School of Electrical Engineering and Computing
Department of Computing

Human Pose Tracking from Monocular Image Sequences

Jinglan Tian

This thesis is presented for the Degree of
Doctor of Philosophy
at
Curtin University

August 2016

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Jinglan Tian	Date

Abstract

Tracking 2D articulated human poses in monocular sequences is an important task in computer vision due to its extensive applicability. Many algorithms have been proposed during the past decades and some progress has been achieved. However, building a robust human motion tracking framework remains a challenging task, especially when body sizes of the tracked target vary a lot within a video sequence, images come with cluttered background, and/or body parts are self-occluded in side-facing poses.

This thesis aims to propose a comprehensive and robust tracking framework for human motion tracking on 2D monocular video sequences including rare and complex motions and proposes various novel approaches to improve the tracking performance. Firstly, to detect and track the human pose in each frame with a proper perspective scale, a multi-scale strategy (MSS) module is proposed to implement scale checking and adjusting at the beginning of the tracking process. It enables the tracking framework to produce satisfactory tracking performance for video sequences when the sizes of the target projection vary, thus improves the performance of the tracking framework especially for scale-variation cases. Secondly, to improve the accuracy of the tracking framework, especially when dealing with rare and complex poses, the representation of body parts and their relations are investigated in this thesis. A mixture of mid-level spatial representations, named poselets, are utilized to constrain relations of multiple body parts, which are more expressive and specific to images. These higher-level information among multiple body parts are used to guide estimation of every single part by providing more image-conditioned information on pose configurations. Thirdly, the human body model used for pose estimation is developed to further improve the tracking performance, especially for active body parts. More dependencies between symmetric and non-adjacent body parts are introduced (AdCon), which can help correct some detection errors due to limbs drifting and double counting. The proposed body model is incorporated in a factor graph, which leads to a significant improvement in the accuracy of human pose tracking while being computationally efficient. Finally, in order to distinguish between the left and right limbs during tracking, a simple yet effective head orientation (looking left or right) estimation is proposed to serve as a complementary tool to assist the

human pose estimation. The inclusion of this step during tracking further reduces the occurrence of double counting and helps distinguish the left and right limbs consistently.

This thesis implements and evaluates a complete framework that utilizes the four proposed approaches to perform the task of tracking human poses based on monocular view. Experiments are conducted on several challenging video sequences (publicly-available or collected by ourselves with no restriction on the environment) and evaluated with reference to the ground truth. The experimental results demonstrate that the proposed framework outperforms existing systems significantly, especially for the active body parts such as forearms and lower-legs.

Acknowledgments

I would like to express my sincere thanks to the following people whose help and contribution make this thesis possible.

First, I would like to thank my main supervisor, Professor Ling Li. She provided an unending source of motivation and support across the full spectrum, from suggesting a really interesting topic to continuous guidance and help in all aspects of my research. I would also like to thank my co-supervisor, Professor Wanquan Liu, who gave me the opportunity to do a PhD and has provided much useful advice and support to me over the course of the PhD. In addition, I am grateful to Dr Patrick Peursum, who provided much help and advice on technical problems at the beginning of my PhD.

Thanks to my mum, my sisters, my huaband and my lovely son, for all their care, support and love. They are one of the main motivations that get me through here.

I must also thank all the people who made their software available for free on the Web, providing invaluable tools for this research. These included Dr Leonid Pishchulin and Dr Mykhaylo Andriluka at Max Planck Institute for Informatics.

Finally, thanks to China Scholarship Council (CSC) and Curtin International Postgraduate Research Scholarships (CIPRS) for providing me with a scholarship to support my PhD study. I would also like to thanks our administrative staff for their paper works and all the IT staff for their technical help. Special thanks to Leo Zhang and Gongqi Lin who participated in my data collection section, which directly contributes to part of this thesis.

Published Work

This thesis includes the following works that have been published over the course of my PhD study.

- Jinglan Tian, Ling Li, and Wanquan Liu. (2014) Multi-scale human pose tracking in 2d monocular images. *Journal of Computer and Communications*. 2, no. 02 : 78.

- Jinglan Tian, Ling Li, and Wanquan Liu. (2014) A robust framework for 2d human pose tracking with spatial and temporal constraints. *International Conference on Digital lmage Computing: Techniques and Applications, IEEE*.

- Jinglan Tian, Ling Li, and Wanquan Liu. (2015) Monocular Human Motion Tracking with Non-Connected Body Part Dependency. *International Conference on Digital Image Computing: Techniques and Applications, IEEE*.

- Jinglan Tian, Yao Lu, Ling Li, and Wanquan Liu. (2016) Tracking human poses in various scales with accurate appearance. *International Journal of Machine Learning and Cybernetics*. pages 1-14.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Monocular cameras are the most widely and easily available sources that record all kinds of human activities. This thesis focuses on the task of tracking articulated 2D human poses in monocular videos. Considerable research effort has been spent on human pose estimation and tracking from monocular sequences over the past few decades due to its extensive applicability in computer vision, such as video surveillance, human-computer interactions, motion analysis, etc. In some approaches pose estimation and tracking depends on electromagnetic sensors that are attached to the human body. However such systems are generally costly and sometimes unstable. Since many applications would benefit from cheaper and more convenient vision-based tracking approaches using cameras, this topic has received increasing attention. The goal of 2D human pose tracking is to track the articulations of people using 2D representations in video sequences. Many algorithms have been proposed during the past decades (Poppe (2007); Zhou and Hu (2008); Ramanan *et al.* (2007); Lu *et al.* (2012b); Ramakrishna *et al.* (2013)). However, building a robust human motion tracking framework using 2D information remains a challenging task.

There are several challenges in 2D human pose tracking. One issue arises from the variations of body size in one video sequence. In order to perform tracking efficiently, most human pose tracking approaches assume that the tracked target in a video sequence are moving with a rather fixed distance to the camera, resulting in the size of the human figure in the video to be constant or near constant, i.e., the perspective scale is fixed. However, videos in reality often contain people moving towards or away from the camera hence appeared in various scales in the videos. The complexities of human motion add into this problem as well, since the human body could lean towards or away from the camera, causing size variations.

Another obvious problem is that, unlike tracking with multiple cameras, no depth information is available in monocular videos, thus cluttered backgrounds and self-occlusions make the problem more complex. Most of 2D human pose tracking frame-

works are inspired by the development of bottom-up pose estimation approaches with the generic Pictorial Structures (PicStr) model, in which the part detectors are trained based on shape features, and the priors over body part connections are assumed to be a tree structure independent to image evidence. This means that shapes similar to limbs in the background and similarities between body parts, especially the symmetric ones, are big issues affecting the performance of this type of tracking frameworks. Double counting is a common problem occurring due to similarities between the symmetric body parts especially when the tracked target appears in sideways poses, i.e., symmetric body part pairs are often arranged at the same location in images when they share a high detection score at the same image evidence. Additionally, the generic PicStr model only considers the dependencies between the connected body parts. However, humans have a distinct ability to maintain body balance and coordination. Therefore for complex poses, in order to improve the performance, the tracking framework should also consider dependencies between or among non-connected body parts.

The confusion between similar-looking body parts, especially the left and right limbs, is also an issue during human pose tracking. With the generic PicStr model, although the frontal poses can be accurately detected especially when all body parts are visible, the left and right limbs are often confused especially when the human body is side-faced due to the overlapping of these body parts and the similarities of their appearances in terms of colour and shape.

This thesis explores a framework for human pose tracking on monocular video sequences based on 2D image information, with specific goals to address the issues mentioned above.

1.1 Aims and Approaches

This thesis aims to propose a comprehensive and highly accurate tracking framework based on 2D monocular video for tracking human motion, including rare and complex motion. The specific objectives of this thesis are as follows:

1. Develop new approaches and algorithms to evaluate the perspective scales, to

estimate and modify the appropriate scale values in order to accurately track human poses with multiple scales in monocular video sequences.

2. Add more dependencies among multiple and non-connected body parts to make the human body model more powerful for human pose estimation and tracking. These dependencies should be more expressive and capture the spatial relationships among multiple body parts. It is expected that body parts can be detected more accurately using these dependencies and the double counting problem can be largely solved while the structure of the basic PicStr model is not changed.

3. Model the dependencies between symmetric body parts to further decrease the rate of double counting occurrence. These dependencies are utilized to constrain the left and right limbs, which greatly complement the shortcomings of the simple traditional tree-based PicStr model by encoding more natural human distinction for body balance and coordination.

4. Propose a simple yet effective method to address the confusion between the left and right limbs during tracking, which is a common issue especially when the tracked target shows different views in a sequence.

To achieve the objectives above, several approaches are proposed in this thesis.

1. To detect and track the human pose in each frame with a proper scale, a scale checking and adjusting step is incorporated into the tracking process. Two metrics are proposed for detecting and adjusting the scale change. One metric is from the height value of the tracked target (*Height_Metric*), which is suitable for sequences where the tracked target generally maintains upright postures with no obvious limbs stretching. For such kind of sequences, a full body detector is proposed to estimate the height of the tracked target in each frame. The other metric is named *PixelCount_Metric* which is able to represent scale changes invariant to motion types, thus is more generic. Specifically, the images are firstly processed with foreground segmentation which aims to obtain an approximate size of the body blob. This blob size is not used to determine the scale directly. Rather it is used to compare with the size of the estimated human body (normally in the shape of bounding boxes) from

pose estimation to determine whether the scale used for the pose estimation is appropriate. If the comparison shows that the scale value used satisfies a preset condition, the algorithm will proceed to the next frame using the same scale value. Otherwise, the scale value will be adjusted and the frame will be re-processed until the preset condition is met.

2. To improve the accuracy of the tracking framework, especially for tracking motions including rare and complex poses, the representation of body parts and their relationship utilized in the tracking framework is re-explored in this thesis. In addition to the basic 10 single body segmentation part defined in the PicStr model, another more expressive mid-level spatial representations that model higher order information between or among a group of body parts (named poselets, which is introduced by Bourdev *et al.* (2010)) are incorporated in this thesis. Poselets are pieces of human poses that are tightly similar in both appearance and configuration spaces, which are employed to capture common configurations and dependencies of multiple body parts. We explore 11 'parts' representing the mid-level poselets information covering various body part groups and the whole body. In reality, these kind of 'parts' (poselets) convey more motion information and provide more useful constraints when searching the basic single body part. Certain groups or all of these poselet representations can be selected in applications to be utilized as mid-level constraints for estimating the configuration of the basic single body part in the PicStr model.

3. To further improve the performance of the tracking framework on limbs, the tree structured PicStr model is augmented by adding more dependencies between symmetric and non-adjacent limbs, which are in fact important factors for human body balancing and coordination. In other words, we propose a framework based on PicStr that not only encodes the information based on relations between connected body parts, but also incorporates additional constraints (AdCon) between symmetric limbs. Specifically, in this thesis, four constraints are implemented between left and right upper/lower arms and legs in the complete human body model, which tend to force the left and right limbs to separate. When combing these AdCon, it is obvious that the human body model is no longer a tree structure because loops are introduced. In order to guarantee high computational efficiency while taking into account

these dependencies between body parts as much as possible, the factor graph method is used in this research to infer the pose estimates.

4. A simple estimation method on the head orientation (looking left or right) is proposed to provide instructive information for the body orientation in order to address the confusion between the left and right limbs during tracking. A head-yaw-estimation step is introduced into the tracking framework to serve as a simple yet effective tool to assist the human pose estimation. In this work, accurate estimation of the head yaw angle is not necessary. We only need a brief indication on whether the human body is roughly facing left or right. A simple skin colour detector and a set of threshold templates is hence used to roughly identify the head orientation of the tracked target. Such information is used to determine the visible side of the body, hence it is assumed that the orientations of the head and the body are consistent in a pose. Such assumption is believed to be true in most human poses.

1.2 Significance and Contributions

There are four main contributions in this thesis - (1) the strategy for dealing with scale variation issue during tracking; (2) the incorporation of higher order dependencies of multiple body parts into the image-conditioned PicStr model; (3) the inclusion of dependencies between symmetric and non-adjacent limbs; and (4) the creative use of head facing orientation to address the confusion between the left and right limbs during tracking. The contributions and their significances are detailed as follows.

1.2.1 Multi-scale strategy

The first major contribution of the thesis is to propose a strategy for the problem of tracking human motion in multiple scales in monocular image sequences. In reality, the tracked targets often moving towards or away from the camera, resulting in their sizes (scales) of their projected images to be changed within a video clip. For different motion types, two metrics are proposed for detecting and adjusting the

scale change in the tracking process. One metric is the height value of the tracked target (*Height_Metric*), which is suitable for sequences where the tracked target has generally upright postures with no obvious limb stretching. The other metric is more generic and invariant to motion types. It is named *PixelCount_Metric* and implemented by comparing the pixel counts of the foreground blobs and the detected body part bounding boxes estimated from pose estimation.

