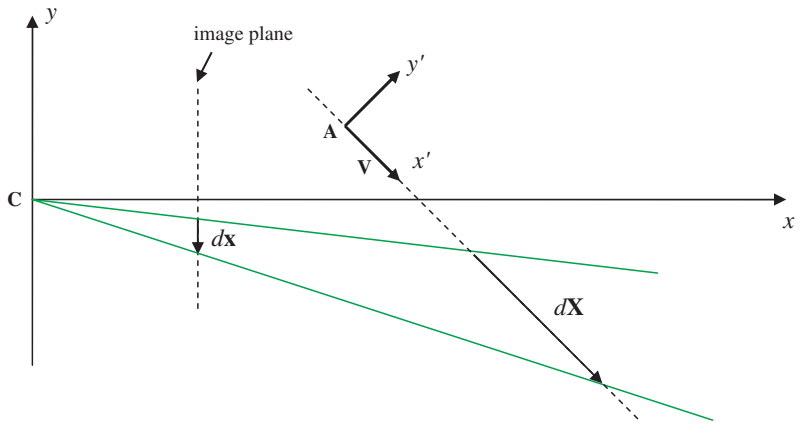


Figure 5.9: *Estimating the motion $d\mathbf{X}$ in $\mathbb{R}^3$ from a motion $d\mathbf{x}$ measured in the image plane. The figure shows a view looking down the z-axis onto the ground plane. $\mathbf{A}$ is the estimated location of the person in the ground plane and $\mathbf{V}$ is their velocity. The axes $\mathbf{x}'$ and $\mathbf{y}'$ show the coordinate frame of the subject. Notice the direction of $\mathbf{x}'$ coincides with that of $\mathbf{V}$ as defined in the text.*

Given the unit vector $\hat{\mathbf{D}}(\mathbf{x}_t)$ that defines the direction of the ray projecting through the feature $\mathbf{x}_t$ in the image plane and the unit vector $\hat{\mathbf{V}}$ which represents the orientation of the person in the ground plane. The accuracy of the approximation used to estimate the motion will be dependent on the scalar product $\hat{\mathbf{D}}(\mathbf{x}).\hat{\mathbf{V}}$. In Figure 5.10 the average motion error is shown by using our approximation applied to the HumanEva data set.

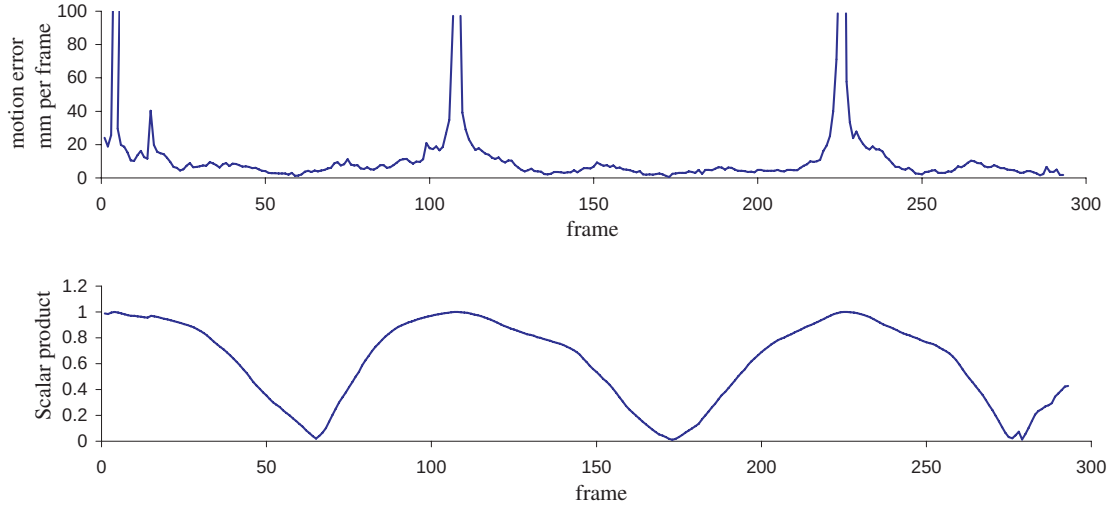


Figure 5.10: *(top) The error in estimating motion by assuming the dominant motion takes place in the direction of travel. (bottom) The corresponding average scalar product* $\hat{\mathbf{D}}(\mathbf{x}).\hat{\mathbf{V}}$.

As this figure shows the error deteriorates as the scalar product approaches unity. This is as any motion in the x-axis of the image plane will then be projected as a very large motion in the plane of travel defined by Equation 5.9. This can be further shown in Figure 5.11 where the average error is plotted as a function of the scalar product $\hat{\mathbf{D}}(\mathbf{x}).\hat{\mathbf{V}}$.

From Figure 5.11 it is clear that the error gets very large when the scalar product is above approximately 0.95, therefore when the scalar product is above this value the motion data is regarded too unreliable to be used and the feature has an equal likelihood of tracking any of the limbs or being in any of the phases. The result is that for a few frames of each sequence there is no observational motion data.

Once the motion has been estimated for each feature in 3-space the global motion of the person V must be subtracted so that the motion is estimated as if the person

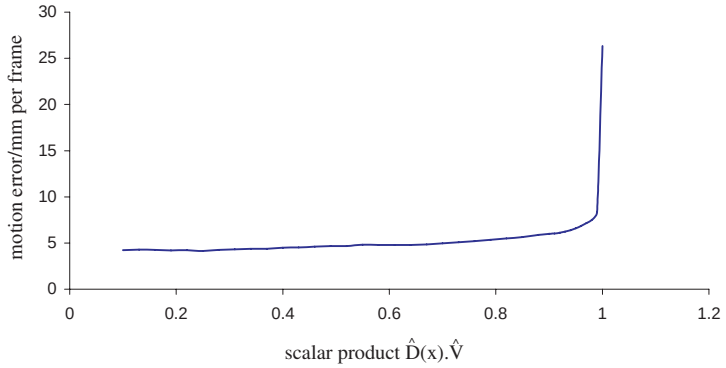


Figure 5.11: *The error in estimating motion by assuming the dominant motion takes place in the direction of travel. This is shown as a function of* $\hat{\mathbf{D}}(\mathbf{x}).\hat{\mathbf{V}}$.

is walking on a treadmill. Following this the likelihood of a feature tracking a particular limb can be estimated along with the phase of the motion using the same technique as described in Chapter 3 except the motion models now used will be three dimensional.

In this section a method has been presented to extract 3D motion from 2D image observations. This has been achieved by assuming the dominant motion occurs in the direction of travel. The accuracy of this assumption has been estimated using ground truth data.

5.4 Pose Extraction

In this section a method is described to estimate pose in each frame independently using just a sparse set of motion features. The inference method presented, whilst still able to exploit the tree structure of the human model, is different to that used in Chapter 4 and designed to integrate over more information. This is as the search space in $\mathbb{R}^3$ is larger than that used in the image plane. The results using this method are stable enough across consecutive frames that the use of high level motion models to enforce temporal coherence as used in Chapter 4 are not needed.

Typically generative methods used to extract 3D pose require hypotheses to be

projected into the image plane. Once projected the goodness of fit can then be calculated between each hypothesis and the observations. This process can be very computationally expensive as often thousands of hypotheses must be projected to extract the pose for a single frame. The presented approach does not require models to be projected into the image plane since it attempts to first map observations in the image plane into $\mathbb{R}^3$ before any search is performed. The result is that all searches can be performed directly in $\mathbb{R}^3$ and no reprojections are required, making it more computationally efficient compared to those that are required to project many hypotheses.

The 3D model used to represent a human is constructed of 15 key joints. Each joint is represented as a Gaussian, where the mean is the location that the joint is expected to be relative to the parent joint from which it is attached and the covariance represents the variation that is expected in the joint's relative position. For the root joint, the mean represents the expected height measured relative to the ground plane. A different set of parameters are learnt for each phase of gait, so each phase has a different representative pose. These models act as a prior over the configuration expected to be observed.

The same notation is followed from Chapter 4. The model is represented by a graph $G = (V, E)$, where $V = \{v_1, .., v_n\}$ defines the set of n vertices, and $(v_i, v_j) \in E$ define the set of edges connecting the vertices. The vertices represent the joints of the articulated object and the edges represent the dependence between connected joints. A particular configuration of this graph can be described by $\mathbf{L} = \{\mathbf{l}_1, .., \mathbf{l}_n\}$, where $\mathbf{l}_i$ is a vector that specifies the 3D location of v_i. The probability of making the observation $\mathcal{O}$ given the part v_i is placed at location $\mathbf{l}_i$ is defined as $p(\mathcal{O}|\mathbf{l}_i)$. The set of edges represent the dependence between connected vertices, where $p(\mathbf{l}_i, \mathbf{l}_j|c_{ij})$ is the probability of placing v_i at $\mathbf{l}_i$ and v_j at $\mathbf{l}_j$ given the connection parameters c_{ij}, which represents the model prior and are the parameters of a Gaussian distribution $\{\mu_{ij}, \mathbf{\Sigma}_{ij}\}$ so that $p(\mathbf{l}_i, \mathbf{l}_j|c_{ij}) = \mathcal{N}(\mathbf{l}_i - \mathbf{l}_j, \mu_{ij}, \mathbf{\Sigma}_{ij})$. The probability of a specific configuration is calculated as

$$P(\mathbf{L}|\mathcal{O}) \propto \prod_{i=1}^{n} p(\mathcal{O}|\mathbf{l}_i) \prod_{(v_i,v_j)\in E} p(\mathbf{l}_i, \mathbf{l}_j|c_{ij}) \tag{5.10}$$

Where the first term on the right hand side represents the observational probability and the second term represents the model prior. Whilst in Chapter 4 Dynamic Programming was used to efficiently calculate the Maximum a Posterior (MAP) of Equation 5.10, here the posterior distribution of the root node $P(\mathbf{l}_r|\mathcal{O})$ is calculated. From this the position of the hip can be extracted using the expectation value of this distribution. The position of each limb can then be estimated by calculating the probable location of that limb conditioned on the selected position of the parent limb. This approach is employed rather than using the MAP estimate as this method integrates information over a much larger region than the MAP, which will just use the observations recovered at the optimal n locations. The tree structure of the model representing a human can still be exploited to efficiently calculate the posterior as described in [35].

The search is performed over a 3D grid $\mathcal{G}$ that is defined to represent a volume in $\mathbb{R}^3$. The algorithm starts at the leaves of the tree and works towards the root node calculating at each step

$$S_j(\mathbf{l}_i) \propto \sum_{\mathbf{l_j} \in \mathcal{G}} \left(p(\mathcal{O}|\mathbf{l}_j) p(\mathbf{l}_i, \mathbf{l}_j|c_{ij}) \prod_{v_c \in C_j} S_c(\mathbf{l}_j) \right) \tag{5.11}$$

Where $v_c \in C_j$ are the children of v_j. The posterior distribution of the root node can then be calculated by

$$P(\mathbf{l}_r|\mathcal{O}) \propto p(\mathcal{O}|\mathbf{l}_r) \prod_{v_c \in C_r} S_c(\mathbf{l}_r) \tag{5.12}$$

The position of the root node can then be calculated using the expectation of the distribution $P(\mathbf{l}_r|\mathcal{O})$. Following this, the approach is to now work back down the tree structure calculating the positions of each subsequent limb. This is achieved by calculating the probability distribution of each limb $\mathbf{l}_j$ conditioned on the selected location of its parent limb $\mathbf{l}_i$ given as

$$p(\mathbf{l}_j|\mathbf{l}_i, \mathcal{O}) \propto p(\mathcal{O}|\mathbf{l}_j)p(\mathbf{l}_i, \mathbf{l}_j|c_{ij}) \prod_{v_c \in C_j} S_c(\mathbf{l}_j) \tag{5.13}$$

Note that the functions $S_j(\mathbf{l}_i)$ will remain unchanged and therefore does not need to be recalculated when traversing back down the tree from the root node. The expectation value of this subsequent probability distribution $p(\mathbf{l}_j|\mathbf{l}_i, \mathcal{O})$ can then be calculated before moving onto the next child node until the position of all limbs have been calculated. The importance of this approach compared to those such as Dynamic Programming or Min-Sum Belief Propagation is the summation over all grid locations contained in Equation 5.11, the consequence of which is that the location of each limb is estimated using all available information. One of the further benefits of calculating the expectation value for each part is that, despite the search being performed on a grid, since the location for each joint is calculated as a weighted mean the result can have sub grid accuracy.

The volume that contains the 3D grid $\mathcal{G}$ over which the search will be performed must be defined. To narrow the search space a separate but equal sized grid is defined for each limb. The positions in $\mathbb{R}^3$ of each grid is defined as $\mathcal{G}_j = \mathcal{G}_i + \mu_{ij}$ where i is the parent of j and the connection parameter μ_{ij} is the average position of limb j relative to i as defined previously. The absolute position and orientation of the root node in the ground plane is defined by those values estimated in Section 5.2, an example of the projected search space is shown in Figure 5.12 for the right foot and left wrist.

As each grid location is already centered at the mean expected position of that limb relative to the parent's location, the functions $S_j(\mathbf{l}_i)$ can be efficiently calculated as a Gaussian convolution

$$S_j(\mathbf{l}_i) \propto \mathbf{G} \otimes \left(p(\mathcal{O}|\mathbf{l}_j) \prod_{v_c \in C_j} S_c(\mathbf{l}_j) \right) \tag{5.14}$$

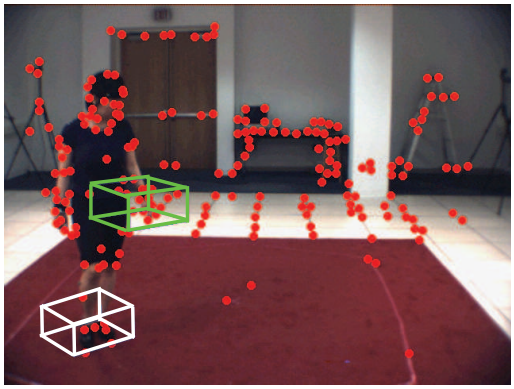


Figure 5.12: *Defining the search volume - the projected search space for the right foot (white) and left wrist (green) are shown.*

where $\mathbf{G}$ is a Gaussian filter with diagonal covariance defined by the connection parameter $\mathbf{\Sigma}_{ij}$ [35]. This is illustrated in Figure 5.13 which shows intuitively how Equation 5.14 can be calculated as a Gaussian convolution.

Before any of the above can be calculated the probability distribution $p(\mathcal{O}|\mathbf{l}_j)$ must be defined. As in Chapter 4 this is constructed from the likelihoods of each feature tracking a specific limb calculated using the feature's motion.

Firstly, the subset of features that lie in the projected search space for each limb are selected. It is then assumed that the true 3D location of each feature could be equally probable at any position along the ray passing through the camera centre that projects to the location of the feature in the image plane. At every position that the ray passes through the probability $p(\mathcal{O}|\mathbf{l}_j)$ is defined as the probability of the feature tracking the jth limb.

The rays pass through a very small volume of $\mathcal{G}_j$ meaning that the probabilities at grid location through which the ray didn't pass must be defined. For this the distance transform is used as in Chapter 4. An example of a resultant probability volume is shown in Figure 5.14 from the original view (a) and an alternate view (b), darker regions represent those with a higher likelihood.

After each distribution $p(\mathcal{O}|\mathbf{l}_j)$ has been calculated it is smoothed and all values are forced in the range $1 \geq p(\mathcal{O}|\mathbf{l}_j) \geq e^{-\rho}$ using

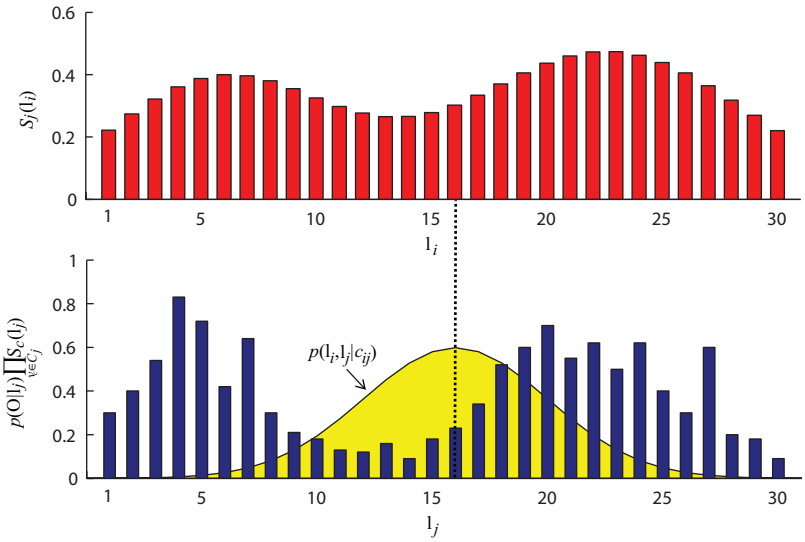
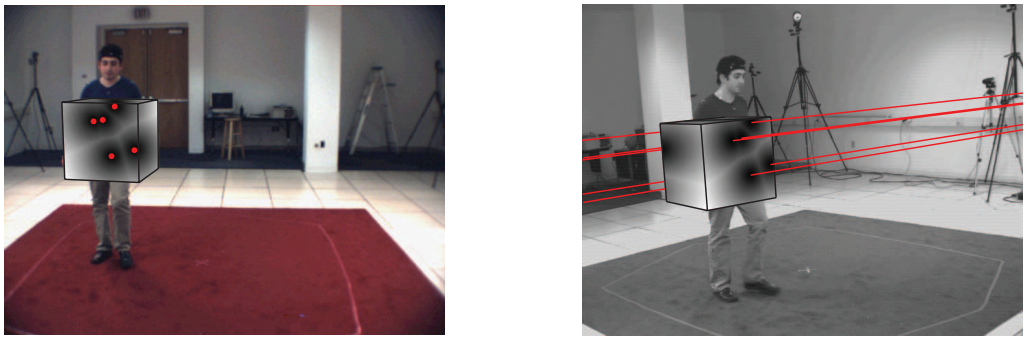


Figure 5.13: *Calculating the function $S_j(\mathbf{l}_i)$. The top shows the resultant function $S_j(\mathbf{l}_i)$. On the bottom is the function $\left(p(\mathcal{O}|\mathbf{l}_j)\prod_{v_c\in C_j} S_c(\mathbf{l}_j)\right)$ shown in blue and the Gaussian function $p(\mathbf{l}_i,\mathbf{l}_j|c_{ij})$ shown positioned to calculate $S_j(\mathbf{l}_i = 16)$.*



(a) (b)

Figure 5.14: *Example of a probability volume $p(\mathcal{O}|\mathbf{l}_j)$. (a) The original image with the projected probability volume, the tracked motion features are shown as red points. (b) The probability volume viewed from a different angle. The features in (a) are represented as rays in (b)*

$$p'(\mathcal{O}|\mathbf{l}_j) = \left(\frac{p(\mathcal{O}|\mathbf{l}_j)}{p(\mathcal{O}|\mathbf{l}_j)_{max}} \right)^{\gamma} \tag{5.15}$$

where γ is defined as

$$\gamma = \frac{\rho}{\log p(\mathcal{O}|\mathbf{l}_j)_{max} - \log p(\mathcal{O}|\mathbf{l}_j)_{min}} \tag{5.16}$$

and $p(\mathcal{O}|\mathbf{l}_j)_{max}$ and $p(\mathcal{O}|\mathbf{l}_j)_{min}$ are the maximum and minimum values of the distribution $p(\mathcal{O}|\mathbf{l}_j)$, ρ represents the order of magnitude required between the maximum and minimum limits of the distribution, in all of the experiments this parameter is set to 10. This smoothing is performed firstly to ensure the distributions are not too sharply peaked but also to ensure that the product of the probability distributions required in the computation of Equations 5.11 and 5.12 does not result in values smaller than the precision of the system.

A probability volume is created using this method for each joint except the root. The probability volume for the root joint acts as a prior for the location of the hip above the ground plane. This is calculated using Equation 4.12 but the subset $\mathcal{B} \subset \mathcal{G}_r$ of the grid is defined to be those locations that are at the expected height of the root above the ground plane, the probability at these locations is set to 1. Following this pose can be extracted for each frame independently.

In this section a method has been presented to estimate pose in each frame independently. The bottom-up search is performed in $\mathbb{R}^3$ by first defining a search volume for each limb in 3-space and then back tracing observations made in the image plane through these volumes. This allows pose to be estimated efficiently since all observations have been mapped back into $\mathbb{R}^3$. In the next section experimental results are provided using the methods outlined in this and the previous sections.

5.5 Experiments

In this section experimental results are presented that show the described approach is capable of achieving quantitative results comparable to the state of the art. The HumanEva data set, used for both learning and testing, is described and learnt models are also presented. Separate models are learnt for jogging and walking and the presented approach is shown to be capable of accurately estimating the 3D pose of a person performing either of these tasks.

5.5.1 HumanEva Dataset

The HumanEva data set [90] is a publicly available data set that consists of 4 subjects performing different gaited and non-gaited actions. Each action is filmed simultaneously from several different views and motion capture data is also recorded for each action. The cameras and motion capture suite are all temporally synchronized. The seven cameras consist of three colour cameras and four black and white, these will be referred to as C1, C2, C3 and BW1, BW2, BW3, BW4 respectively. The intrinsic and extrinsic camera parameters are provided for each camera.

Each sequence is filmed at a frame rate of between 60Hz and 120Hz, however, all video used in the presented experiments is down sampled to 30Hz to keep the temporal resolution consistent with sequences used in Chapters 3 and 4.

The data set is split into three partitions, training, validation and testing. The motion capture data for the testing partition is withheld and an online validation system is provided to obtain quantitative results for this subset. All of the presented results will use the validation partition of the data set for testing. As the presented method is monocular it can be tested on each sequence, filmed from each camera independently. As a validation set is not provided for Subject 4, only the first three subjects are used, their physical appearance is shown in Figure 5.15. Using this subset of data will allow the presented approach to be tested on 21 different sequences for each action. Each subject will be referred to as S1, S2 and S3.