The significance of such an approach is that it enables the tracking framework to produce satisfactory tracking performance for video sequences with scale variations and a wide range of motion types. It can detect the poses even when the scale of the tracked target changes a lot, thus improve the performance of the tracking framework especially for scale-variation cases.

1.2.2 Modelling higher order dependencies of multiple body parts

The second major contribution is to combine an image-conditioned model that incorporates higher order dependencies of multiple body parts named poselets to the PicStr model. The positions of multiple body parts are often correlated in most human motion and activities. Such property has not been reflected in the generic PicStr approach hence limits the accuracy of pose estimation and tracking. A mixture of mid-level spatial representations captures multiple body parts configurations and dependencies.

The significance of this model is that it introduces some more expressive spatial constraints into the PicStr model, which is highly effective in dealing with problems such as double counting during the tracking process. Moreover, the poselet representation increases the flexibility of the PicStr approach by utilizing a set of image specific part appearance and dependencies.

1.2.3 Modelling dependencies between symmetric and non-adjacent body parts

Another contribution of this thesis is to augment the tree-based structure PicStr model with dependencies between symmetric and non-adjacent limbs, which are in fact important factors for human body balancing and coordination. Such additional constraints (AdCon) between symmetric limbs force the tracking to follow the motion biologically. All unary terms and body part dependencies utilized in this approach are incorporated in a factor graph to allow for efficient inference. The proposed model leads to a significant improvement in the accuracy of human pose tracking while being computationally efficient.

The dependencies between symmetric and non-adjacent limbs can help to correct some detection errors occurring in the generic PicStr model due to limbs drifting in some scenarios involving a cluttered background. Additionally, by introducing these dependencies, symmetric body part pairs are largely avoided to be arranged at the same location in images. Therefore, the double counting problem that often occurs within the generic PicStr model will be addressed to a large extent. This approach also improves the tracking performance by adding more flexibility to the generic PicStr model.

1.2.4 Avoiding left/right confusion with head facing orientation

The final major contribution of this thesis is to include a head orientation estimation step to address the confusion between the left and right limbs during tracking due to their similar appearances. We propose to use the head orientation (looking left or right) to provide instructive information for estimating the body orientation. A simple yet effective head orientation detection step is introduced into the tracking framework to assist the human pose estimation.

The inclusion of this step during tracking further reduces the occurrence of double counting and helps to distinguish the left and right limbs consistently. With the face orientation determined, the system can decide whether the left or right side of the

human body is definitely visible and deduce the state of its counterpart subsequently.

1.3 Thesis Structure

The rest of this thesis is organized as follows.

In Chapter 2, a review of related work in the field of human pose tracking is presented. Firstly, the existing approaches for detecting a full body within a scene are described, especially the approaches closely related to this thesis, i.e., background subtraction and detection with part-based models. Discussions on approaches for human pose estimation is presented next, which include top-down estimation approaches and bottom-up approaches. The human body part models and structure constraints for the bottom-up approaches are described. Finally, some typical tracking approaches are described with discussions on their advantages and limitation.

In Chapter 3, the general ideas of the pictorial structures (PicStr) model and a 2D human pose tracking framework based on the PicStr model are described. It includes the procedure and approach of constructing the accurate appearance model and the final tracking system. After that, the overview of the proposed complete tracking system is presented, which includes four proposed components: scale validation, poselets detection, filtering part estimates using dependencies between symmetric body parts, and head (body) orientation detection.

In Chapter 4, the strategy for multi-scale tracking is explored and two metrics are proposed and presented. Firstly, the Height_Metric is used for dealing with multi-scale tracking of motions with basically upright postures. The height of the tracked target is obtained from a full body detection. Secondly, the PixelCount_Metric is described by providing the details of the method for obtaining the pixel numbers utilized on evaluating scale values. The performances of the proposed multi-scale algorithm with two metrics are evaluated separately.

In Chapter 5, a mixture of mid-level body part representations (poselets) are included to model more expressive dependencies between/among multiple body parts. Firstly, the poselet representation is presented including poselet descriptions, poselet

detectors training and testing. Next, the unary and pairwise terms of poselets conditioned PicStr model are detailed. The tracking system with poselets incorporated is then presented. Finally, experiments are conducted to analyze the performance of the poselets-conditioned PicStr model and compare it against other state-of-the-art approaches.

In Chapter 6, two components are described for the 2D human pose tracking framework. One is the approach for modelling dependencies between symmetric and non-connected body parts. The other is the head orientation estimation. Firstly, the additional constraints (AdCon) between symmetric body parts are introduced into the PicStr model. The details on how to factorize the terms of the whole model mathematically are also presented. Then the proposed head facing orientation (HeadOri) estimation approach is described. Experiments are conducted to evaluate the performances of the tracking framework with all components incorporated: multi-scale stratedy, PicStr with additional constraints on symmetric body parts, head orientation estimation, and the poselets conditions proposed in the previous chapters.

Finally, Chapter 7 provides a summary of this thesis, its contributions and potential future directions.

Chapter 2

Background

The research goal in this thesis is to design a human pose tracker for monocular image sequences with higher accuracy. It should accurately recover the geometric location of the human body parts in each frame, track their movement independent from activities and be computationally efficient. In this chapter, we review the literature that is closely related to this thesis, i.e., human detection, pose estimation and tracking.

This chapter is organized as follows. Section 2.1 describes the existing approaches for detecting a full body within a scene, especially the approaches closely related to this thesis, i.e., background subtraction and detection with part-based models. Discussions on approaches for human pose estimation are presented next in Section 2.2. In this Section, top-down estimation approaches are discussed in subsection 2.2.1 and bottom-up approaches are presented in subsection 2.2.2 where we describe human part models and structure constraints. In Section 2.3, some typical tracking approaches are described with discussions on their advantages and limitations. Finally, a summary of the chapter is presented in Section 2.4.

2.1 Full Body Detection

Full body detection is to find people and detect the entire human as a single object, which is a fundamental requirement for most computer vision systems for human pose estimation and tracking. In most cases, a system of pose estimation or tracking requires knowing a rough region that contains a human before a specific posture can be estimated. Moreover, most of the video sequences are recorded with complex or cluttered background especially under uncontrolled environment. Thus finding people is usually a pre-processing step for human pose estimation and tracking in

order to reduce search space. For a complete tracking system, such as the work presented in this thesis, detecting the region of interest (ROI) is an important pre-processing step before analyzing poses and tracking motions. This section reviews the approaches for human detection.

Given video sequences from static cameras, background subtraction is a widely used approach for detecting moving objects. Lots of methods for performing background subtraction have been proposed (Wren *et al.* (1997); Koller *et al.* (1994); Cucchiara *et al.* (2003); Power and Schoonees (2002); Stauffer and Grimson (1999); Seki *et al.* (2003)). All of them try to effectively estimate the background model from the temporal sequence of the frames. More details are given below in Section 2.1.1. In this kind of approach, the region of interests (ROI) of an image, which in this thesis is the part representing a human body, is encoded as a whole and the common representation of the ROI is a silhouette.

Another kind of approaches for human body detection is using part-based models, which is quite popular for detecting objects from static images. Part-based human models date back to the generalized cylinder models of Binford (1971) and the pictorial structures of Fischler and Elschlager (1973) and Felzenszwalb and Huttenlocher (2005). A great number of work on part models for human detection have presented in various forms (Amit and Trouvé (2007); Burl *et al.* (1998); Crandall *et al.* (2005); Felzenszwalb and Huttenlocher (2005); Fergus *et al.* (2003); Fischler and Elschlager (1973); B.Leibe and Schiele (2007); Weber *et al.* (2000a); Felzenszwalb *et al.* (2010)). The basic premise is that human body can be modelled as a collection of local templates or models that deform and articulate with respect to one another. More details of these approaches are provided in Section 2.1.2.

Additionally, interactive segmentation is a popular technique to separate an image into two segments: 'object' and 'background'. GrabCut (Rother *et al.* (2004)) is a robust interactive approach for the segmentation problem in computer vision. It extracts foreground pixels via iterated Graph Cut optimization by defining some pixels as nodes of a graph. This methodology has been applied to the problem of human body segmentation with high success (Ferrari *et al.* (2009b)). Rother *et al.* (2004) propose to find a binary segmentation of an image by formulating an energy minimization scheme using colour information. Given a colour image I, a trimap T is defined by the user consisting three regions: T_b, T_f and T_u, which contains

background, foreground and uncertain pixels, respectively. The segmentation is defined as an array $\alpha = (\alpha_1, ...\alpha_n), \alpha_i \in 0, 1$, assigning a label to each pixel of the image indicating if it belongs to background or foreground. The final segmentation is performed using a minimum cut algorithm (Boykov and Jolly (2001); Boykov and Funka-Lea (2006); Boykov and Kolmogorov (2004)). The classical GrabCut algorithm is summarized in Algorithm 2.1.

Algorithm 2.1 GrabCut algorithm

1: Trimap T initialization with manual annotation.
2: Initialize $\alpha_i = 0$ for $i \in T_b$ and $\alpha_i = 1$ for $i \in T_u \cup T_f$.
3: Initialize background and foreground models from sets $\alpha_i = 0$ and $\alpha_i = 1$ respectively, with k-means.
4: Learn model parameters for background and foreground from pixel data.
5: Estimate segmentation: Graph-cuts.
6: Repeat step 4 and step 5, until convergence.

In the following subsections, we review the first two types of approaches that are closely related to this thesis.

2.1.1 Background subtraction

Background subtraction is often applied for video segmentation in a scene monitored by a fixed camera. The foreground is detected by comparing the current scene image against a learned background model of the scene. Most modern background subtraction algorithms do not literally perform a subtraction operation. Instead, it is a more generic process of classifying a pixel as either foreground or background according to some criteria. Thus, background subtraction is in fact a process of modelling the background and foreground and using a classifier to discriminate between the two.

Early approaches used a non-adaptive but globally thresholded background to detect foreground pixels (Wiklund and Granlund (1986)). Most researchers abandoned this method soon because of the need for manual initialization, and a variety of adaptive approaches have been developed.

A standard adaptive method is averaging the images over time and creating a back-

ground which is similar to the current static scene except where motion occurs. Ridder *et al.* (1995) model each pixel with a Kalman filter which makes the system more robust to lighting changes in the scene. Koller *et al.* (1994) improve this method and applies it to an automatic traffic monitoring system. It utilizes a selective update scheme to include only the probable background values into the estimate of the background. The Pfinder system proposed by Wren *et al.* (1997) models each pixel of the background with a single Gaussian distribution, but represents the foreground objects with a multi-class statistical model in the YUV colour space. The system reports good results for indoor scenes with an initialization step. In a different application that monitors traffic scenes, Friedman and Russell (1997) attempt to classify the pixel values into a mixture of three Gaussians, one for each class observable in their scenes, i.e., road, shadow and vehicle colour. These Gaussians are updated via incremental expectation-maximization (EM) algorithm (Neal and Hinton (1998)) to achieve a good approximation of the optimal parameters online. Cucchiara *et al.* (2003) propose to use the median value of the last n frames as the background model.

Another work trying to use a multi-colour background model per-pixel is implemented by Stauffer and Grimson (1999). They employ an adaptive nonparametric Gaussian mixture model, where the number of Gaussian distributions is usually between three and five depending on how much variation exists in the scene. This method is tolerant to small noises caused by small repetitive motions or small camera displacement. Power and Schoonees (2002) elegantly describe the theoretical framework supporting the Stauffer et al's approach and provide corrections at the same time.

KaewTraKulPong and Bowden (2002) propose an improved model based on Stauffer et al's work, with update equations, initialization method and the introduction of a shadow detection algorithm. The model has been widely utilized and become almost a standard method for background subtraction. Hence some details of this model is provided here. Each pixel in the scene is represented by a mixture of K Gaussian distributions, each with a weight w_k for Gaussian k parameterised by $N(\mu_k, \sigma_k)$. The K Gaussian distributions are ordered based on the fitness value w_k/σ_k and the first B distributions are used as a model of the background of the scene. Background subtraction is performed by marking a foreground pixel that is more than 2.5 standard deviations away from any of the B distributions. In this

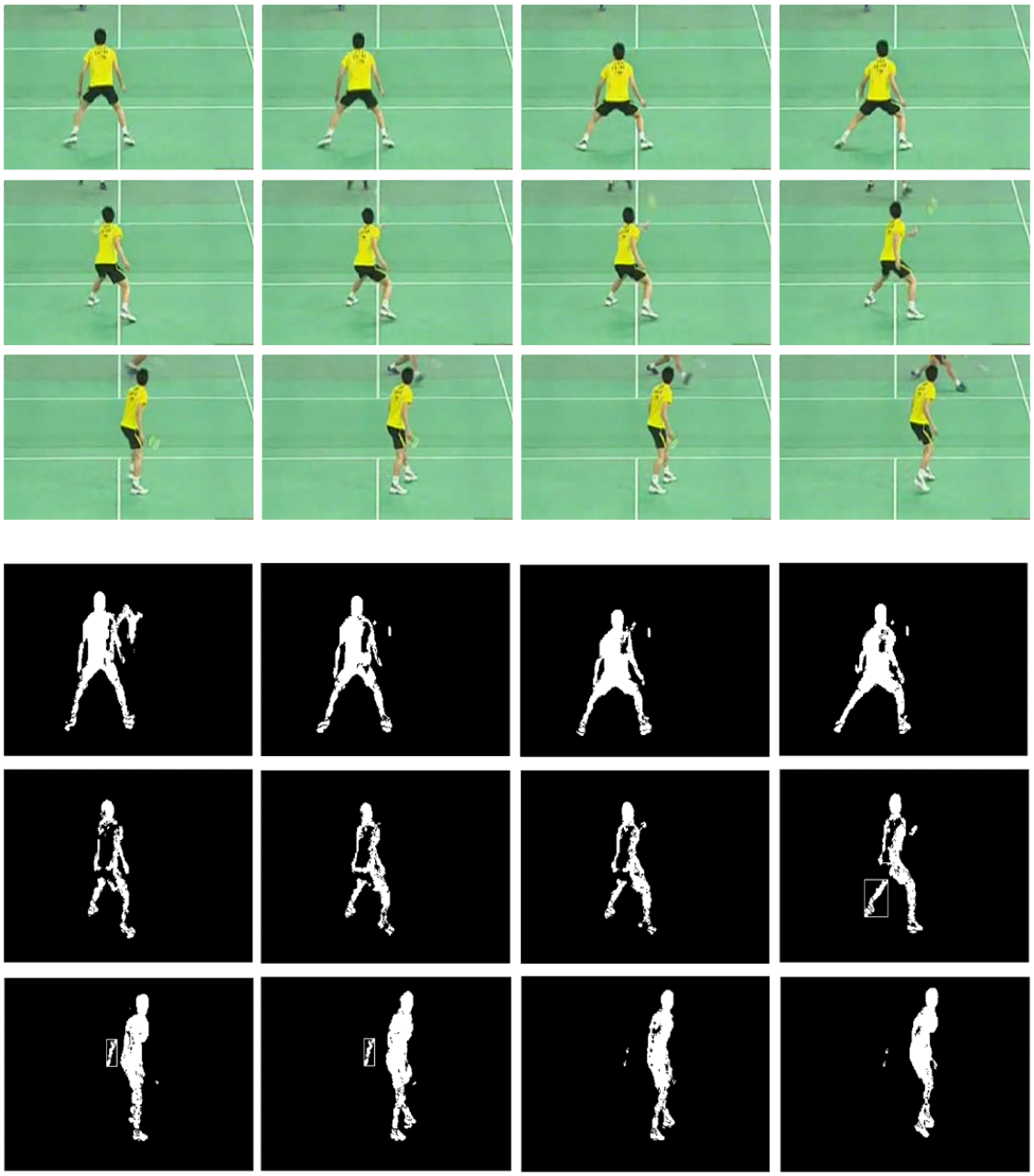


Figure 2.1: Foreground segmentation examples from the badminton sequence with the model proposed by KaewTraKulPong and Bowden (2002). The top screenshots are original frames selected from the sequence and the bottom ones are the segmented foreground blobs.

thesis, since our framework focuses on the image sequences taking by fixed cameras, this approach is chosen as a preprocessing step in order to obtain the foreground blobs during scale evaluation process. Figure 2.1 shows some screenshots of the foreground segmentation results for the badminton sequence.

In general, the obtained silhouettes contain some noise due to imperfect extractions. Also, they are somewhat sensitive to different viewpoints. However, they provide a good approximation of the human body with a great deal of information encoded and are considered sufficient as the anthropometry of the person being tracked.

2.1.2 Detection with part-based models

Object detection based on part-based models is quite efficient for static images in uncontrolled environment. In this kind of approach, each part is detected separately and a human body is detected if some or all parts are available in a geometrically configuration. Part-based approaches are popularly used for human motion tracking due to its ability to deal with great variations in appearance due to body articulation.

Felzenszwalb and Huttenlocher (2005) use the pictorial structure approach where an object is described by its parts and the geometric arrangement is captured by a set of springs connecting pairs of parts. Also, they develop efficient inference algorithms for matching the geometric information to images. Ioffe and Forsyth (2001) represent parts as projections of straight cylinders and propose efficient ways to incrementally assemble these segments into a full body assembly. Body plans proposed by Forsyth and Fleck (1997) are another representation that encodes particular geometric rules for defining deformations of local part templates. Mikolajczyk and Schmid (2005) represent body parts as co-occurrences of local orientation features. The system proceeds by detecting features, then parts and eventually humans are detected based on assemblies of parts. Fergus *et al.* (2003) and Weber *et al.* (2000b) propose the constellation models using a sparse set of locations determined by an interest point operator and arranging their geometries by a Gaussian distribution. The patchwork of parts model from Amit and Trouvé (2007) is similar as the pictorial structures approach and explicitly considers how the appearance models of overlapping parts interact.

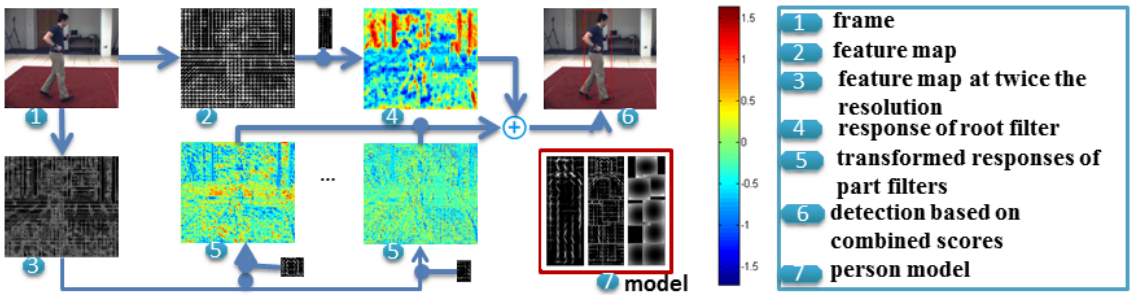


Figure 2.2: Detection process. Here only the transformed responses for the head and left-lower leg are presented. The model shown in (7) includes three components: a coarse root filter, several higher resolution part filters, and a spatial model for the location of each part relative to the root.

A particularly common approach for object detection is the star-model proposed by Felzenszwalb *et al.* (2010) based on mixtures of multiscale deformable part models. The system relies on new methods for discriminative training with partially labelled data and combines a margin-sensitive approach for data-mining with the latent Support Vector Machine (SVM). The model of the full body detector is defined by a coarse 'root' filter similar to the Dalal-Triggs filter on histogram of oriented gradients (HOG) features Dalal and Triggs (2005) which approximately covers the full body, and a series of higher resolution part filters that cover smaller parts of the human body. In implementation, the part filters capture features at twice the spatial resolution to the features of the root filter. The part filters are collected by a graphical model with deformation prior (Figure 2.2 (7)).

An hypothesis of the detection specifies the location of each filter in the model, $z = (p_0, ..., p_m)$, where p_i is the position for the i_{th} filter. At a particular position of an image, the score of a hypothesis is computed by the response of the root filter plus the sum of the transformed responses of each part filter (Figure 2.2 (1)-(6)). Note that the transformed responses are obtained from the responses of part filters minus a deformation cost that depends on the relative position of each part with respect to the root (the spatial prior).

The score of a hypothesis z can be expressed in terms of a dot product between a vector of model parameters β and a feature vector $\psi(H, z)$ as:

$$score(z) = score(p_0, ..., p_m) = \beta \cdot \psi(H, z). \tag{2.1}$$

Here, β is obtained by concatenating the root filter, the part filters, and the deforma-

tion cost weights; H is a feature pyramid; $\psi(H, z)$ is a concatenation of subwindows from the feature pyramid and part deformation features. Detecting a person in an image means to find a root location with high score and the corresponding part locations with optimal displacements:

$$score(p_0) = \max_{p_1,...,p_m} score(p_0, ..., p_m). \tag{2.2}$$

The detector is implemented with the deformable part-based model (DPM) framework, and the publicly available software Girshick *et al.* (2012) is utilized. Please see Felzenszwalb *et al.* (2010) for further implementation details.

In our project, the detection result of the ROI in an image is defined by a bounding box (bbox) $B = (x_1, y_1, x_2, y_2)$ with the upper-left and lower-right corners being at (x_1, y_1) and (x_2, y_2) respectively. To ensure our tracking system to be invariant to the size of the human body appeared in different images, the bounding box area is cropped out and resized to a patch with a normalized height h that is derived from the scale-normalized training set. In this work h is set as 200. The normalized bounding boxes form the final ROIs.

2.2 Human Pose Estimation

Human pose estimation is to find the pose parameters of all body parts depending on the human body model and image observations. There are two main classes for human pose estimation: top-down and bottom-up. In top-down approaches, a projection of the human body is matched with the image observations. Instead, bottom-up approaches find individual body parts first and then assemble them into a human body. We review these two classes in this section.

2.2.1 Top-down estimation

Top-down approaches match a projection of the human body with the image observation, i.e., searching over model parameters using a comparison between a predicted

view of the person and the image. This is often stated as an analysis-by-synthesis approach. The most common top-down approach for pose estimation is implemented by template matching. Oren *et al.* (1997) detect upright pedestrians with arms hanging at their side by a template matcher. Ramanan (2007) use a distinctive 'stylized pose' (lateral walking pose) to search for the object with an edge-based detector. Niyogi and Adelson (1994); Cutler and Davis (2000) detect walking by a simple periodic structure generated in a motion sequence.

In the top-down pose estimation approaches, a local search is often performed around an initial pose estimate Barrón and Kakadiaris (2004); Bregler *et al.* (2004). It is computationally expensive due to the high dimensionality of the pose space. Moreover, it is generally difficult to evaluate the similarity between model parameters and image regions and reliably choose the best parameters. Any optimization function in top-down methods is going to have a local extremum where a hypothesized pose lies over that region. The resulted estimates are easy to drift or become confused. To overcome these problems, a posteriori pose estimate is often found by applying gradient descent on the cost surface Wachter and Nagel (1997). The search can also be performed in the image domain. Delamarre and Faugeras (2001) use forces between extracted silhouettes and the projected model to refine the pose estimation. Gavrila and Davis (1996) take a top-down approach with search-space decomposition, where poses are estimated in a hierarchical coarse-to-fine strategy. In other words, the torso and head are estimated first and then down to the limbs and the initial pose prediction is based on a constant joint angle acceleration. The analysis-by-synthesis approach is applied in a discrete fashion, resulting in a limited number of possible solutions per joint.

Top-down estimations often need a manual initialization in the first frame of a sequence and often cause problems with self-occlusions. Moreover, errors are propagated through the kinematic chain. For example, an inaccurate estimation for the torso/head part would cause errors in estimating the orientation of body parts lower in the kinematic chain. To overcome this problem, Drummond and Cipolla (2001) introduce constraints between linked body parts in the kinematic chain, where a pose is described by the rigid displacement of each body part. This in turn yields an over-parameterized system which is solved in a weighted least-squares framework.

2.2.2 Bottom-up estimation

Bottom-up pose estimation approaches start by finding body parts and then assembling these into a human pose. They have the advantage of not requiring manual initialization and hence can be used as an initialization for top-down approaches. Moreover, temporal constraints can be used to cope with occlusions. The standard bottom-up pose estimation process takes into account part models and structure constraints (e.g., physical constraints).

2.2.2.1 Part models

Part models can be considered as fixed-size templates that are used to generate part detections by scanning over the image observation and finding patches with higher score. Intuitively, given an image I, a pixel location is defined as $l_i = (x_i, y_i)$ and the descriptor for part i extracted from a fixed size image patch centred at l_i will be written as $\phi(I, l_i)$. The part model is utilized as a comparison template with $\phi(I, l_i)$ to compute scores at all locations in an image.

Colour models

The simplest part model is directly based on pixel colour, which is based on the observation that the appearance of individual body part generally remains unchanged even in different poses. Colour models can be encoded with a histogram or a parameterized Gaussian or a mixture of Gaussians.