For training purposes, only motion capture data is required as no appearance models

Figure 5.15: *Subjects contained in HumanEva dataset.(a) S1. (b) S2. (c) S3.*

need to be learnt. Models are learnt from the training partition of the data set.

All errors are calculated as the average Euclidian distance between the ground truth and extracted poses. All pose estimation errors will be measured relative to the pelvis location. This is so that results from the presented approach can be compared to published results using other methods.

5.5.2 Model Learning

Models were learnt from the training partition of the HumanEva data set that is composed of only motion capture data. The motion capture data contains some noisy measurements and for some parts of the sequences limbs were not tracked at all. For each subject a portion of each test sequence was selected where all of the main limbs appeared to be accurately tracked. In total 1150 frames of data were used to learn all models this corresponds to about 40 seconds of training data.

In each of the training sequences the subject walked in a circle. Before models could be learnt the training data had to be adjusted so that each subject was walking as if on a treadmill. This was achieved by estimating the trajectory of the person across the ground plane corresponding to the location of the pelvis. Then any global motion or changes in orientation could be compensated for. Each of the subjects was aligned such that the x-axis pointed along the direction of travel, the y-axis lied across the shoulders and the z-axis represented the vertical.

The training data of each subject was further scaled by the average walker height,

this normalised the data so that the average height across each training sequence was unity. This was performed so that large uncertainties wouldn't be present as a result of walker height, rather than as a result of variations in the action being performed.

Following this each sequence of ground truth was sliced into individual gait cycle examples as in Chapter 3. In total 31 complete gait examples were used to learn models. The average gait cycle length was 38 frames, therefore the resultant models contained the same number of phases. The same selection of motion models are shown in Figure 5.16 as were presented in Chapter 3. In total 15 motion models were learnt including separate models for left and right limbs.

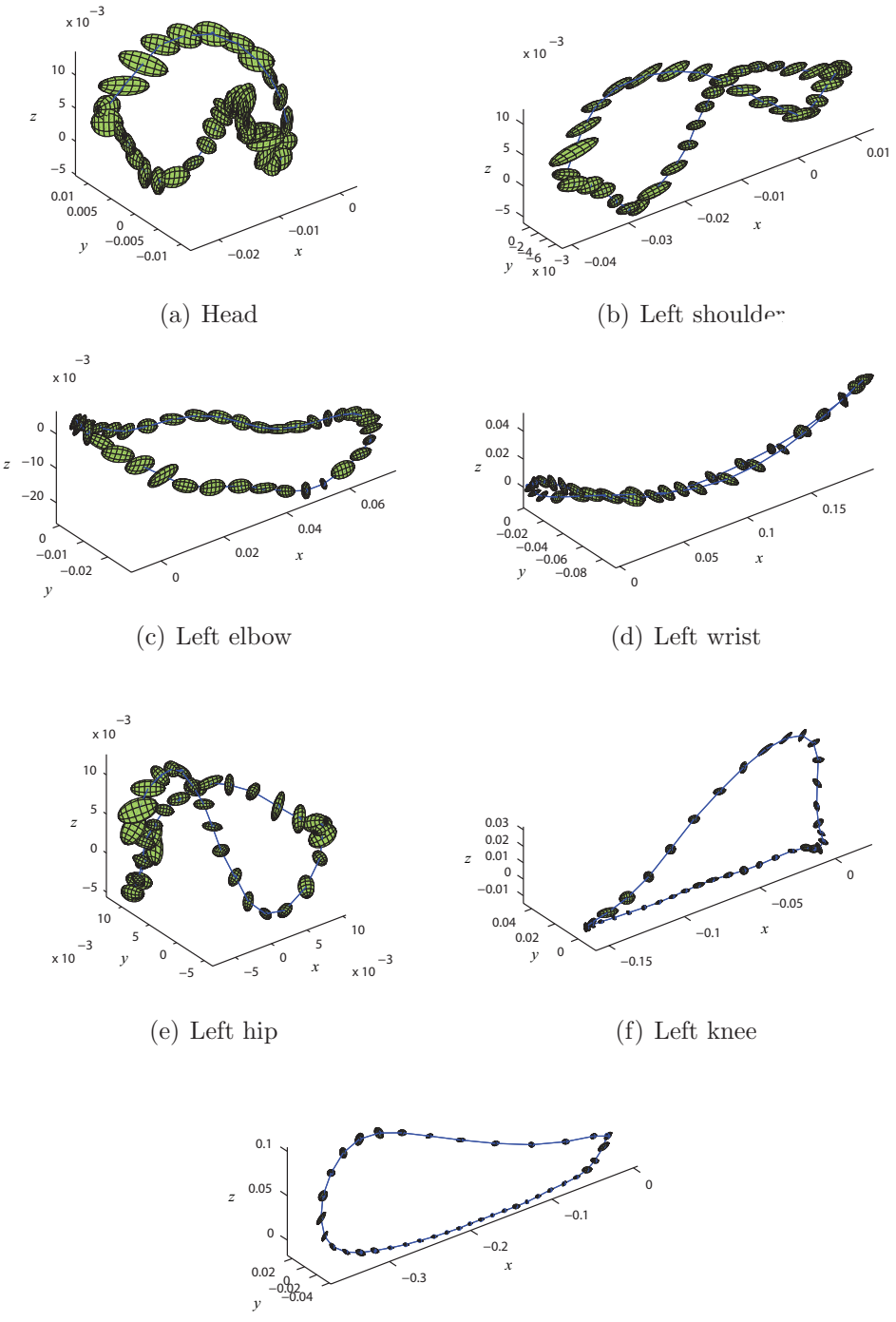
In Figure 5.17 exemplar prior poses corresponding to different gait phases are shown. The large covariance on the head is a result of the motion capture markers not being consistently placed in the same location on each subject. Large covariances can also be observed on the hands as people swing their arms by different amounts.

All the models shown in this section have been scaled by the average walker height. Before being applied to a sequence each of the models will be scaled by the height of the subject being observed, the height is automatically extracted as described in Section 5.2.

5.5.3 Results

For each sequence the KLT feature tracker was used to extract and track 200 features per frame. Any features that could no longer be tracked were automatically replaced with an alternate feature by the algorithm. The approach was tested on each of the three subjects for each of the seven cameras independently.

The average error of estimating the ground plane trajectory is shown in Table 5.1. For comparison three methods are shown, the first is a particle filter (PF) without any post processing. The second method is to temporarily smooth the image locations of the pelvis with a Gaussian filter before estimating the corresponding Pelvis position on the ground plane. The final method is that outlined in Section 5.2,



(a) Head

(b) Left shoulder

(c) Left elbow

(d) Left wrist

(e) Left hip

(f) Left knee

(g) Left foot

Figure 5.16: *3D motion models of walking learnt from ground truth data. Ellipsoids show one standard deviation of the variation expected in this motion. Each model has been scaled for walker height.*

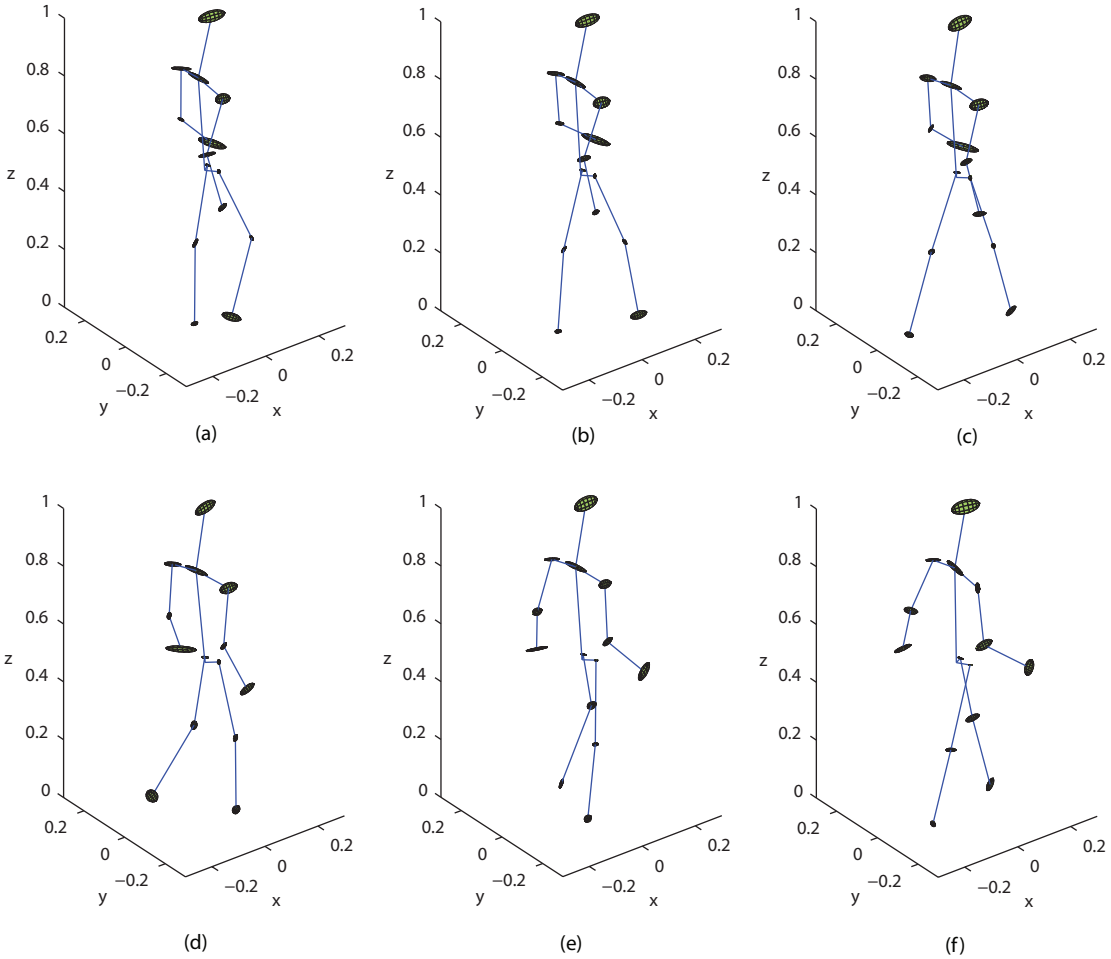


Figure 5.17: *Exemplar poses from the spatial model. Ellipsoids show one standard deviation of the variation expected in the location of each limb relative to the parents. Each model has been scaled for walker height.*

where a polynomial is fitted to the ground plane.

As can be seen from the results the polynomial achieves the most accurate results. The average error for Subject 2 is particularly poor, this is as often no features are tracked on the legs of this subject for extended periods, the result is that the particle filter estimates the person's located as being much further from the camera than it actually is.

method	S1	S2	S3	mean
PF	288.2	603.1	339.0	410.1
PF + Gaussian ($\sigma = 30$)	300.9	564.2	320.0	395.0
PF + Polynomial	213.7	520.8	250.3	328.3

Table 5.1: *Average trajectory estimation error for three different methods. (top) PF - particle filter. (middle) particle filter with the motion smoothed using a Gaussian filter. (bottom) using a 5th order polynomial. Each value is measured in mm*

For the sequences in the HumanEva data set a 5th order polynomial was found empirically to achieve the lowest error in ground plane trajectory estimation. It is likely a person walking a more complicated route would require a higher order polynomial, or alternatively walking a simpler route may need a lower order model. This is one of the limitations with this approach, an online model such as a Kalman filter may provide a better estimate on sequences that have an unknown length and this would also allow online processing. Alternatively, regularization could be used so that a higher order polynomial could always be applied, but higher order terms could be suppressed.

In Table 5.2 the average phase estimation error is shown for each sequence. The average error is noticeably higher for Subject 2 than the others, this is as estimating the motion of each feature in $\mathbb{R}^3$ is dependent on the estimated location of the subject in the ground plane in each frame. Since the error in measuring the ground plane trajectory is higher for Subject 2 the error in the phase is also likely to be higher. However, as the model has 38 phases an average error of 2.2 is still accurate and corresponds to a temporal error in estimating the phase of approximately $0.11s$.

To estimate pose the search space for each limb was defined as a cube with edges of length $400mm$,$400mm$ and $200mm$ in the x, y and z axis respectively. The

subject	C1	C2	C3	BW1	BW2	BW3	BW4	mean
S1	1.3	0.8	1.5	1.3	1.6	0.7	1.0	1.2
S2	2.1	1.8	1.4	2.6	2.5	2.3	2.5	2.2
S3	1.7	1.0	1.0	0.9	1.8	0.8	2.0	1.3
mean	1.7	1.2	1.3	1.6	1.9	1.3	1.8	1.5

Table 5.2: *Average phase estimation error.*

resolution of the grid that the search was performed over was set to be $5mm$. The presented technique required no manual initialisation, except that the HMM used to estimate the phase defined in Chapter 3 is unable to ascertain whether the left or right foot is the first to move forward. This is because, as described in Chapter 3, the number of states in the model was reduced by two so that feature tracking opposing limbs didn't vote for opposing phases. This information must be manually defined before tracking commences, however, this constitutes a binary value rather than approaches that require the 3D location of all limbs to be defined in the first frame of a sequence.

In Figures 5.18 to 5.20 some sample frames are shown with the estimated pose projected onto each frame. This shows the very close agreement between the estimated pose and that of the walker. On all sequences the projected pose shown is as it was estimated in $\mathbb{R}^3$, the poses have not in any way been realigned so that the pelvis locations of model and ground truth coincide. On the bottom of Figure 5.19 the set of features used is shown demonstrating the sparsity of the data being exploited.

Figure 5.21 shows the average error measured relative to the pelvis calculated over all limbs as a function of time. The blue line shows the error for S1 which had very accurate tracking and phase estimation, the red line shows the error for S2 which had poor ground plane tracking. Whilst the error for S1 is fairly consistent the error for S2 changes dramatically as the estimated location of the walker drifts from that of the ground truth.

This effect can further be seen in Figure 5.22 where in the middle frame shown, the location of the walker has been estimated to be much further from the camera than they really are. The reason for this can be seen in the features being used, there are very few features tracking the feet. Whilst it would be expected the presented approach could overcome missing features for a few frames, if an occlusion lasted

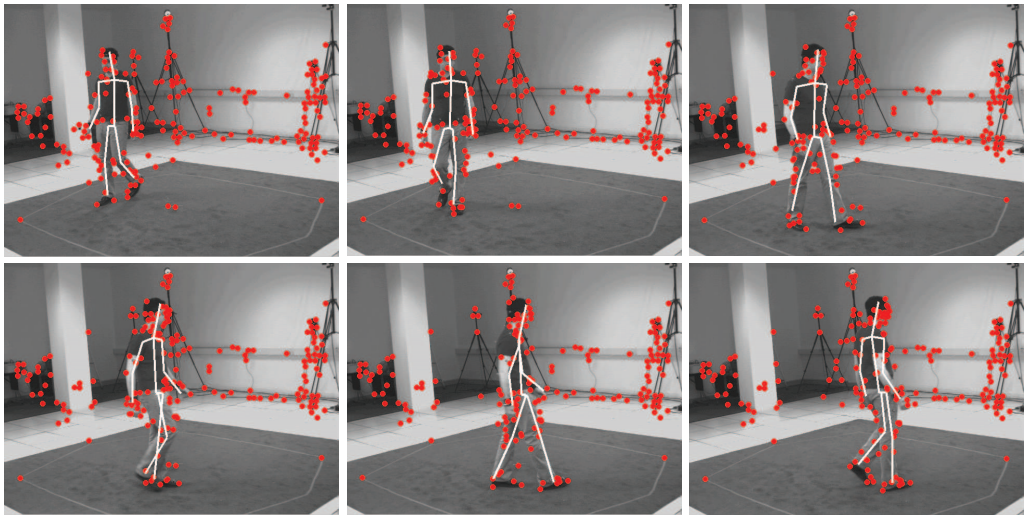


Figure 5.18: *Exemplar frames of Subject 2 filmed from camera BW4 showing the extracted pose of a person walking overlaid on the original image, the sparse tracked features are also shown.*

for a prolonged period the tracking would drift. However, as shown the approach has recovered again by the last frame shown.

Quantitative results for all camera views and each person are shown in Table 5.3. An average error of $72.4 \pm 27.5mm$ was achieved over all sequences and subjects. For comparison the results presented in [106] are used as this work represents state of the art, they also only used a monocular view and they use the same error metric. In [106] a particle filter is used for monocular tracking combined with a model that allows only physically plausible motions to occur. They present an accuracy of 64.2 ± 23.4 on a similar sequence showing that our method is capable of achieving comparable results, they also use a particle filter with no model for comparison that achieves an accuracy of $219.2mm$. This demonstrates that our approach is capable of estimating pose with a much higher accuracy then just using a standard particle filter. Currently, each frame requires 2 seconds to be processed excluding feature extraction, this is in comparison to 62 seconds as reported in [106].

One of the limitations with the presented approach is that the orientation of the person in each frame is estimated from the ground plane trajectory calculated in Section 5.2, if this is inaccurate the resultant pose will also be inaccurate. To overcome this, the presented method can be applied iteratively so that after the

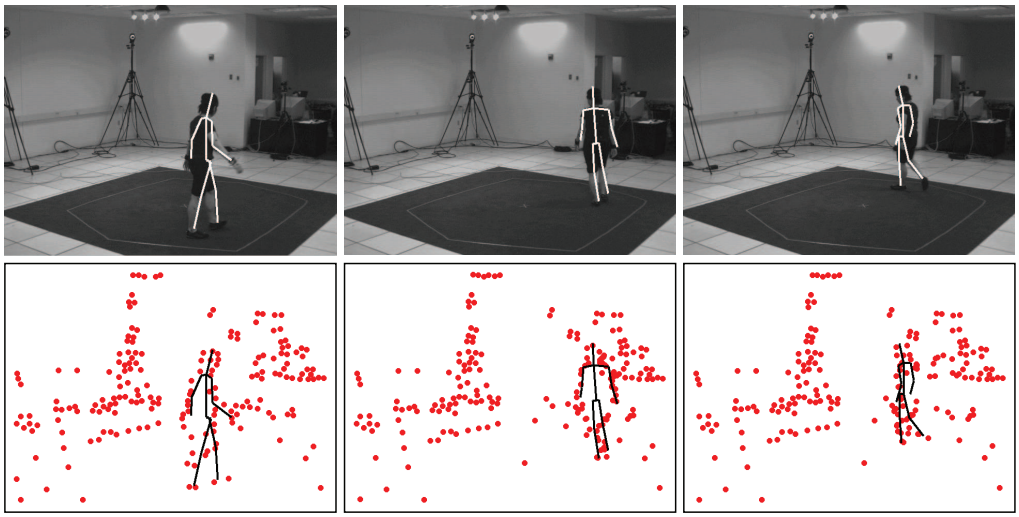


Figure 5.19: *Exemplar frames of Subject 1 filmed from camera BW1 showing the extracted pose of a person walking overlaid on the original image (top) and on the features used (bottom).*

Camera	S1	S2	S3	mean
C1	55.9 ± 12.4	107.0 ± 64.6	83.1 ± 29.5	82.0 ± 25.6
C2	57.7 ± 13.0	65.2 ± 27.4	50.4 ± 9.5	57.8 ± 7.4
C3	62.8 ± 19.4	138.8 ± 76.7	57.0 ± 14.1	86.2 ± 45.6
BW1	53.1 ± 8.7	88.3 ± 35.7	54.7 ± 14.1	65.4 ± 19.8
BW2	66.2 ± 20.1	99.3 ± 52.5	60.0 ± 16.0	75.2 ± 21.2
BW3	55.0 ± 12.5	129.6 ± 68.7	60.2 ± 26.8	81.6 ± 41.7
BW4	50.4 ± 7.5	60.8 ± 30.8	64.8 ± 20.4	58.7 ± 7.4
mean	57.3 ± 13.4	98.4 ± 50.9	61.5 ± 18.3	72.4 ± 27.5

Table 5.3: *Pose estimation errors for walking measured in mm for each subject and camera.*