For example, skin colour detector works well for head part detection because a head part contains many skin pixels. Left and right limbs often look similar in appearance because clothes tend to be symmetric (Ramanan and Forsyth (2003)). Upper and lower limbs often look similar in appearance depending on the particular types of clothing worn (Tran and Forsyth (2010)). Body part proximity and symmetry in colour are essential information to prune the search space. Mori *et al.* (2004) first perform image segmentation based on appearance cues. The segments are then classified by body part locators for half-limbs and torso that are trained on image cues. From this partial configuration, the missing body parts can be derived. An additional colour information that can be utilized is background consistency, which

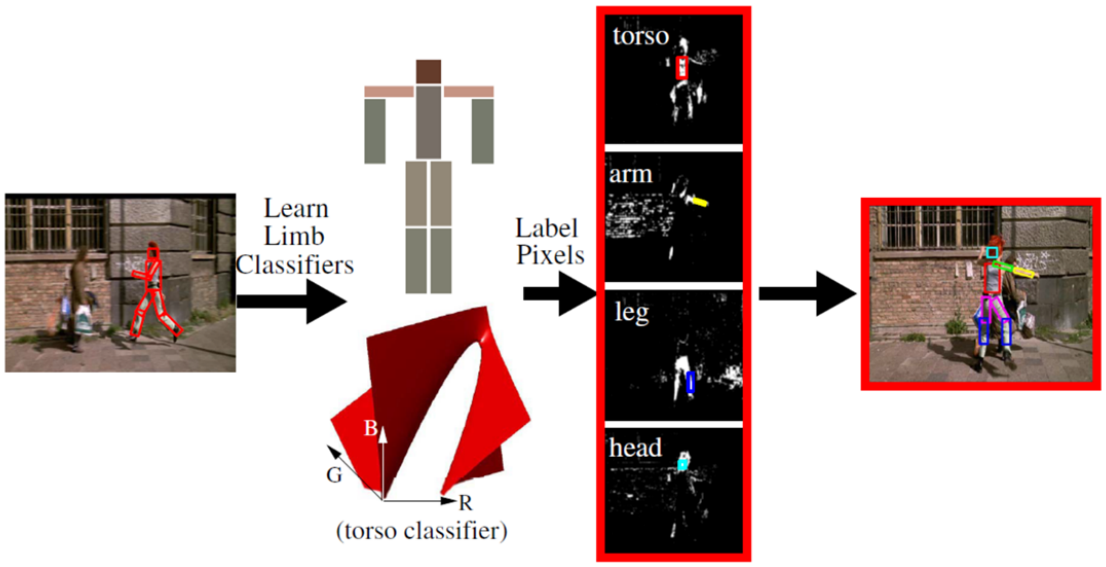
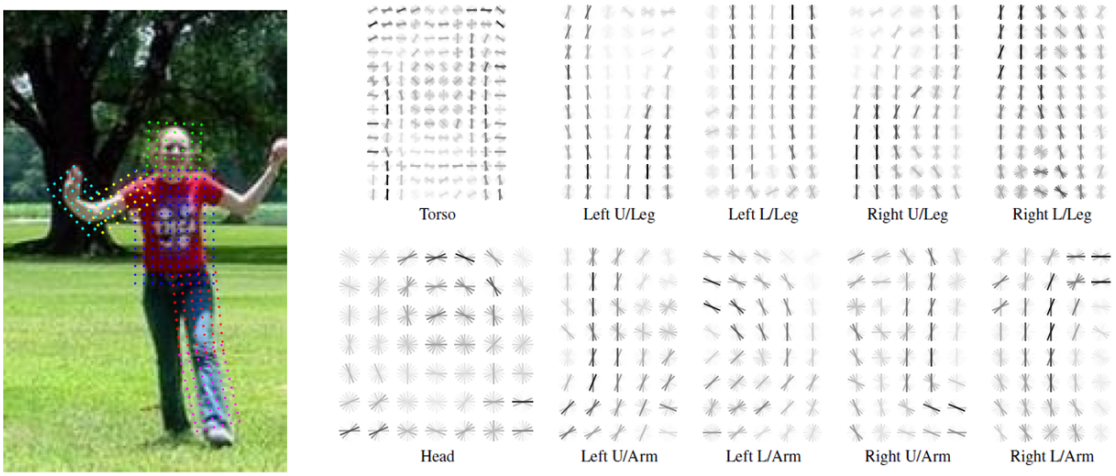


Figure 2.3: A single-scale walking pose detector is first applied on each frame to detect the lateral walking pose(left). Given the estimated limb positions from that detection, a quadratic logistic regression classifier is learned for each limb in the RGB space, considering the masked limb pixels as positives and all non-person pixels as negatives. In the middle left column, the learned decision boundary for the torso and the remaining limb classifiers are illustrated. The classifiers are then used to localize torso, head and limbs (masks shown on the middle right). These masks for candidate limbs are searched and arranged in a pictorial structure, which yields the recovered configurations on the right. (Reproduced from Ramanan *et al.* (2005))

can easily assist in separating the object from the background. Ferrari *et al.* (2008) learn some appearance model parameters by applying a foreground-background segmentation (based on 'GrabCut') on the output of an upright person detector.

Colour models are more specific to different illumination, viewpoint or clothing. They are often utilized as a complementary method to improve the performance of the systems that based on more generic models. Sometimes, the colour models are learned from detection results based on shape models. For example, an overview of the approach learning colour-based body part models proposed by Ramanan *et al.* (2005) is illustrated in Figure 2.3.



(a) HOG locations (b) HOG descriptors

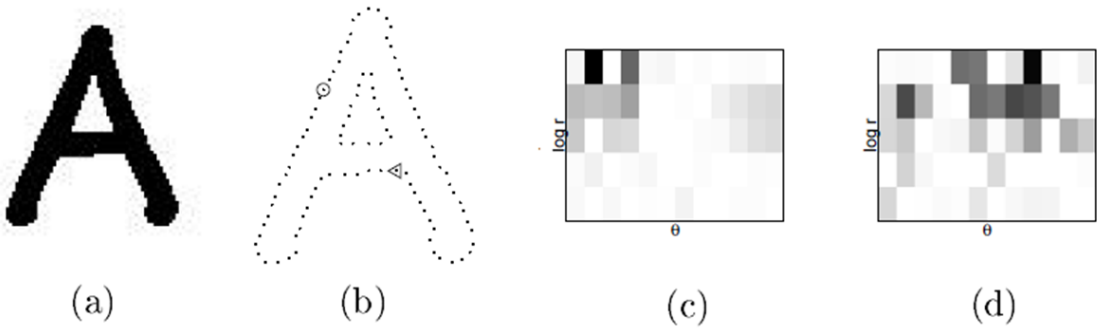
Figure 2.4: HOG descriptors. (a) Coloured blocks indicate the location and size of the HOG descriptors in ground truth position for the image shown. Each dot marks the centre of a spatial bin. (b) The extracted descriptors represent the image patch by HOG magnitude over a grid of spatial cells. Bars represent the magnitude in each orientation bin, with darker bars indicating stronger gradient. (Johnson and Everingham (2009))

Shape models

Most approaches do not directly deal with pixel data, but some features designed to be more generic and invariant to small changes and differences in lighting conditions.

Edges appear in the image when there is a substantial difference in intensity at different sides of the image location. It can be extracted robustly at low cost. E.g., the Canny edge detector (Canny (1986)) is an edge detection operator that can detect a wide range of edges in images. Edges are invariant to lighting conditions to some extent, but are unsuitable for dealing with cluttered background. They are usually located within an extracted silhouette and are often utilized as a pre-processing for object shape extraction and recognition.

Most successful shape descriptors in object recognition is the invariant ones, such as Scale-invariant feature transform (SIFT) (Lowe (2004)), shape context (Belongie *et al.* (2000)) and histogram of oriented gradients (HOG) (Dalal and Triggs (2005)). We will go through HOG and shape context descriptors below since they are particularly common representations and are used extensively in this work.



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

Figure 2.5: Shape context computation. (a) Original shape. (b) Sampled edge points. (c)(d) Shape context for reference samples marked by $\circ, \triangleleft$ in (b). Each shape context is a log-polar histogram of the coordinates of the remaining point set measured using the reference point as the origin. Here, 5 and 12 bins for $\log r$ and θ are used respectively. (Dark=large values) (Reproduced from Belongie *et al.* (2000))

Dalal and Triggs (2005) show that Histogram of Oriented Gradient (HOG) descriptors perform well for human detection compared to other existing feature sets including wavelets Viola *et al.* (2005). HOG feature extraction is based on evaluating well-normalized local histogram of image gradient orientations in a dense grid. The basic idea is that local object appearance and shape can often be characterized by the distribution of local intensity gradient or edge directions. Thus in practice, HOG extraction is implemented by dividing the image window into small spatial regions ('cells'), for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. The combined histogram entries form the representation. Specifically, image gradients are first computed at each pixel with a simple 1-D centred derivative mask $[-1, 0, 1]$. Note that for colour images, gradients are calculated separately for each colour channel, and the one with the largest norm is taken as the pixel's gradient vector. Gradients are then binned into one of the (typically) 9 orientations over local neighbourhoods of 8×8 pixels. A particularly simple implementation is to compute histograms over non-overlapping neighbourhoods. Finally, the orientation histograms are normalized by accumulating a measure of local orientation statistics over some larger spatial regions ('blocks') , e.g., 16×16 pixels. Figure 2.4 visualizes the HOG feature.

Belongie *et al.* (2000) introduce a shape descriptor, the *shape context*, for shape matching and shape-based object recognition. They argue that the shape context descriptor is tolerant to all common shape deformations. Shape context analysis

begins by converting the edge elements of a shape into a set of N feature points, which can be on internal or external contours. Then for a point P on the shape, a coarse histogram of the relative coordinates of the remaining $N-1$ points is computed. This histogram is defined to be the shape context of P. To improve the robustness of the descriptor, a log-polar coordinate system is selected and the histogram is binned with 12 equally spaced angle bins and 5 equally spaced log-radius bins. An example of shape context computation is illustrated in Figure 2.5.

Others

Convolutional neural networks (CNNs) is attracting more attentions, which is date back to 1990s (e.g., LeCun *et al.* (1998)). Along with the rise of support vector machines (SVM), it once fell out of fashion particularly in computer vision. However, Krizhevsky *et al.* (2012) rekindled interest in CNNs by showing substantially higher image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Deng *et al.* (2012); Endres and Hoiem (2010)). Girshick *et al.* (2014) use affine image warping technique to compute a fixed-size CNN input from region proposals and achieve high performance on object detection and segmentation. They name their method R-CNN: Regions with CNN features.

Shape and colour features can be combined when dealing with object detection and pose estimation. The body parts are usually described by 2D shape-based templates. Often, these templates produce many false positives, as there are many limb-like regions in an image. Lu *et al.* (2012b,a) propose to construct robust and specific colour model for each part based on the results from a generic shaped-based appearance model. Ramanan (2006) improve the detection accuracy iteratively. In the first iteration, only edges are used to locate possible body parts. Then a rough region-based colour model for each body part and the background is built from these locations and finally new locations are found using this model and the process is repeated.

2.2.2.2 Structure constraints

The human shape in 2D models are often represented as rectangular or trapezoid-shaped patches (see Figure 2.6). To compose all parts into a full body with high

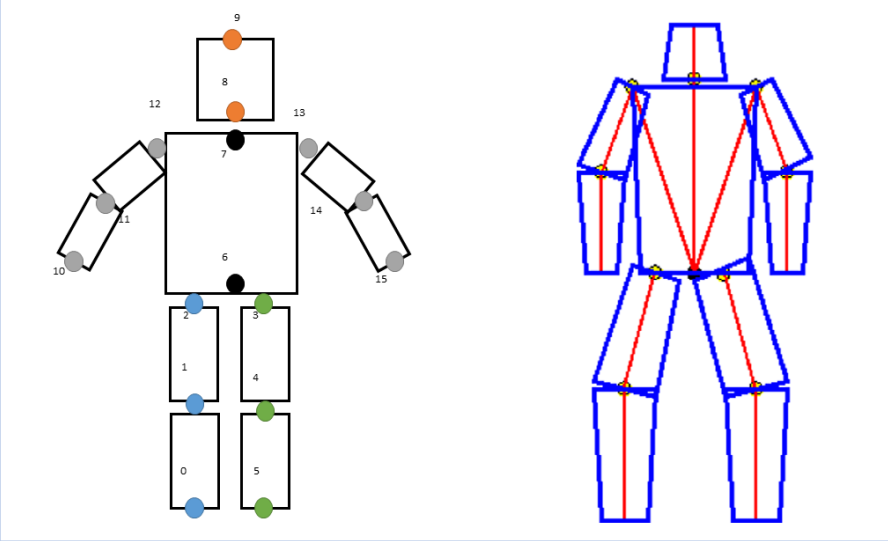


Figure 2.6: 2D human body shape model with rectangular and trapezoid patches.

accuracy, the structure model and structure constraints are of great importance. Graphical deformable models are commonly utilized for 2D human pose estimation issue because of the articulation human body.

Assuming a N-part body, with the $i-th$ part denoted as l_i and the full body configuration as L, then $L = (l_1, l_2, ..., l_N)$. A graphical model of this full body represents a collection of all parts L and a set of pairwise relationships between the parts. Denoting the image observations by D, the energy of full body configuration L defined by a graphical model is given by

$$E(L; D) = \sum_{n=1}^{N} E^u(l_n; D) + \sum_{n \sim m} E^p(l_m, l_n), \tag{2.3}$$

where the pairwise relationships between body parts are represented as $n \sim m$. According to the kind of the kinematic chin they follow, the structure of the graphical model is divided into two classes: tree structure model and non-tree structure model.