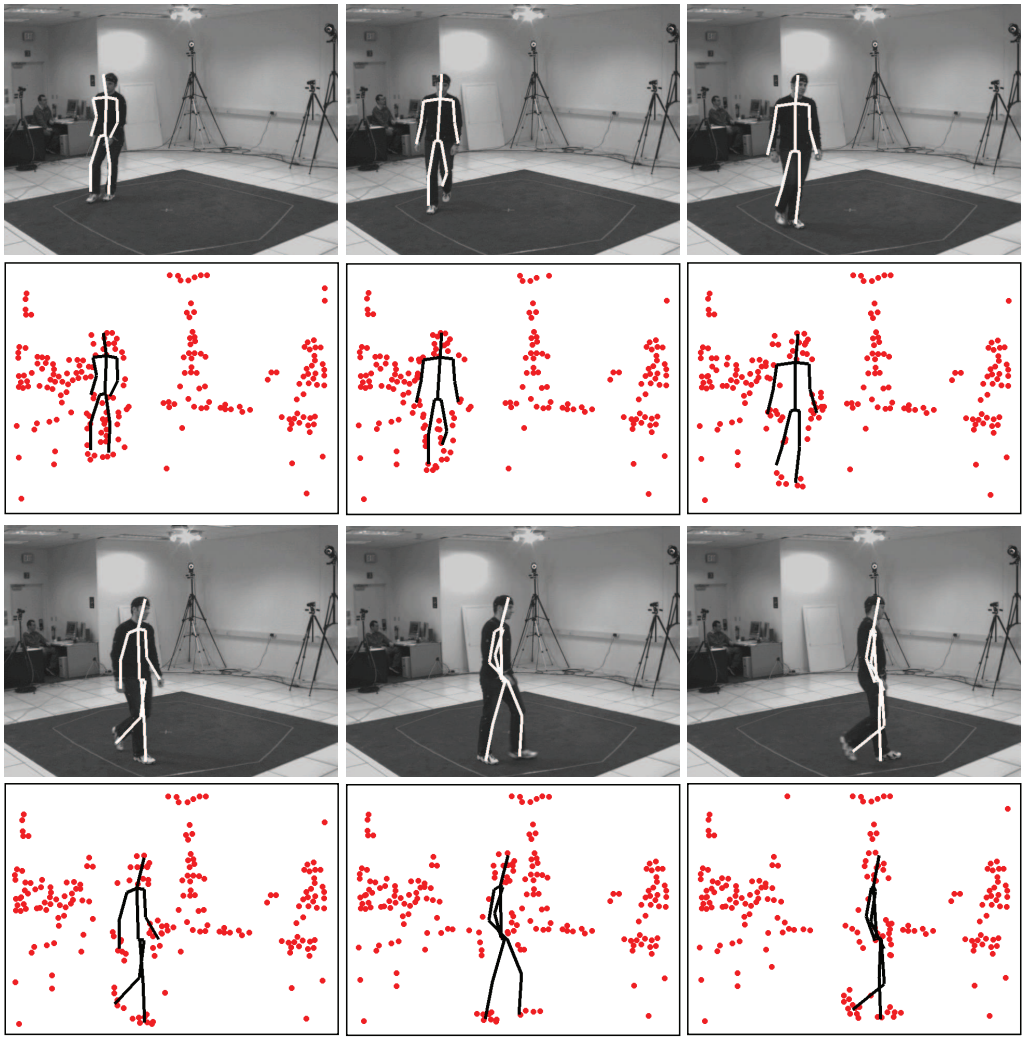


Figure 5.20: *Exemplar frames of Subject 3 filmed from camera BW2 showing the extracted pose of a person walking overlaid on the original image (top) and on the features used (bottom).*

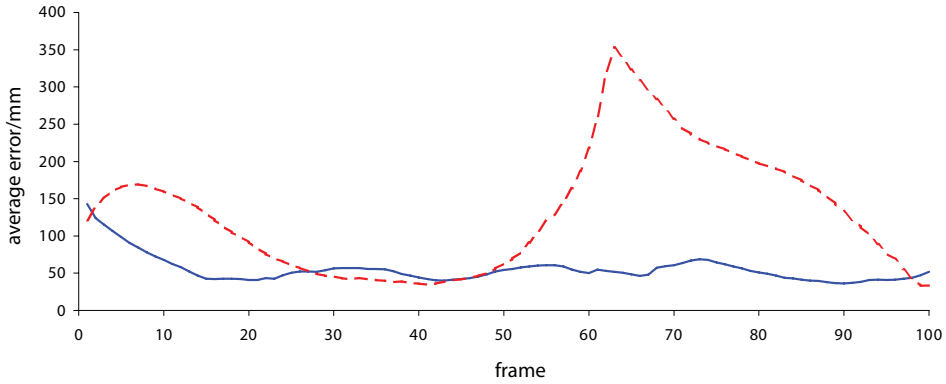


Figure 5.21: *Example of the average relative tracking error as a function of time. Blue solid line shows average error for Subject 1 camera C1. Red dashed line shows average error for Subject 2 camera C3.*

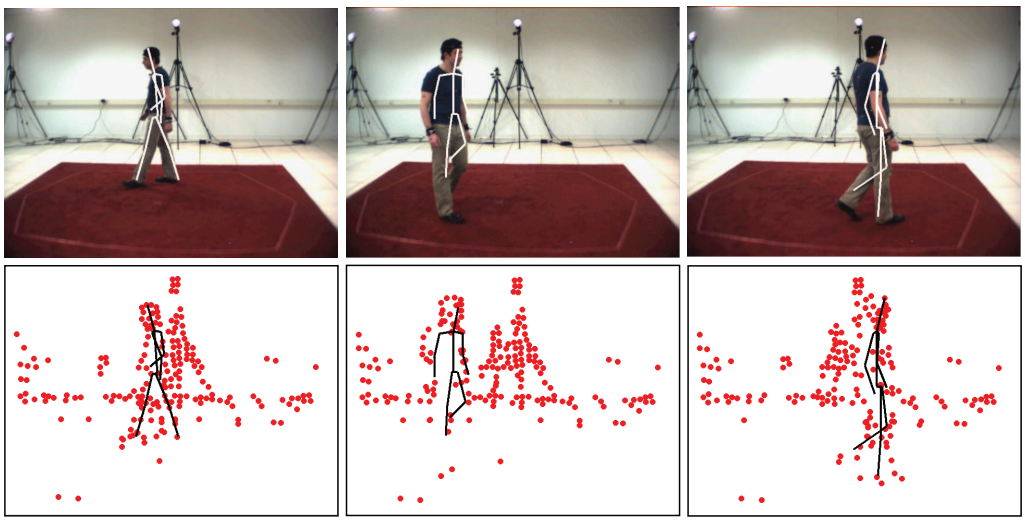


Figure 5.22: *Exemplar frames of Subject 2 filmed from camera C2 showing the extracted pose of a person walking overlaid on the original image (top) and on the features used (bottom).*

pose has been extracted from each frame, the ground plane trajectory can then be re-estimated using this new information, following which the pose for each frame can then be re-estimated and so on. If a coarse to fine search is employed the additional expense of these extra searches is relatively low, a search using a grid resolution of $20mm$ takes just 0.02 seconds per frame and a resolution of $10mm$ requires 0.21 seconds per frame. In this work 3 iterations at $20mm$ are used, followed by one at

$10mm$ and a final $5mm$ resolution search. The results for each iteration are presented in Table 5.4, these show that the error is reduced in each subsequent iteration. Using this method an accuracy of $65.8 \pm 23.3mm$ is achieved. This corresponds to about a 10% reduction in the error with no significant extra computational cost. The average result presented for two out of the three subjects used shows the average result is better than that reported in [106], however, this is achieved in a $30th$ of the time demonstrating the efficiency of the presented approach.

Subject	1. ($ds = 20$)	2. ($ds = 20$)	3. ($ds = 20$)	4. ($ds = 10$)	5. ($ds = 5$)
S1	60.7 ± 12.5	60.5 ± 12.2	60.2 ± 12.0	54.1 ± 11.9	52.7 ± 11.9
S2	99.7 ± 50.3	97.4 ± 47.6	94.5 ± 43.4	89.6 ± 43.0	88.3 ± 42.4
S3	67.4 ± 16.4	66.8 ± 15.6	66.1 ± 15.1	58.5 ± 15.5	56.4 ± 15.9
mean	75.9 ± 26.4	74.9 ± 24.1	73.6 ± 23.5	67.4 ± 23.5	65.8 ± 23.3

Table 5.4: *Pose estimation errors for walking calculated over all camera views for different iterations of the algorithm, shown by the number at the top of each column. ds is the resolution that the search was performed over measured in mm.*

5.5.4 Jogging

A model was also learnt for jogging, as the motion capture for S1 was very noisy only S2 and S3 are used for training. Jogging is a much more difficult action to estimate pose for using the presented method as there are large limb movements across frames, this will make feature tracking more difficult and result in noisier observations. Some exemplar prior poses are shown in Figure 5.23. The model used consisted of just 24 phases as a single gait cycle, as expected, is much shorter than that of walking. Some example frames with the extracted poses overlaid is shown in Figures 5.24 to 5.26.

Quantitative results are shown in Table 5.5. Despite S1 not being included in the training set accurate poses are still recovered for this subject. An average error of $69.4 \pm 20.2mm$ is achieved using the iterative method described above. As with the walking examples the worst average error is reported for Subject 2. This is largely a result of poor feature tracking on the lower body. Whilst the presented approach can overcome limbs not being tracked for short durations, if a part of the body is not tracked for longer periods the performance will degrade. This is to be expected

and is equivalent to a binary silhouette approach that is unable to extract a clean silhouette.

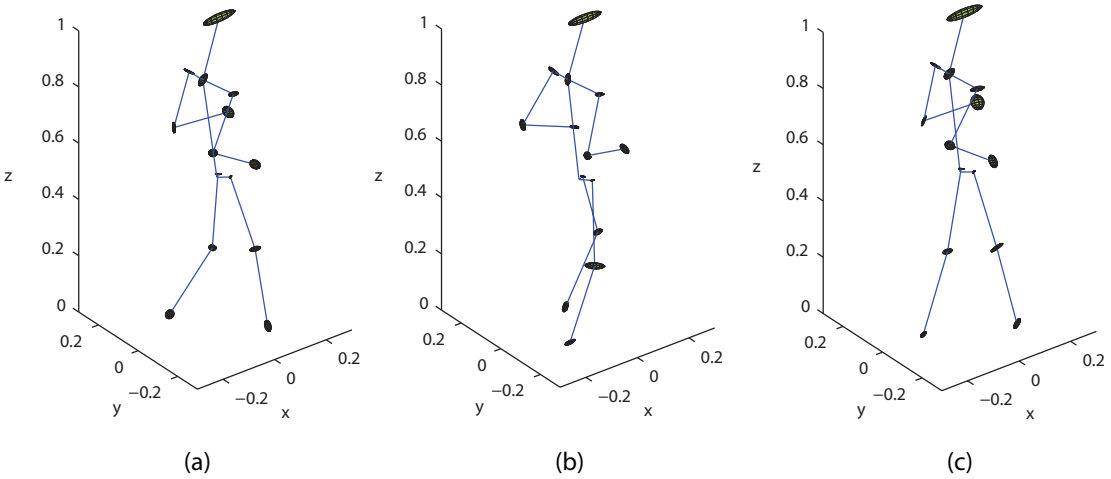


Figure 5.23: *Exemplar poses from the jogging spatial model. Ellipsoids show one standard deviation of the variation expected in the location of each limb relative to the parents. Each model has been scaled for subject height.*

5.5.5 Other Scenes

To further demonstrate the presented approach two extra scenes were filmed with people walking through them. One scene was filmed indoors and the other outdoors. Both scenes contained considerably more clutter and thus provided a more challenging environment than those contained in the HumanEva data set. Both scenes were filmed at a resolution of 720 by 576 pixels at 25fps. As these were captured at 25fps rather than 30fps, as in the HumanEva data set, the models used were resampled so that a complete gait cycle maintained the same temporal length. The resultant models contained 32 phases. The cameras were calibrated by first measuring the position of known objects in the scene and their corresponding positions in the image. Following this the projection matrix could be calculated using the Direct Linear Transform algorithm [48].