Tree structure model

Figure 2.7 (a) shows a tree based structure model for a 10-part full human body. The root part is torso and the leaves nodes are left/right lower legs/arms. It implies that the left and right body limbs are detected independently given a root torso. This tree model not only naturally captures the kinematic structure, but allows for

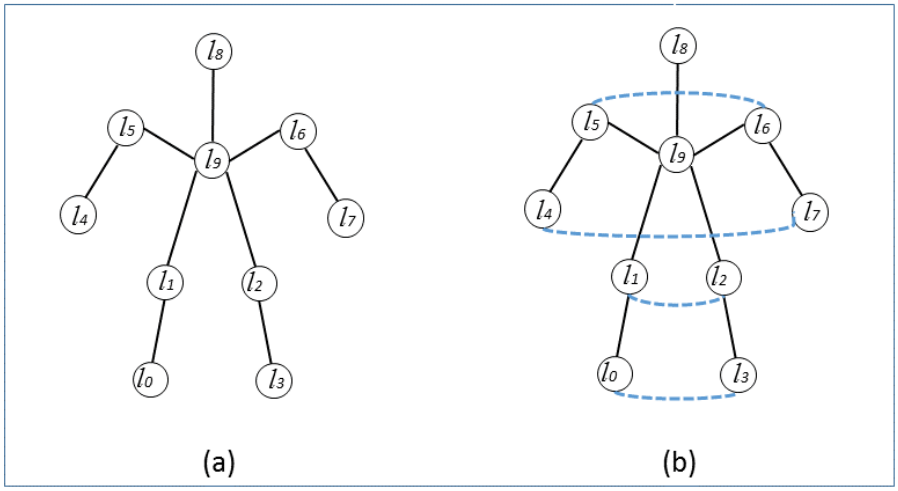


Figure 2.7: Structure model for a 10-part full human body.

efficient inference that is linearly in time in the number of nodes. Felzenszwalb and Huttenlocher (2000) optimize the tree formulation in order to perform estimation in $O(nh)$ time, where h is the number of discrete locations for each of the n parts.

The most common and successful tree based model is the Pictorial Structures (PicStr) model introduced by Fischler and Elschlager (1973) and borrowed by Felzenszwalb and Huttenlocher (2005) to find object in an image. Felzenszwalb and Huttenlocher (2005) use the concept of pictorial structures to model the coherence between body parts and all body parts are modelled with 2D appearance models. An efficient dynamic programming algorithm is used to find an optimal solution in the tree of body configurations. Ronfard *et al.* (2002) also use the pictorial structures concept but replace the body part detectors with more complex ones that learn appearance models using SVM. Song *et al.* (2003) involve feature points and inference the full configuration on a tree model. Sigal *et al.* (2003) model the spatial structure constraints between body parts as arcs and the pose estimation is also simply inference in the tree based graphical model.

Non-tree structure model

Tree models are limited by the fact that they do not capture information about relations between or among non-connected body parts by joints. Thus some important constraints such as balance and coordination has been introduced to further improve the estimation performance. For example, one can add constraints between

the arms and legs to account for balance. Figure 2.7 (b) shows a non-tree based structure model for a 10-part full human body by adding repulsive factors between symmetric legs and arms. Lan and Huttenlocher (2005) extend the tree structure with correlations between body parts. For walking, correlations between upper arm and leg swings are used, resulting in more robust pose estimations. Sigal and Black (2006a) introduce occlusion-sensitive image likelihoods, which introduces loops in the graphical model. Sigal and Black (2006b) focus on obtaining 3D poses from these 2D pose description.

When introduce more constraints between non-connected parts, the spatial structure of the full body is not a tree anymore, which causes the computational complexity of estimation to be exponentially increased in the size of the largest clique in the graph. Lan and Huttenlocher (2005) investigate a technique for adding constraint to the model while not greatly increasing the computational cost of estimation. They introduce a small number of latent variables to represent residual correlations between parts that are not captured by a tree model. Tian *et al.* (2015) introduce a factor graph to factorize all parameters that are encoded in the full non-tree model (shown in Figure 2.7 (b)). The transferred factor graph is illustrated in Figure 6.4 (b), where each part l_i is represented by a variable node (empty circle), each local function f_j is represented by a factor node (solid square), and an edge connects a variable node l_i to a factor node if and only if l_i is an argument of f_j.

2.3 Human pose tracking

Human pose tracking is achieved by estimating poses in every frame, with temporal coherence integrated between poses in successive frames. In other words, poses in every frame can be obtained by means of pose estimation algorithms and then the tracker corresponds poses across frames.

Assume the state to be inferred is X_t, the image observation is Z_t which is utilized to estimate the state, then the state and observation history are $X_{1:t} = (X_1, ..., X_t)$ and $Z_{1:t} = (Z_1, ..., Z_t)$, separately. The tracker is to estimate the probability of states (poses) over time $X_{1:t}$ or the pose X_t at time t given the sequence of observations

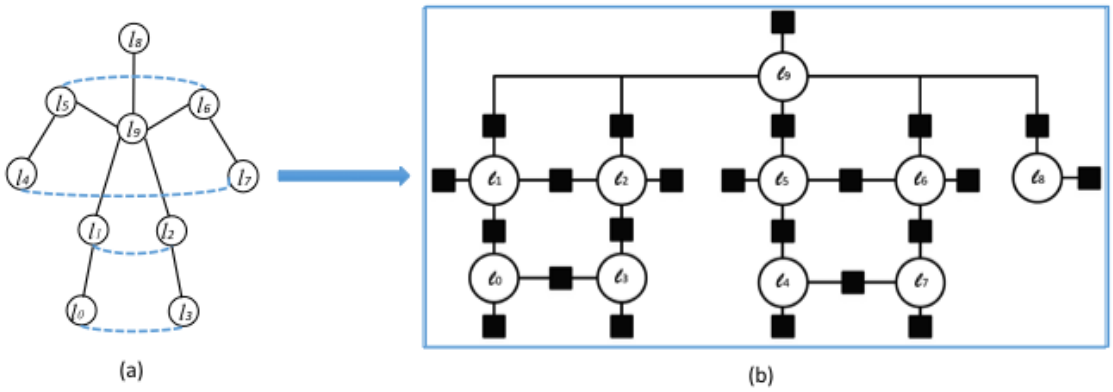


Figure 2.8: The model shown in (a) is the full body model integrated with constraints between symmetric legs and arms (the blue dash lines). The whole model is transferred to a factor graph shown in (b). Each part l_i is represented by a variable node (empty circle), a factor node (solid square) denoting each local function f_j, and an edge connecting a variable node l_i to a factor node if and only if l_i is an argument of f_j. (Tian *et al.* (2015))

$Z_{1:t}$. According to the Bayes' rule,

$$p(X_{1:t}|Z_{1:t}) \propto p(Z_{1:t}|X_{1:t})p(X_{1:t}), \tag{2.4}$$

where the first term on the right hand is the likelihood of the observations given state model, the second term is the prior of state dynamics.

$p(X_t|Z_{1:t})$ is a filtering distribution, i.e., the marginal of the posterior distribution over states conditioned on observations $Z_{1:t}$:

$$p(X_t|Z_{1:t}) = \int_{X_1} ... \int_{X_{t-1}} p(X_{1:t}|Z_{1:t}). \tag{2.5}$$

Generally, two assumptions are imposed to simplify the model. Firstly, the 1^{st}-order Markov model is used for state dynamics:

$$p(X_t|X_{1:t-1}) = p(X_t|X_{t-1}). \tag{2.6}$$

Then the sequence prior dynamics is given as

$$p(X_{1:t}) = p(X_1) \prod_{j=2}^{t} p(X_j|X_{j-1}) \tag{2.7}$$

The second assumption is that the observations are conditionally independent, thus the likelihood can be written as

$$\begin{aligned} p(Z_{1:t}|X_{1:t}) &= p(Z_t|X_t)p(Z_{1:t-1}|X_{1:t-1}) \\ &= \prod_{\tau=1}^{t} p(Z_\tau|X_\tau) \end{aligned} \tag{2.8}$$

With Equations 2.7 and 2.8, the tracker can be written as:

$$\begin{aligned} p(X_{1:t}|Z_{1:t}) &\propto p(Z_{1:t}|X_{1:t})p(X_{1:t}) \\ &= \prod_{t} p(X_t|X_{t-1})p(Z_t|X_t), \end{aligned} \tag{2.9}$$

and the filtering distribution is finally written as:

$$\begin{aligned} p(X_t|Z_{1:t}) &= \int_{X_1} ... \int_{X_{t-1}} p(X_{1:t}|Z_{1:t}) \\ &= Cp(Z_t|X_t)p(X_t|Z_{1:t-1}) \qquad (2.10) \\ &= Cp(Z_t|X_t) \int_{X_{t-1}} p(X_t|X_{t-1})p(X_{t-1}|Z_{1:t-1}). \qquad (2.11) \end{aligned}$$

where C is a constant and obviously, Equation 2.11 is a recursive form of the posterior distribution; and $p(X_t|X_{t-1})$ is the motion model.

2.3.1 Kalman filter

One of the early and most widely-used methods for tracking is the Kalman filter and its variants Bar-Shalom *et al.* (2004); Brookner (1998); McKenna *et al.* (2000); Mittal and Davis (2003). The Kalman filter is an efficient and optimal filter for tracking a target that is following a linear trajectory whose dynamics are observed via measurements that are corrupted by Gaussian noise, i.e.,

$$X_t = AX_{t-1} + \eta_d \tag{2.12}$$

$$Z_t = BX_t + \eta_m, \tag{2.13}$$

where $\eta_d \sim N(0, \sigma_d), \eta_m \sim N(0, \sigma_m)$.

The Kalman filter is a recursive algorithm that updates its estimate at each time instant t based on the observation. For a given time instant t, the algorithm includes

two steps. The first step is to make a prediction of the observation at time instance t by rolling forward the estimate at time $t-1$ using the linear dynamics. The second one is to refine the prediction into an estimate of the position at time t by choosing a point between the observation at time t and the predicted position for t. The location that this point falls depends on how confident the Kalman filter is in its prediction. As more observations are collected the Kalman filter becomes more and more certain as to the true trajectory of the target, since the filter assumes that the target will always stay on a linear course. Many approaches extend the basic Kalman filter algorithm. Bar-Shalom and Li (1993); Efe and Bonvin (2002) propose the concept of adaptive Kalman filters which aim to handle manoeuvring targets. Switching Kalman filters are proposed by Murphy (1998) to switch between different possible linear dynamics models (Stauffer and Grimson (2000)) or observation configurations.

In reality, the object motion and interactions between objects often produce complex nonlinear dynamics, so Gaussianity is not preserved and the Kalman filter is not a good choice for these cases.

2.3.2 Particle filter

A popular approach to approximately inference in non-linear tracking is Monte Carlo filter (particle filters). The distribution of the state X_t is represented by a set of particles and these particles are propagated through a dynamic model. State particles are re-weighted by evaluating the likelihood. The particle filters have proved effective for scenarios in which manual initialization is possible or there exist strong dynamic models (e.g., known motion such as walking). Isard and Blake (1998) propose a conditional density propagation algorithm (named as CONDENSATION) that is based on factored sampling and extend to apply iteratively for tracking successive images in a sequence. Figure 2.9 illustrates the iterative process as applied to sample-sets. Deutscher *et al.* (2000) develop the particle filter and modify it for searching high dimensional configurations. Based on annealing, the algorithm uses a continuation principle to introduce the influence of narrow peaks in the fitness function gradually. This algorithm is termed annealed particle filtering and can recover full articulated body motion efficiently.