For each sequence 300 KLT features were extracted and as the sequences were typically shorter than those used in the HumanEva data set a polynomial of order 3

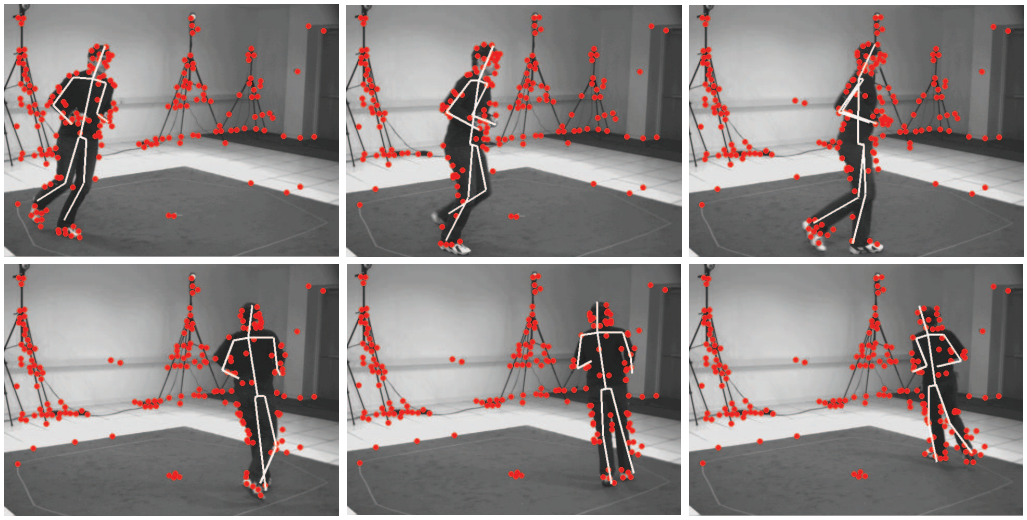


Figure 5.24: *Exemplar frames of Subject 3 filmed from camera BW3 showing the extracted pose of a person jogging overlaid on the original image, the sparse tracked features are also shown.*

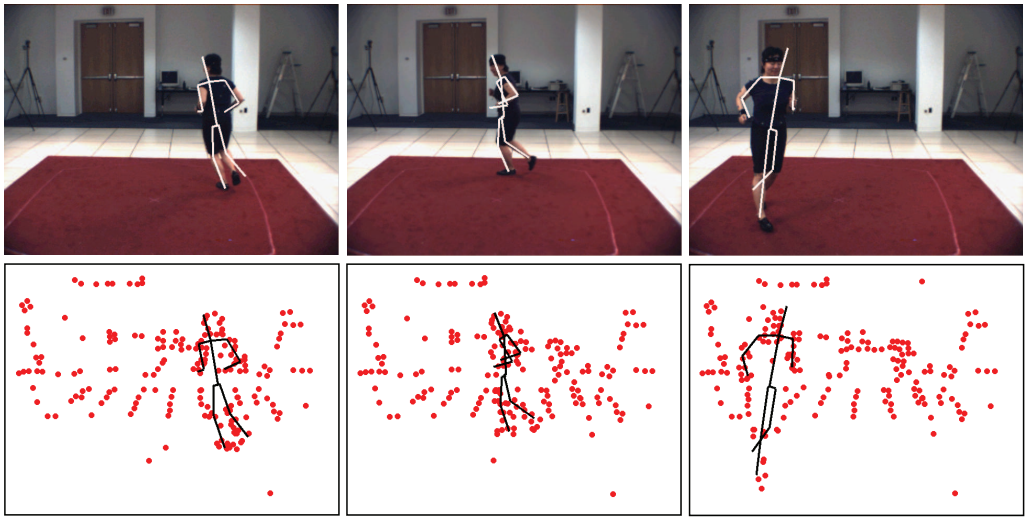


Figure 5.25: *Exemplar frames of Subject 1 filmed from camera C1 showing the extracted pose of a person jogging overlaid on the original image(top) and on the features used (bottom).*

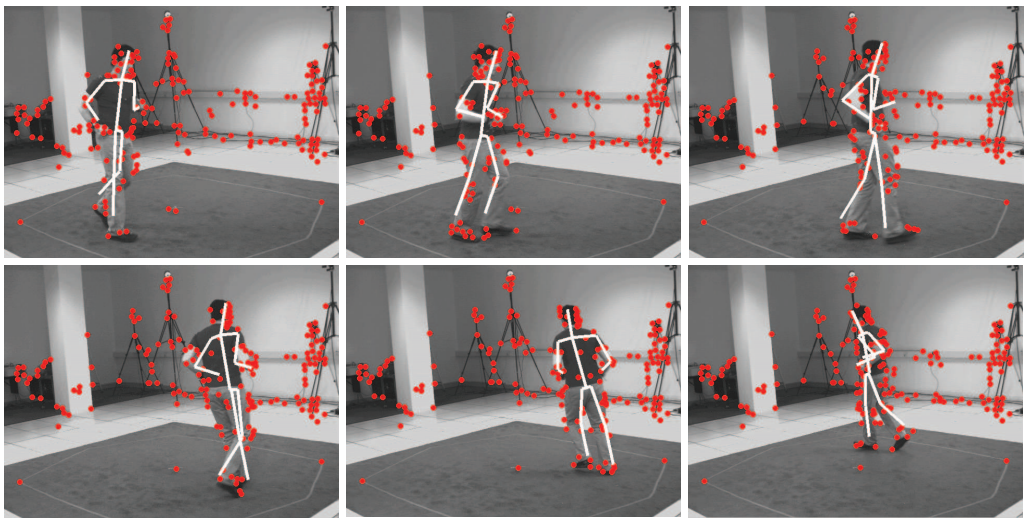


Figure 5.26: *Exemplar frames of Subject 2 filmed from camera BW4 showing the extracted pose of a person jogging overlaid on the original image, the sparse tracked features are also shown.*

Camera	S1	S2	S3	mean
C1	71.1 ± 10.1	63.5 ± 10.9	56.3 ± 26.5	63.7 ± 15.8
C2	77.2 ± 11.5	88.8 ± 38.6	39.0 ± 7.5	68.3 ± 19.2
C3	75.8 ± 11.5	87.2 ± 35.9	47.4 ± 13.4	70.1 ± 20.3
BW1	75.0 ± 12.8	88.2 ± 35.1	51.7 ± 13.1	71.7 ± 20.3
BW2	76.4 ± 13.1	105.6 ± 66.4	38.4 ± 8.7	73.5 ± 29.4
BW3	86.0 ± 19.3	97.5 ± 29.7	42.6 ± 12.4	75.4 ± 20.4
BW4	70.1 ± 10.6	74.6 ± 19.3	44.9 ± 17.6	63.2 ± 15.8
mean	75.9 ± 12.7	86.5 ± 33.7	45.8 ± 14.2	69.4 ± 20.2

Table 5.5: *Pose estimation errors for jogging measured in mm for each subject and camera.*

was used. Some exemplar frames of the estimated pose for the indoors scenes are shown in Figures 5.27 and 5.28. As can be seen the extracted poses closely resemble those of the subject being observed. In Figure 5.27 it can be seen that the KLT features are much more evenly distributed across the scene than in sequences from the HumanEva data set. It should be noted that the algorithm has no prior over the trajectory that the observed individual will walk across the scene.

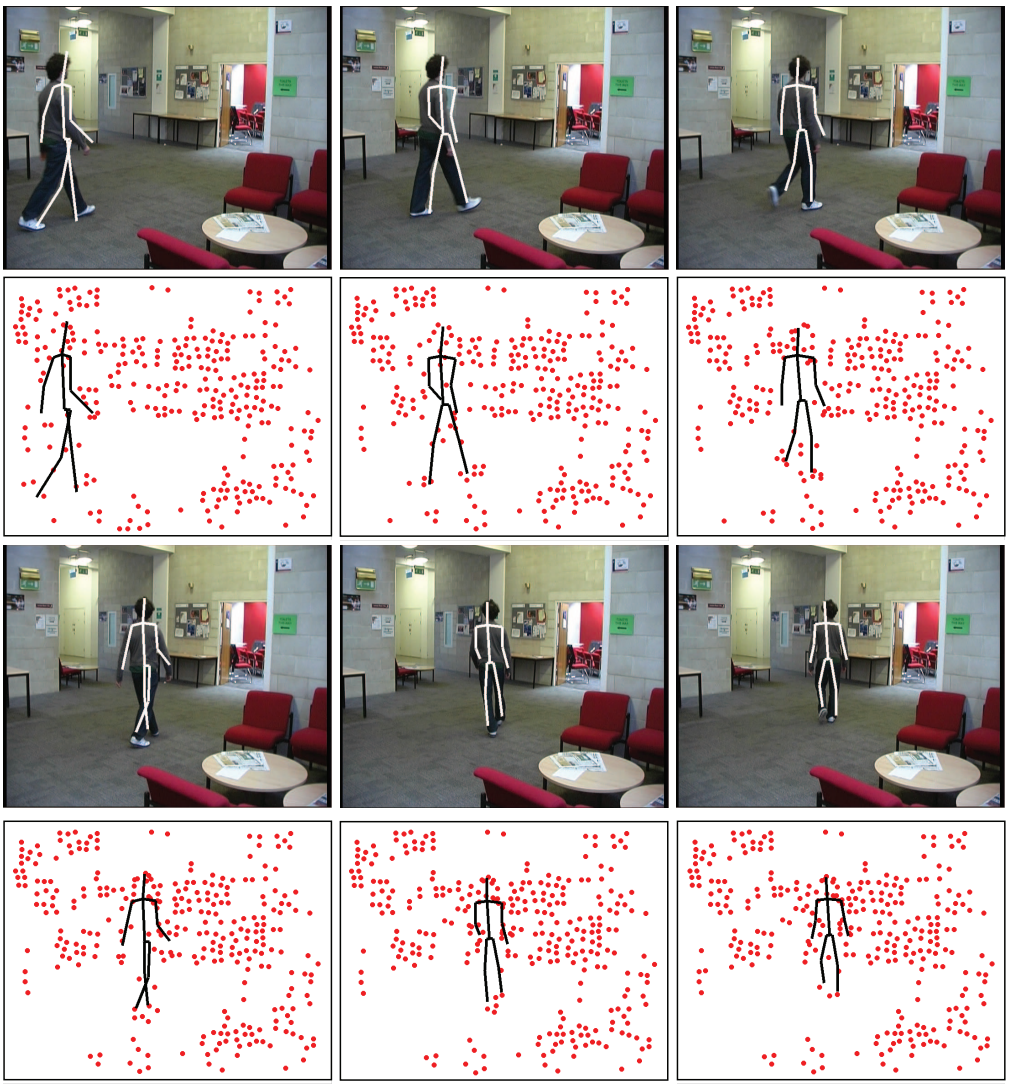


Figure 5.27: *Exemplar frames showing the extracted pose of a person walking overlaid on the original image(top) and on the features used (bottom).*

In Figures 5.29 and 5.30 exemplar frames are shown from two sequences that were filmed outdoors. This scene is particularly challenging since the ground is very textured. The result is that the KLT feature tracking is very poor since the appearance

Figure 5.28: *Exemplar frames showing the extracted pose of a person walking overlaid on the original images, the features used are also plotted.*

of features tracking the edge of the subject will change rapidly as they cross the highly textured floor. However, despite this the algorithm still accurately estimates pose.

As discussed the only initialisation needed is to label which foot is first forward, whilst this constitutes a single boolean value it is non-trivial to estimate. A method that can be used to estimate this value is to perform a low resolution search over the entire sequence assuming in turn that each foot is first forward. The correct forward leg is then taken to be the hypothesis that returns the highest likelihood. This method tested across all sequences in the HumanEva data set achieved an accuracy of 76%, significantly higher than chance. The reason perfect accuracy is not achieved is that the resultant likelihood of a pose is dependent on the location of each of the features, the exact position of which is largely down to chance. Whilst using this method would result in a drop in accuracy for those sequences where the incorrect leg is estimated as being the first forward, it provides a method so that the entire process of estimating 3D pose is completely automatic, no prior is needed on the trajectory the subject will walk through the scene or even their height, this has all been automatically extracted without the need for user intervention.

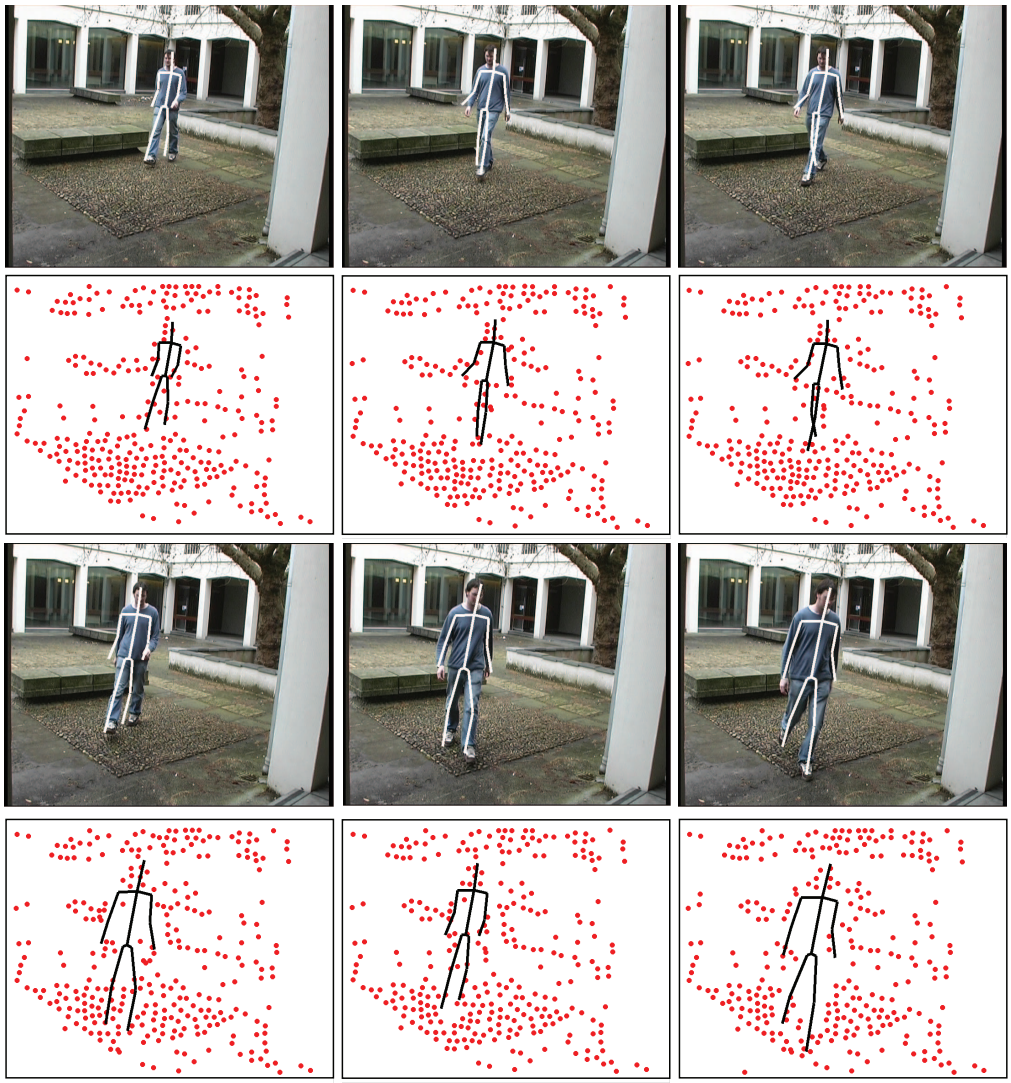


Figure 5.29: *Exemplar frames showing the extracted pose of a person walking overlaid on the original image(top) and on the features used (bottom).*

Figure 5.30: *Exemplar frames showing the extracted pose of a person walking overlaid on the original images. The features used are also shown.*

5.6 Summary

In this chapter a method has been presented to accurately extract the 3D pose of a person using just sparse motion features. This was achieved on sequences captured from just a single view requiring only 2 seconds to process each frame. This approach has been demonstrated on scenes filmed from different angles, containing people with different appearances, gait characteristics and moving along different trajectories. All of which was achieved using a single model for each gaited action. The method was also applied to more complex and difficult scenes filmed both indoors and outdoors.

These examples highlight the robustness and versatility of this approach. Quantitative results were presented for both walking and jogging and showed that results comparable to the state of the art are achievable. Models used were learnt directly from motion capture data and could be applied to people walking at any orientation to the camera, without needing to learn separate models for each representative view. Furthermore, it was demonstrated that the presented approach could be performed iteratively, at very little computational cost, to improve pose estimation errors.

The contribution of this chapter was to demonstrate that 3D pose can be extracted from the motion of a sparse set of automatically selected features as described in the thesis statement. This work illustrates the dense amount of information present in just a moving set of lights.

Chapter 6

Conclusions and Further Work

6.1 Summary

The work presented in this thesis has demonstrated that the low-level motion of a sparse set of moving features can be exploited to estimate the 3D pose of a subject performing gaited actions. This has been achieved using bottom-up searches, articulated models and a dynamic representation that has allowed the motion of each tracked feature to be interpreted using a probabilistic framework.

In Chapter 3 two dimensional models were introduced that modeled the likelihood of observing a particular motion given a feature is tracking a specific limb in a specific temporal state. The discriminative power of this representation was quantified performing three tasks, detecting gait, detecting which limb a feature was most likely tracking and estimating what temporal state a particular model was in. A HMM was then used to estimate the gait phase of the person being observed in each frame. The work in this chapter demonstrated that despite the sparseness and high levels of noise contained in the tracked features high-level information could still be extracted and this could be achieved using only motion, no structural cues were used.

In Chapter 4 the results of Chapter 3 were exploited and dense probability maps were constructed, these represented the likelihood of a limb being at a particular

location in the image plane. Given these dense probability maps, phase dependent spatial priors were used to perform efficient bottom-up searches. Improvements to the efficiency of the search were presented by performing each search over the angle of each joint relative to its parent. This approach was demonstrated on both humans walking and a lion walking, the model of which was learnt from a cheetah. The results of this chapter illustrated that 2D pose estimation, given some prior expectation of direction and scale, was achievable, suggesting 3D pose estimation may be realised.

Chapter 5 extended the methods presented in Chapters 3 and 4 to three dimensions. This was achieved by first estimating the trajectory the subject walked across the ground plane. This provided constraints on both their position and orientation through the assumption that they face along the direction of travel. Both 3D motion and spatial models were learnt directly from motion capture data. 3D motion trajectories were extracted for each feature using their observed 2D motion in the image plane. As well as allowing 3D pose to be extracted this relaxed several assumptions from the previous chapter, such as having prior knowledge of the subject's height or direction, methods were presented to extract this information automatically. Following this bottom-up searches were used to extract 3D pose in each frame independently. In this chapter 3D pose estimation has been achieved using a single viewpoint and just a sparse set of noisy, automatically extracted features. The presented method required no manual initialisation and operates at 2 seconds per frame. It has been applied to different scenes filmed from different viewpoints, using the same 3D models. The contribution of this chapter was to demonstrate that as described in the thesis statement the motion of a sparse set of features can be used to extract 3D pose of subjects performing gaited actions, using only a single view.

The motivation for this work came from psychophysical experiments using the MLD. However, the biological motion community were unable to provide answers as to whether 3D pose estimation from the MLD was possible or, assuming it was, how accurately humans could perform this task. In this thesis it has been demonstrated that this is at least possible and results have been provided. Whilst initially motivation was provided by the community, it can perhaps be claimed that in this thesis a step has been taken beyond current understanding of human perception, this in turn could provide motivation for those interested in human perception of biological

motion.

A summary of the main conclusions made from the work contained within this thesis are described below:

- The motion of a single tracked feature contains extractable information about the state of the gaited action being observed and which limb the observed motion is most likely that of.
- Integrating over all observed features the phase of the gaited action being observed can be accurately estimated without exploiting any information about the form of the observations.
- Despite the sparsity of the observed features, novel poses can still be inferred from the structure of the features that are not representative of those contained in the original training set.
- 3D motion trajectories can be extracted from 2D motions observed in the image plane by assuming the dominant motion occurs in the subject's direction of travel. Furthermore, the error introduced through this approximation is small enough so that phase can still be accurately estimated.
- A comparison of computation times showed that it is more efficient to create 3D probability volumes, in which 3D searches can be directly performed, rather than repeatedly reprojecting hypotheses into the image plane.
- Finally, from the work contained in this thesis, it can be concluded that *it is possible to extract the 3D pose of gaited actions using only the motion of a sparse set of features automatically extracted and using only a single viewpoint.*

6.2 Future Work

A current limitation with the work presented in this thesis is that it is dependent on observing only gaited actions. Several of the assumptions made would fail for any other type of action. However, experiments using the moving light display have never demonstrated that unknown actions could be recognised, this implies that

using this sparse representation to extract pose from an unknown motion is not possible. Most of the tasks accomplished by humans observing the MLD have been recognition tasks and therefore assume a model of the motion being observed is already known. However, it would be interesting to investigate whether the pose of a person performing an unknown action could be recovered, this would be of interest to the psychophysics as well as the computer vision community since it suggests a set of rules (e.g. rigid structure) may exist so that 3D pose can first be extracted before action recognition takes place.

Using multiple cameras would also be of interest, this would be expected to improve accuracy and feature matching across neighboring views would allow real 3D motion data to be extracted. Using this information, tasks such as first estimating the trajectory across the ground plane would not be necessary.

Finally, this work could be integrated with appearance models. The method presented in this thesis is considerably faster than current appearance approaches so could be used to initialise appearance approaches so that strong appearance models could be learnt (i.e. using color histograms, texture etc). Alternatively, it could be used to initialise the search in each frame so that optimization using more discriminative appearance models could be performed using local gradient descent methods.

Bibliography

[1] A report on the surveillance society. *For the Information Commissioner by the Surveillance Studies Network*, 2006.

[2] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):555–560, 2008.

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

[4] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[5] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[6] D. Batra, T. Chen, and R. Sukthankar. Space-time shapelets for action recognition. In *IEEE Workshop on Motion and video Computing*, pages 1–6, 2008.

[7] A. Baumberg and D. Hogg. An efficient method for contour tracking using active shape models. In *Proceedings of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 194–199, 1994.

[8] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 2006. ISBN 0387310738.

[9] A. Bissacco, M.-H. Yang, and S. Soatto. Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[10] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[11] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *IEEE International Conference on Computer Vision*, pages 462–469, 2005.