Although particle filters are capable of dealing with multi-mode and non-linear cases,

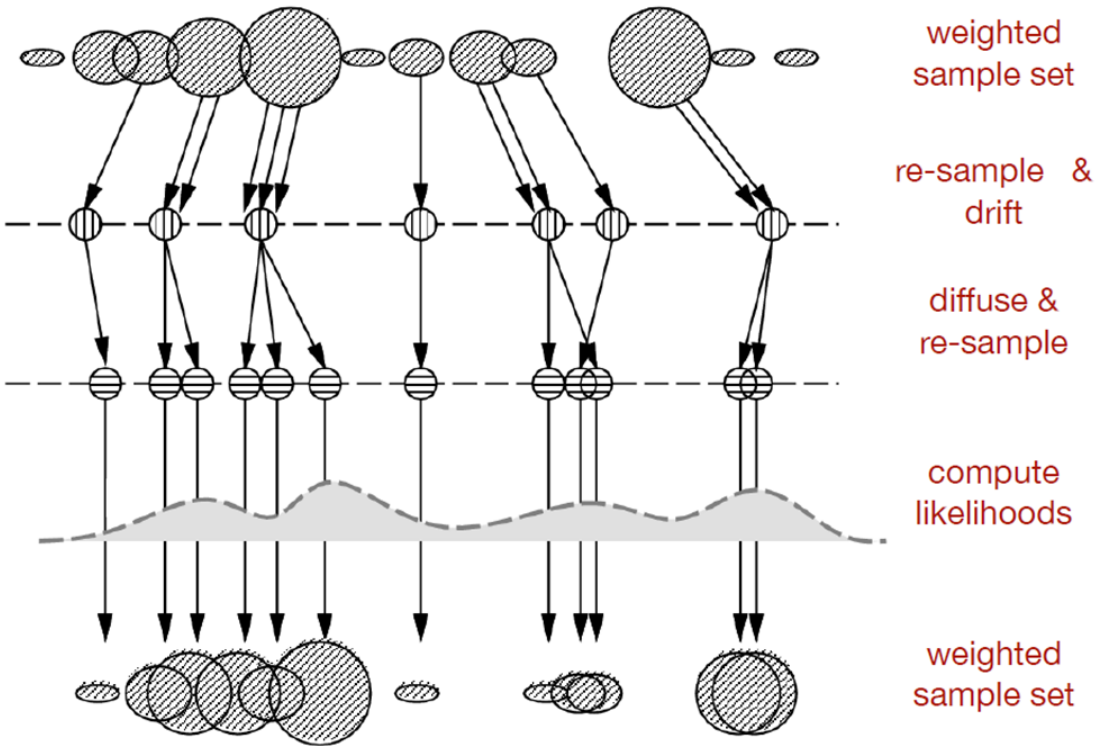


Figure 2.9: One time-step in the Condensation algorithm: three steps (drift-diffuse-measure) of the probabilistic propagation process are represented by steps in the Condensation algorithm. (reproduced from Isard and Blake (1998))

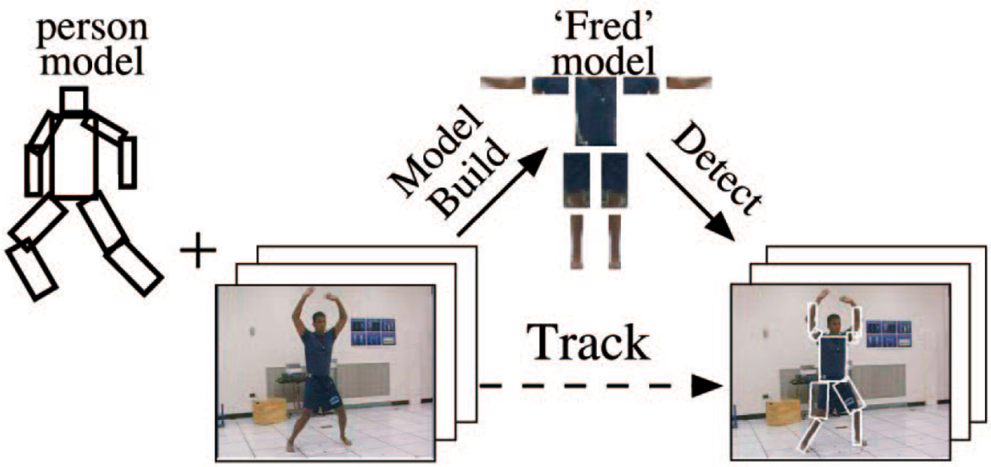


Figure 2.10: The people tracker. Initially, a detuned edge-template is used as a generic person-model. Then an instance-specific model capturing a person's appearance is built from the video data. Finally, it tracks the person by detecting that model in each frame. (Ramanan *et al.* (2007))

a big problem is that such trackers need to be manually-initialized in the first frame. Additionally, the likelihood can be highly confused in cluttered scenes. For example, there may be many image regions that locally look like a limb, which can result in particles drifted to the wrong mode. Drifting and the requirement for manual initialization seem to be related and one way to build a robust tracker is to rely less on the dynamic model but put more attention on the detection results.

2.3.3 Tracking by detection

Tracking-by-detection approaches tend to be more robust because independent pose estimation is performed for each frame, which means that the tracker can re-initialize from any frame. A post-processing stage can also be applied on the resulting pose estimates to remove the temporal noise. Ramanan *et al.* (2007) propose a tracking-by-detection framework that works in two stages: first build a model of body appearance and then track by detecting that model in each frame (see Figure 2.10).

For a N-part human body model, the tracker is a spatial-temporal model with NT parts by replicate the body model for T frames. Following a 1^{st}-order Markov model

and a pictorial structure model,

$$p(X_{1:T}^{1:N}|Z_{1:T}) \propto \prod_{t=1}^{T}\prod_{i=1}^{N} p(X_t^i|X_{t-1}^i)p(X_t^i|X_{t-1}^{\pi(i)})p(Z_t|X_t^i) \tag{2.14}$$

As a notation convention, the superscripts denote the body parts (i ranges over the torso plus left/right upper/lower arms/legs) and subscripts denote frames $t \in 1, ..., T$. The term $\pi(i)$ reperesents the parent of part i, following the tree structure. To efficiently infer this model, an attractive strategy is to estimate the pose in each frame and then optimize these poses by a local and simple motion model. Ramanan *et al.* (2007) set the local motion model by bounding the velocity:

$$p(X_t^i|X_{t-1}^i) \propto I(\left\|X_t^i - X_{t-1}^i < \upsilon_{max}\right\|). \tag{2.15}$$

Given an arbitrary video, part appearance models must initially be clothing-invariant, which can be accomplished by learning an edge-based model.

When building a human body appearance model, edge (Andriluka *et al.* (2009, 2012); Mori *et al.* (2004); Ramanan (2007)) or colour features are typically used to identify body parts. Ramanan *et al.* (2007) propose a tracking-by-detection framework that works in two stages: it first builds a model of body appearance and then it tracks by detecting that model in each frame (see Figure 2.10). The system first searches for a distinctive 'stylized pose' with an edge-based detector (they used a lateral walking pose) and then extracts the colour model from the best detected pose. However, this approach requests that the stylized pose exists clearly in the video. Ferrari *et al.* (2009a) use a generic parts detector based on edges and a pictorial structure to estimate many possible pose configurations per frame in a sequence. Within these estimates, the one with the maximum posterior at each frame is chosen as the correct estimate. All the chosen estimates are then used to construct a specific appearance model for the tracked person. A drawback of this approach is that the generic detector cannot guarantee that the maximum posteriors for all frames always correspond to the correct pose. Incorrect appearance models can be produced and consequently cause tracking errors. To overcome this issue, the approach by Lu *et al.* (2012b) clusters pose estimates from generic parts detectors across all frames and selects the largest cluster per-limb as the indicator (they call them the *'correct'* estimates) for the specific appearance model. This method shows better performance than Ferrari *et al.* (2009a) and Ramanan (2007). However, the pixels included in the so-called *'correct'* estimates in Lu *et al.* (2012b) are usually contaminated by some non-target

pixels due to a loose-fitting body model. These pixels may either be from the background of the scene or from other body parts of the tracked person. Consequently the accuracy of the specific appearance models constructed is affected, so is the performance of the pose tracking based on them. Therefore, Lu *et al.* (2012a) propose a secondary analysis step to identify and eliminate the non-target pixels from the *'correct'* estimates in order to obtain a more accurate/uncontaminated specific appearance model. Their method is successful based on the fact that the initial part detectors can provide a set of high-scoring detections. Later they use the learned appearance models to produce a dense track.

2.4 Chapter Summary

This chapter has presented a review of the literature that is relevant to this thesis. It begins with a review of current methods for finding people in a scene, including their strengths and shortcomings. Specially, two classes of approaches are presented in detail, i.e., background subtraction and detection with part-based models. The next section explores common methods for human pose estimation which is an important task in this thesis. Top-down and bottom-up approaches for human pose estimation are reviewed and compared. Part models and structure constraints are described which are two aspects in bottom-up approaches. Following this is the theoretical formulations for pose tracking. Approaches for human pose tracking are also reviewed, with particular focus on tracking-by-detection techniques due to their relevance to this thesis.

Chapter 3

Tracking System Overview

For constructing a robust tracking system, two kinds of evidence need to be balanced. One is the image evidence of the body configuration and the other is motion dynamics. A great number of previous works focus on high-level reasoning, e.g., probabilistically modelling human motions (Sidenbladh *et al.* (2002, 2000); Fablet and Black (2002)). They use the configuration in the current frame and a motion dynamics model to predict the configuration in the next frame. Such predictions can be refined using image data. Stochastic search methods, such as the annealed particle filter (APF) proposed by Deutscher *et al.* (2000) and its variants described in Kaliamoorthi and Kakarala (2013), are used widely in pose tracking due to their efficiency. However, these works generally require knowing a specific motion or establishing a motion dictionary. Complex motions are often non-linear and unpredictable hence it is generally difficult to establish a suitable motion dynamics model.

An alternative approach is to ignore motion dynamics and detect human postures in each frame, using cues such as appearance (Bai and Li (2012); Sullivan and Carlsson (2002)) or local motion (Song *et al.* (2000)) or both (Viola *et al.* (2005)). The types of motion to be tracked can be unrestricted and it is not necessary to establish or train motion models beforehand. Low-level image features are powerful for detection (Ponce *et al.* (2011)). Body appearances are generally stable throughout a video sequence because people tend not to change clothes from frame to frame. As such, recent methods are generally in favour of the tracking by detection idea. Ramanan *et al.* (2005) propose to build a discriminative appearance model from an easily detectable canonical pose detector, and use this model as a limb detector in a pictorial structure framework (Fischler and Elschlager (1973); Felzenszwalb and Huttenlocher (2005)) to detect figures in both the current and successive frames. Approaches combining detection and tracking have proven useful (Ramanan *et al.* (2007); Okuma *et al.* (2004); Ramanan *et al.* (2005); Lu *et al.* (2012b)). Among them, Lu *et al.* (2012b) propose a tracking algorithm combining edge-based (generic)

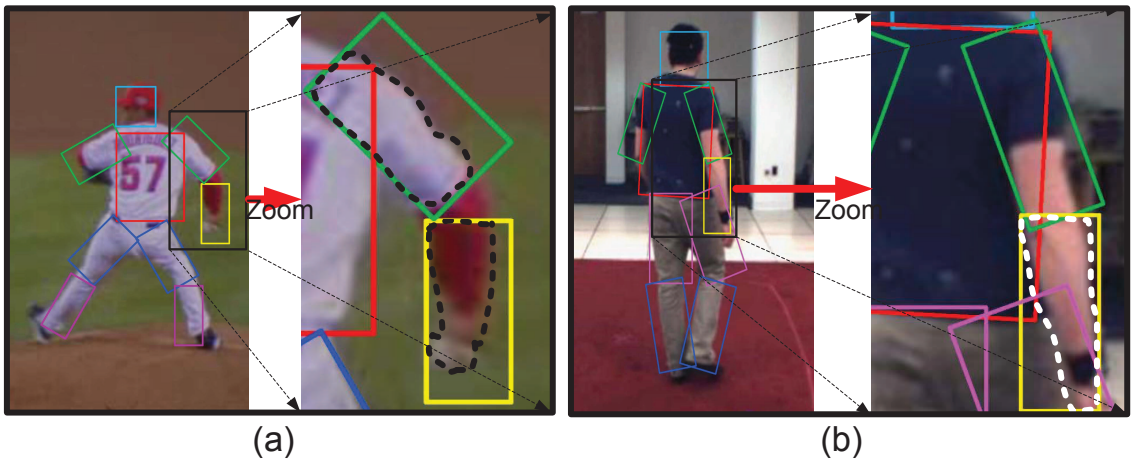


Figure 3.1: Two examples show that both target pixels and non-target pixels co-exist in the so-called *correct* estimates. Figure (a) shows that non-target pixels may come from the background. Figure (b) shows that non-target pixels may also come from other non-target body parts.

and colour-based (specific) appearances and produce very satisfactory results. They first processed the images using a set of generic human body detectors based on the shape feature, then clustered pose estimates from these detectors across all frames and selected the largest cluster per-limb as the indicator for the specific appearance model. However, the estimates that are utilized to construct the specific appearance models sometimes cannot perfectly cover the area of the body part, as shown the examples in Figure 3.1. The 'contamination' might be caused by background pixels or pixels of other body parts that exist in the clustered estimates. To overcome this issue, Lu *et al.* (2012a) propose to build an accurate and uncontaminated appearance model for tracking by eliminating such negative pixels and then track the figure's pose by detecting the constructed appearance model in each frame.

This chapter presents a tracking system with an optimal colour-based specific appearance model that is learned by pixel clustering (Lu *et al.* (2012a)). An overview of the human pose tracking framework based on an accurate specific appearance model is shown in Figure 3.2. Part (a) shows the whole tracking process, which consists of three major components: generic pose detection, accurate specific appearance model, and final pose tracker. The human body is modelled with 10 rectangles. The system firstly detects each body parts using a set of AdaBoosted part detectors based on the shape feature (generic appearance) Andriluka *et al.* (2009) for a T-frame sequence. All the detection results are then analyzed in order to extract

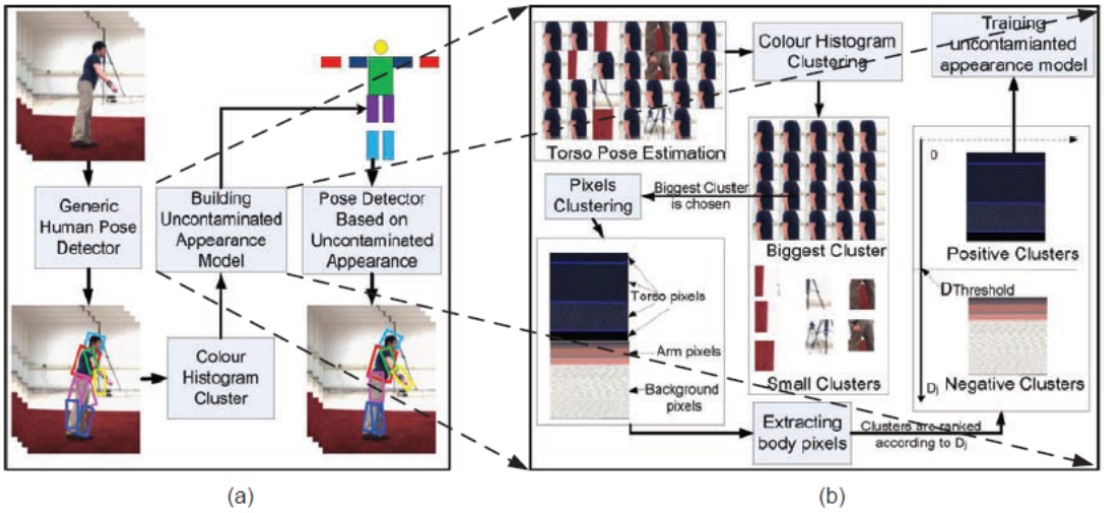


Figure 3.2: (a) Overview of the human pose tracking system. (b) Visualization about how to build an accurate specific appearance model for human pose tracking based on the results from generic human pose detection and results of the colour histogram clustering. Lu *et al.* (2012a). The process of training the torso appearance is used to illustrate the approach.

specific appearance information and the proposed algorithm for building an accurate specific appearance model is shown in Figure 3.2 (b). Finally, human poses are tracked by detecting this accurate appearance model in all frames. Similar to other 2D tracking systems, the human body is represented here by the pictorial structures model (PicStr). In practice, motion is certainly a useful cue for detection, so a simple dynamic restriction is incorporated into the tracking system, which assumes the body moves relatively slowly between successive frames. Thus a kinematic pose tracker is formed by a tree-structured graphical model (PicStr) with a simple temporal constraint incorporated.

Although this tracking system is experimented and proved that it can track people in various motions from a single view with automatic initialization and the performance is superior compared to several state-of-art 2D tracking systems, it is only for sequences in which the tracked target appears in the same or similar size. Moreover, tracking performance for the most active body part, such as the arms, is relatively low compared to the other parts of the body. Therefore, our main task in this thesis is dealing with tracking in various scales and exploring approaches to improve the tracking performance for active limbs.

Based on the system proposed by Lu *et al.* (2012a), a new 2D human pose tracking system is proposed in this thesis and detailed in Section 3.4.

The structure of this chapter is as follows. Section 3.1 provides the details of the pictorial structures model that is the base model for 2D human pose tracking. The procedure of constructing the accurate appearance model and The final tracking system proposed in Lu *et al.* (2012a) are detailed in Section 3.2 and Section 3.3. Based on this framework, the overview of the proposed tracking system in this thesis is presented in Section 3.4. Finally, some concluding marks and discussion are given in Section 3.5.

3.1 Pictorial Structures Model

A pictorial structures model for an object is given by a collection of parts with connections between certain pairs of parts. A natural way to express such a model is using an undirected graph $G = (V, E)$, where the vertices $V = \{v_1, ..., v_n\}$ correspond to the n parts, and there is an edge $(v_i, v_j) \in E$ between each pair of connected parts v_i and v_j (Felzenszwalb and Huttenlocher (2005)). In this work, we use a 10-part model for a human body: head, torso, and left/right lower/upper arms/legs. The projection of each body part is approximately modelled as a rectangle based on the assumption that a rigid human body part is more or less cylindrical. (see Figure 3.3).

The body configuration is denoted as $L = \{l_i\}$, $i \in \{0 \sim 9\}$, where $l_i = (x_i, y_i, \theta_i, s_i)$ represents the body part centred at (x_i, y_i) in image coordinate with orientation θ_i (illustrated with yellow circles and short lines in Figure 3.3 (a)), s_i is the scale factor of the body part, defined to be relative to the size of the corresponding part in the training set. In this chapter, we use the model trained from the *People* dataset gathered in Ramanan (2007), which includes annotated humans across a variety of views, articulations and activities.

Given an object appearance model C and an image evidence I, the posterior of the human body configuration L can be written as:

$$p(L|I, C) \propto p(L)p(I|L, C), \tag{3.1}$$

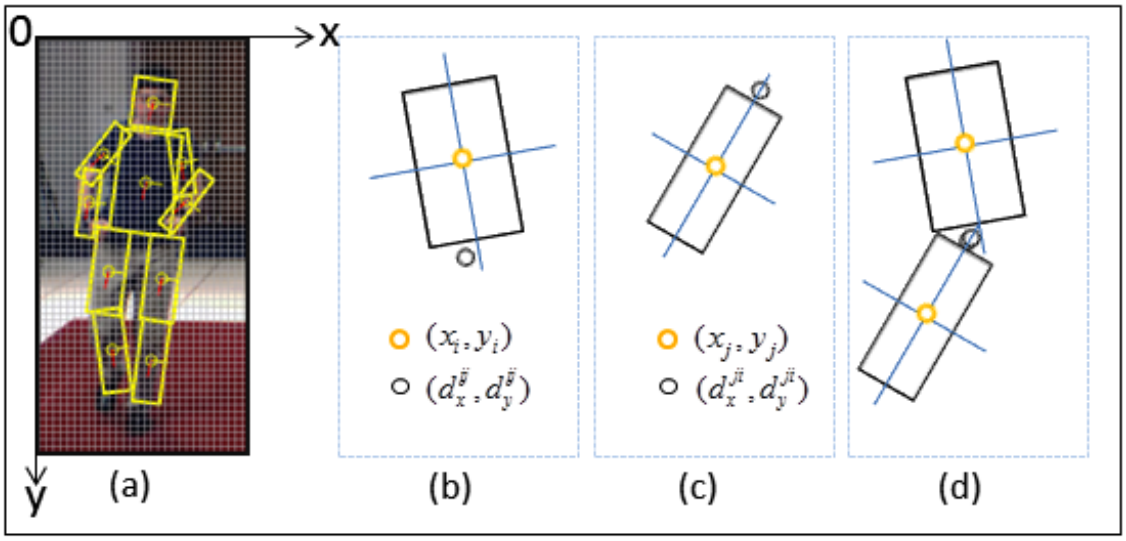


Figure 3.3: Representation and kinematic prior of an articulated object. (a) a body is represented using 10 bounding boxes configured with the centre locations in image coordinate illustrated with yellow circles. The short lines starting from the yellow circles show the local coordinate system of each part. (b) and (c) illustrate two parts in their own local coordinate systems. Two points (d_x^{ij}, d_y^{ij}) and (d_x^{ij}, d_y^{ij}), indicated by the black circles, represent the position of the joint, each in the coordinate system of the corresponding part. (d) shows the ideal configuration of the connected parts, i.e., the two joint positions overlap.

where $p(I|L,C)$ is the likelihood of the image evidence, and $p(L)$ represents the geometric relations between the connected parts called the configuration prior. Each body part is assumed to be conditionally independent given the part configuration l_i and the part appearances C_i. The likelihood $p(I|L,C)$ is decomposed into the product of single part likelihoods. In order to enable exact and efficient inference, the configuration prior is restricted to form a tree structure.

The configuration prior. For an articulated object, pairs of parts are connected by flexible joints. The first term in the right-hand side of (3.1) is the configuration prior $p(L)$ representing the constraints between the connected parts in the probabilistic form. A tree-structured graphical model is used to encode such constraints (i.e., the kinematic dependencies between body parts).

In a tree graphical model, each child node only depends on the immediate parent node, thus the prior on part configurations can be factorized as

$$p(L) = p(l_0) \prod_{(i,j)\in E} p(l_i|l_j), \tag{3.2}$$

where E is the set of all edges in the kinematic tree, l_0 denotes the root node, which is the torso in our case, similar to many other approaches based on PicStr. The prior for the root node configuration $p(l_0)$ is assumed to be uniform to allow for a wide range of body postures.

A pair of connected parts l_i and l_j is illustrated in Fig. 3.3 (b), (c) and (d). The position of the joint is specified by two points (d_x^{ij}, d_y^{ij}) and (d_x^{ji}, d_y^{ji}), represented by black circles in Figure 3.3 (b) and (c), each in the local coordinate system of the corresponding part. These points overlap in an ideal configuration as indicated in Figure 3.3 (d).

The joint probabilities between the dependent child body part l_i and its immediate parent part l_j are denoted as pairwise terms $p(l_i|l_j)$, as in Felzenszwalb and Huttenlocher (2005), which are modelled by Gaussian distributions allowing for efficient inference. As pointed out in Felzenszwalb and Huttenlocher (2005), the spatial distribution between connected parts is possibly well captured by a Gaussian distribution in a transformed space even when it is not Gaussian distribution in the image coordinates. The part configuration $l_i = (x_i, y_i, \theta_i, s_i)$ is transformed into the

local coordinate system of the joint between the two parts by the transformation:

$$T_{ij}(l_i) = \begin{pmatrix} x_i + s_i d_x^{ij} cos\theta_i - s_i d_y^{ij} \sin\theta_i \\ y_i + s_i d_x^{ij} sin\theta_i + s_i d_y^{ij} \cos\theta_i \\ \theta_i + \widetilde{\theta}_{ij} \\ s_i \end{pmatrix}. \tag{3.3}$$

where $d^{ij} = (d_x^{ij}, d_y^{ij})^T$ is the joint position between parts l_i and l_j represented in the coordinate system associate with part l_i, and $\widetilde{\theta}_{ij}$ is the relative angle between the two parts. The $p(l_i|l_j)$ can now be represented as:

$$p(l_i|l_j) \propto \mathcal{N}(T_{ji}(l_j) - T_{ij}(l_i)|\mu^{ij}, \Sigma^{ij}), \tag{3.4}$$

where T_{ji} is the transformation that maps the configuration of the parent part l_j to the local coordinate system of the joint between parts i and j, μ^{ij} is the relative orientation, and Σ^{ij} is the covariance matrix of the parts. All these parameters in the kinematic tree prior can be learned from the training data.

Likelihood model. The other important part in the pictorial structures model is the likelihood $p(I|L, C)$. For simplicity, assuming that each body part is conditionally independent given the part configuration l_i and the part appearances C_i, the likelihood $p(I|L, C)$ is decomposed into the product of single part likelihoods:

$$p(I|L, C) = \prod_{i=0}^{N} p(I_i|l_i, C_i). \tag{3.5}$$

In the implementation of the system, a set of *Boosted part detectors* are adopted, which are pre-trained by Andriluka *et al.* (2009) to model the part likelihoods. The appearance of each body part C_i is represented by a shape context descriptor and trained with an AdaBoost classifier to predict the presence of a part. They are generic appearances describing shape features and have proven to be an effective generic approach for human body detection.

Using (3.2) and (3.5), the posterior (3.1) is factorized as

$$p(L|I, C) \propto p(l_0) \prod_{i=0}^{N} p(I_i|l_i, C_i) \prod_{(i,j)\in E} p(l_i|l_j). \tag{3.6}$$

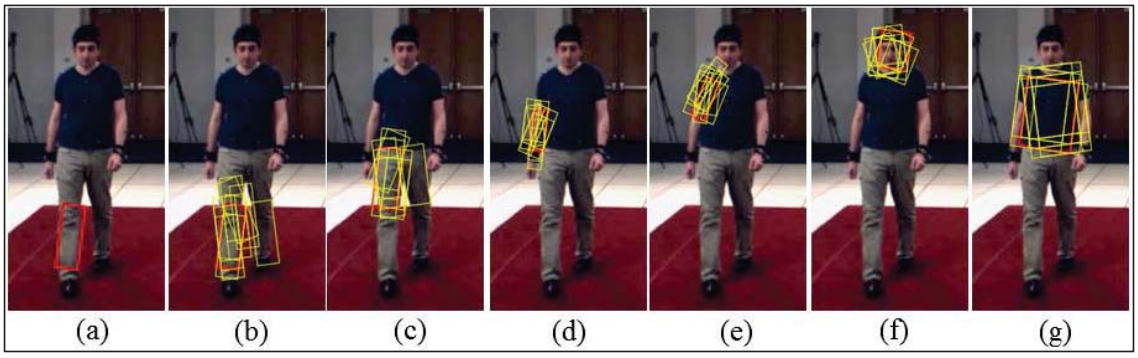


Figure 3.4: Samples of the top estimates for specific body parts. The primary estimate is illustrated in red (such as the red bounding box in (a) and in other images) and the alternative estimates for specific part are shown with yellow color((b)-(g)).