[12] I. Bouchrika and M. S. Nixon. Markerless feature extraction for gait analysis. In *5th Chapter Conference on Advances in Cybernetic Systems*, pages 55–60, 2006.

[13] C. Bregler. Learning and recognizing human dynamics in video sequences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 568–574, 1997.

[14] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8–15, 1998.

[15] M. Brubaker and D. J. Fleet. The kneed walker for human pose tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[16] M. Brubaker, D. J. Fleet, and A. Hertzmann. Physics-based human pose tracking. In *NIPS- Workshop on Evaluation of Articulated Human Motion and Pose Estimation. EHuM06*, 2006.

[17] M. Brubaker, D. J. Fleet, and A. Hertzmann. Physics-based person tracking using simplified lower-body dynamics. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[18] J. Coughlan, A. Yuille, C. English, and D. Snow. Efficient deformable template detection and localization without user initialization. *Computer Vision and Image Understanding*, 78(3):303–319, 2000.

[19] R. Cutler and L. S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796, 2000.

[20] J. Cutting and L. Kozlowski. Recognizing friends by their walk: Gait perception without familiarity cues. *Bulletin Psychonometric Society*, 9:353–356, 1977.

[21] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[22] H. Dee and D. Hogg. Detecting inexplicable behaviour. In *British Machine Vision Conference*, pages 477–486, 2004.

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

[24] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 126–133, 2000.

[25] J. Diard, P. Bessiere, and E. Mazer. A survey of probabilistic models using the bayesian programming methodology as a unifying framework. In *Proceedings of the International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2003.

[26] M. Dimitrijevic, V. Lepetit, and P. Fua. Human body pose detection using bayesian spatio-temporal templates. *Computer Vision and Image Understanding*, 104(2):127–139, 2006.

[27] W. H. Dittrich. Action categories and the perception of biological motion. *Perception*, 22:15–22, 1993.

[28] W. H. Dittrich, T. Troscianko, S. E. G. Lea, and D. Morgan. Perception of emotion from dynamic point-light displays represented in dance. *Perception*, 25:727–738, 1996.

[29] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE international workshop on: Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[30] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, pages 726–734, 2003.

[31] C. Ek, P. Torr, and N. Lawrence. Gaussian process latent variable models for human pose estimation. *Machine Learning for Multimodal Interaction*, pages 132–143, 2008.

[32] A. Fathi and G. Mori. Human pose estimation using motion exemplars. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[33] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[34] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. In *Cornell Computing and Information Science Technical Report TR2004-1963*, 2004.

[35] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal on Computer Vision*, pages 55–79, 2005.

[36] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[37] P. F. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 66–73, 2000.

[38] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[39] N. I. Fischer. *Statistical Analysis of Circular Data*. 1993. ISBN 0521568900.

[40] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications ACM*, 24(6):381–395, 1981.

[41] M. A. Fischler and R. A. Elsclager. The representation and matching of pictorial structures. In *IEEE Transactions on Computer, 22(1)*, pages 67–92, 1973.

[42] A. Fossati, M. Dimitrijevic, V. Lepetit, and P. Fua. Bridging the gap between detection and tracking for 3d monocular video-based motion capture. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[43] J. Gao and J. Shi. Multiple frame motion inference using belief propagation. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[44] J. Gao, A. Hauptmann, and H. Wactlar. Combining motion segmentation with tracking for activity analysis. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 699–704, 2004.

[45] D. Gavrila. Pedestrian detection from a moving vehicle. In *European Conference on Computer Vision*, pages 37–49, 2000.

[46] D. Gibson, N. Campbell, and B. Thomas. Quadruped gait analysis using sparse motion information. In *International Conference on Image Processing*, volume 3, pages 333–336, 2003.

[47] S. L. Hanunna. *Quadruped Gait Detection in Low Quality Wildlife Video*. PhD thesis, University of Bristol, Bristol, UK, 2008.

[48] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. ISBN 0521623049.

[49] D. D. Hoffman and B. E. Flinchbaugh. The interpretation of biological motion. *Biological Cybernetics*, 42:195–204, 1982.

[50] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal on Computer Vision*, 29:5–28, 1998.

[51] H. Jiang and D. R. Martin. Global pose estimation using non-tree models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[52] G. Johansson. Spatio-temporal differentiation and integration in visual motion perception. *Psychological Research*, 38:379–393, 1974.

[53] G. Johansson. Visual perception for biological motion and a model for its perception. *Perception and Psychophysics*, 14:201–210, 1973.

[54] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 38–44, 1996.

[55] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In *American Control Conference*, volume 3, pages 1628–1632, 1995.

[56] X. Lan and D. P. Huttenlocher. A unified spatio-temporal articulated model for tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 722–729, 2004.

[57] X. Lan and D. P. Huttenlocher. Beyond trees: Common-factor models for 2d human pose recovery. In *IEEE International Conference on Computer Vision*, pages 470–477, 2005.

[58] I. Laptev and T. Lindeberg. Space-time interest points. In *IEEE International Conference on Computer Vision*, volume 1, pages 432–439, 2003.

[59] C.-S. Lee and A. Elgammal. Body pose tracking from uncalibrated camera using supervised manifold learning. In *NIPS - Workshop on Evaluation of Articulated Human Motion and Pose Estimation. EHuM06*, 2006.

[60] M. W. Lee and R. Navatia. Human pose tracking using multi-level structure. In *European Conference on Computer Vision*, volume 3, pages 368–381, 2006.

[61] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 878–885, 2005.

[62] R. Li, M.-H. Yang, S. Sclaroff, and T.-P. Tian. Monocular tracking of 3d human motion with a coordinated mixture of factor analyzers. In *European Conference on Computer Vision*, volume 2, pages 137–150, 2006.

[63] F. Loula, S. Prasad, K. Harber, and M. Shiffrar. Recognizing people from their movement. *Journal of experimental psychology. Human perception and performance*, 31(1):210–220, 2005.

[64] G. Loy, M. Eriksson, J. Sullivan, and S. Carlsson. Monocular 3d reconstruction of human motion in long action sequences. In *European Conference on Computer Vision*, volume 4, pages 442–455, 2004.

[65] G. Mather and L. Murdoch. Gender discrimination in biological motion displays based on dynamic cues. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 258:273–279, 1994.

[66] G. Mather and S. West. Recognition of animal locomotion from dynamic point-light walker displays. *Perception*, 22:759–766, 1993.

[67] A. Micilotta, E.-J. Ong, and R. Bowden. Real-time upper body detection and 3d pose estimation in monoscopic images. In *European Conference on Computer Vision*, pages 139–150, 2006.

[68] J. Mitchelson and A. Hilton. Hierarchical tracking of multiple people. In *British Machine Vision Conference*, 2003.

[69] C. B. Moler. *Numerical Computing with Matlab*. Society for Industrial & Applied Mathematics, 2004. ISBN 0898715601.

[70] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision*, volume 3, pages 666–680, 2002.

[71] R. Navaratnam, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Hierarchical part-based human body pose estimation. In *British Machine Vision Conference*, 2005.

[72] P. Neri, C. Morrone, and D. Burr. Seeing biological motion. *Letters to nature*, 395:894–896, 1998.

[73] E.-J. Ong, A. S. Micilotta, R. Bowden, and A. Hilton. Viewpoint invariant exemplar-based 3d human tracking. *Computer Vision and Image Understanding*, 104(2):178–189, 2006.

[74] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., 1988. ISBN 0-934613-73-7.

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

[76] V. Rabaud and S. Belongie. Counting crowded moving objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 705–711, 2006.

[77] L. R. Rabiner. A tutorial on hidden markov models and selected apllications in speech recognition. In *Readings in Speech Recognition*, pages 267–296, 1990.

[78] M. M. Rahman and S. Ishikawa. Human motion recognition using an eigenspace. *Pattern Recognition Letters*, 26(6):687–697, 2005.

[79] D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 467–474, 2003.

[80] D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: tracking people by finding stylized poses. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 271–278, 2005.

[81] C. Rao and M. Shah. View-invariance in action recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 316–322, 2001.

[82] L. Raskin, E. Rivlin, and M. Rudzsky. Using gaussian process annealing particle filter for 3d human tracking. *EURASIP Journal on Advances in Signal Processing*, 2008.

[83] J. Rittscher, A. Blake, A. Hoogs, and G. Stein. Mathematical modelling of animate and intentional motion. *Philisophical Transactions of The Royal Society London Biological Sciences*, 358(1431):475–490, year = 2003.

[84] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. H. S. Torr. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[85] R. Rosales and S. Sclaroff. Inferring body pose without tracking body parts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 721–727, 2000.

[86] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *International Conference on Pattern Recognition*, volume 3, pages 32–36, 2004.

[87] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *IEEE International Conference on Computer Vision*, volume 2, pages 750–757, 2003.

[88] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[89] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic Tracking of 3D Human Figures using 2D Image Motion. In *European Conference on Computer Vision*, pages 702–718, 2000.

[90] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. In *Brown University Technical Report*, 2006.

[91] L. Sigal, M. Isard, B. H. Sigelman, and M. J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *NIPS*, pages 1539–1546, 2003.

[92] L. Sigal, B. Sidharth, S. Roth, M. Black, and M. Isard. Tracking loose-limbed people. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 421–428, 2004.

[93] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 447–454, 2001.

[94] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 390–397, 2005.

[95] Y. Song, L. Goncalves, and P. Perona. Learning probabilistic structure for human motion detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 771–777, 2001.

[96] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 605–612, 2003.

[97] S. Sumi. Upside-down presentation of the johansson moving light-spot pattern. *Perception*, 13:283–286, 1984.

[98] X. Sun, C.-W. Chen, and B. S. Manjunath. Probabilistic motion parameter models for human activity recognition. *International Conference on Pattern Recognition*, 1, 2002.

[99] M. Thirkettle, C. P. Benton, and N. E. Scott-Samuel. Contributions of form, motion and task to biological motion perception. *Journal of Vision*, 28:1–11, 2009.

[100] N. F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2:371–387, 2002.

[101] I. Ulusoy and C. M. Bishop. Generative versus discriminative methods for object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 258–265, 2005.

[102] R. Urtasun, D. J. Fleet, and P. Fua. Temporal motion models for monocular and multiview 3d human body tracking. *Computer Vision and Image Understanding*, 104(2):157–177, 2006.

[103] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 238–245, 2006.

[104] N. Vaswani, A. Roy Chowdhury, and R. Chellappa. Activity recognition using the dynamics of the configuration of interacting objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 633–640, 2003.

[105] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 734–741, 2003.

[106] M. Vondrak, L. Sigal, and O. C. Jenkins. Physical simulation for probabilistic motion tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[107] A. D. Wilson and A. F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):884–900, 1999.

[108] C. R. Wren and A. P. Pentland. Dynamic models of human motion. In *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, pages 22–27, 1998.

[109] X. Xu and B. Li. Learning motion correlation for tracking articulated human body with a rao-blackwellised particle filter. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[110] X. Zhao and Y. Liu. Generative tracking of 3d human motion by hierarchical annealed genetic algorithm. *Pattern Recognition*, 41(8):2470 – 2483, 2008.

[111] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2004.