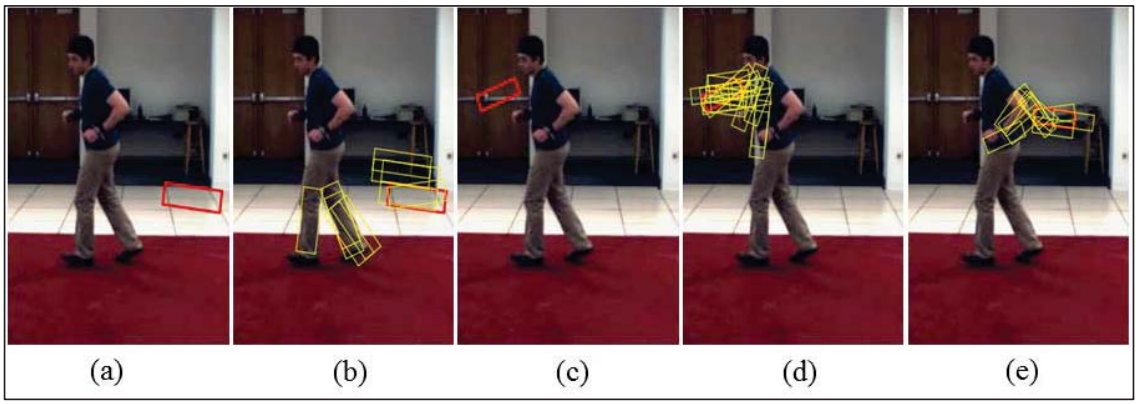


Figure 3.5: Samples of the wrong estimates in the *primary estimate* set. The primary estimate is illustrated in red bounding box (b-box) and the alternative estimates are bounded in yellow. The b-box in (a) and (c) are the primary estimates for the left-lower-leg and left-upper-arm separately, but they are wrong estimates due to the 'arm-like' shape and other noises in the images. On the contrary, the *probable alternative estimates* (in yellow colour) in (b) and (d) include the correct estimate. Similar examples for the right-lower-arm are shown in (e).

Generic part estimates. With the model described above, human postures can be detected by finding the most probable configuration of each body part. Due to each possible configuration l_i being viewed as an probabilistic estimate corresponding to $p(l_i|I, C)$, each part detector in fact generates multiple estimates, samples of which are shown in Figure 3.4. For one part in each frame, all estimates can be identified as two types: the optimum estimate and a set of probable alternative estimates. The optimum estimate called *primary estimate* (p_m) is the most probable location for part m based on the generic appearance model. Beside the optimum estimate for part m, a set of sub-optimal estimates called *probable alternative estimates* are also obtained by sampling the top N posteriors except for the maximum posterior (in Lu *et al.* (2012b), N is selected to be 30), denoted as Q_m.

For a T-frame sequence, the primary estimates for part m in all frames are denoted as a set S_m, and $S_m = \{p_m^1, p_m^2, \ldots p_m^T\}$, where the superscript is the index of image frames in the sequence. Because the part detectors rely on generic shape features, not all the optimum candidates in S_m are guaranteed to be the correct estimates. For example, as shown in Figure 3.5, the 'limb-like' shapes or other noises around a limb often confuse this shape-based part detectors. Fortunately, experiments show that, for a sequence, the colour information provides a convenient and effective supplement for avoiding such erroneous estimates as much as possible Lu *et al.* (2012b).

3.2 Building an Accurate Specific Appearance Model

In this section we present the approach for building an accurate specific appearance model based on body part estimates from the generic pose detector. As mentioned above, there are both correct and false estimates in the primary estimate set S_m. If the correct and false estimates can be separated and the correct estimates identified, the colour information of the body part could be extracted, thus an appearance model specific to the tracked target can be constructed.

Based on this reasoning, the system first follows the approach proposed by Lu *et al.* (2012b) to cluster the colour histogram of primary estimates across T frames in a sequence (each body part clustered separately), under the assumption that the dominant cluster represents the *'correct'* estimates of the tracked target. Lu *et al.*

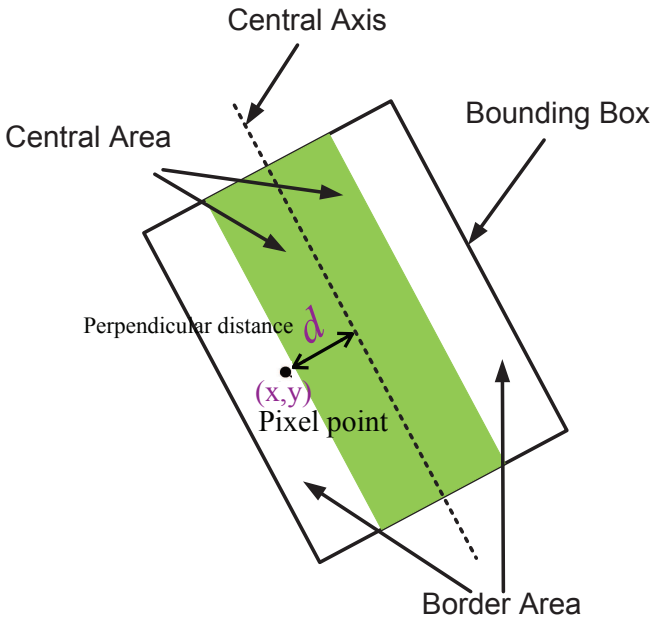


Figure 3.6: An estimate is represented by a bounding box in an image. The green area represents the central area and the white area represents the border area of the bounding box. It also illustrates how to compute a perpendicular distance d of the pixel (x,y) with respect to the long axis of the bounding box.

(2012b) have proven that this assumption is reasonable as long as enough correct estimates are obtained (more than any sets of false estimates that fall randomly on the same colours).

However, even in the so-called correct estimates, they would definitely include some non-target pixels, i.e., the estimates are contaminated, which may either be from the image background or from other body parts of the tracked person, as in the examples shown in Figure 3.1. In other words, even with the *'correct'* estimates, there is a possibility that the colour features they provided might not be good enough for the representation of some particular body parts, especially if the so-called 'contamination'is not trivial. A secondary analysis procedure is hence introduced to identify and eliminate the non-target pixels from the *'correct'* estimates, rather than using them directly to build the specific appearance model for a body part, as an attempt to obtain a more accurate/uncontaminated specific appearance model. Two assumptions are utilized in the procedure. Firstly, within a bounding box identified as a correct estimate, the number of body pixels (the target pixels) is expected to exceed the number of non-target pixels, which is reasonable given that the detection

of Lu *et al.* (2012b) is relatively accurate and the body model is not too loose a fit. Secondly, as illustrated in Figure 3.6, body pixels tend to locate in the central area of the bounding box and most of the non-target pixels would locate along the border.

In Section 3.2.1 we briefly explain how to cluster the colour histograms of the primary estimate set S_m. The procedure described in Section 3.2.2 is an extension of the clustering procedure which aims to improve the accuracy of the appearance model obtained by addressing the 'contamination' problem.

3.2.1 Colour histogram clustering

For each estimate p_m^t in the primary estimate set S_m, a corresponding image patch (a rectangular area) A_m^t would be specified by the estimate. In the proposed framework, the size of the image patch A_m^t could vary due to the scale variation between different frames. We firstly resize the image patches for each body part in T frames according to the corresponding scale values which are determined by the strategy described later in Chapter 4 and obtain a series of new image patches $\{a_m^t\}$ with the same size. The patch a_m^t bounds the position of limb m and is represented by the colour histogram of the pixels (a feature vector (v_m^t)) in that patch. Specifically, the histogram is calculated by projecting the pixels of an image patch onto the L, a, b axes in the CIE *Lab* colour space separately, with each of the three colour channels divided into 10 evenly distributed bins. A feature vector (v_m^t) is hence created consisting of a 30-bin Lab colour histogram. The set of the primary estimates (S_m) for part m is then transformed to a set of feature vectors $\{v_m^t\}_{t=1}^T$.

These image patches with feature vectors $\{v_m^t\}_{t=1}^T$ are then clustered by the mean-shift procedure (Comaniciu and Meer (2002)), a nonparametric density estimation technique. The set of colour histogram feature vectors $\{v_m^t\}_{t=1}^T$ can be viewed as T data points, $t = 1, \ldots, T$ in the 30-dimentional space. The mean-shift procedure is an iterative scheme in which we find the mean position of all feature points within a hypersphere of radius h, re-centre the hypersphere around the new mean, and repeat until convergence. The constant h can be adjusted to control the merging of the clusters. Points in one cluster tend to represent similar colours. The points in the biggest cluster are then identified as the *'correct'* estimates, and a subset $U_m \subseteq S_m$

is used to denote this set of correct estimates

$$U_m = \{p_m^{t_n}\}_{n=1}^{N}, \tag{3.7}$$

where n indexes the subset of times t_n which are considered correct estimates. For each element $p_m^{t_n}$ in the subset U_m, a corresponding image patch $a_m^{t_n}$ would be obtained.

3.2.2 Improving the appearance model by de-contamination

In order to identify the target (body) pixels for part m, we further analyze the pixels specified in the set U_m. A group of colour vectors $Y_m^{t_n}$ are constructed to represent one image patch $a_m^{t_n}$. Again, the colour vectors are extracted in the CIE Lab colour space,

$$Y_m^{t_n} = \{K_i\}_{i=1}^{w*h}. \tag{3.8}$$

where i corresponds to one pixel in the image patch $a_m^{t_n}$, and w, h represent the width and the height of the image patch $a_m^{t_n}$.

Let

$$Y_m = \cup_{n=1}^{N}\{Y_m^{t_n}\}. \tag{3.9}$$

All pixels specified by estimates in U_m are transformed into colour vectors in Y_m.

When (3.8) is substituted into (3.9), the set Y_m can also be denoted as

$$Y_m = \{K_i\}_{i=1}^{L}, \tag{3.10}$$

where $L = w * h * N$ and N is the size of the subset U_m.

Pixel clustering

After all pixels specified by the estimates in U_m are represented by colour vectors in Y_m, pixel clustering can be implemented.

The mean shift algorithm is applied for pixel clustering procedure to obtain the modes of the colour vectors Y_m. According to the mean shift algorithm, a set of l modes $Y_{m_c} = \{K_{c_j}\}_{j=1}^{l}$ are generated, which represent the local maxima points,

where $l \ll L$. Thus each vector in Y_m is attached to a specific mode and the set of colour vectors Y_m is partitioned as

$$Y_m = \cup_{j=1}^{l} M_c^j \tag{3.11}$$

where M_c^j is one of the clusters which corresponds to the mode K_{c_j} in the set of l modes Y_{m_c}.

Identifying target pixels

The goal here is to divide the subsets M_c^j into positive subsets containing the pixels from the body part being modelled (target pixels) and negative subsets containing the pixels from background and other body parts (non-target pixels). Figure 3.7 shows two examples of the results from the pixel clustering, one is for the lower left leg (a single-colour body part) and the other is for the upper left arm (a multi-colour body part). The pixel clustering procedure is performed on them using the approach described in Section 3.2.2. In the right column of Figure 3.7 (a), two clusters are obtained for the lower-left leg , where Cluster #1 represents the target pixels and Cluster #2 represents the non-target pixels clustered from all pixels in the patches shown in the left column of Figure 3.7 (a). In Figure 3.7 (b), it can be clearly seen that the target pixels are in two different colours, i.e., the navy colour and the skin colour, which are separated into Cluster #1 and Cluster #2 respectively shown in the right column of Figure 3.7 (b). Cluster #3 illustrates the non-target pixels in the patches of the upper left arm.

How to determine which cluster (or clusters) represents the target pixels? It is reasonably assumed that the colour of the pixels along the central axis of a body part should be most representative of the target pixel colours for that body part since human body parts are typically vertically arranged (e.g., clothes colouring, etc.). Besides, it is expected that most of the target pixels are located in the central area of its corresponding bounding box, and the number of target pixels is larger than the number of non-target pixels in a *'correct'* estimate. An average distance D_j for a subset M_c^j is defined to identify the positive subsets. A colour vector K_i in Y_m corresponds to a pixel (x, y) in certain image patch $a_m^{t_n}$ which is enclosed by a bounding box. As shown in Figure 3.6, for a pixel of coordinate (x, y), a perpendicular distance d with respect to the long axis of the bounding box can be obtained. Thus every K_i in Y_m would correspond to a perpendicular distance d. For

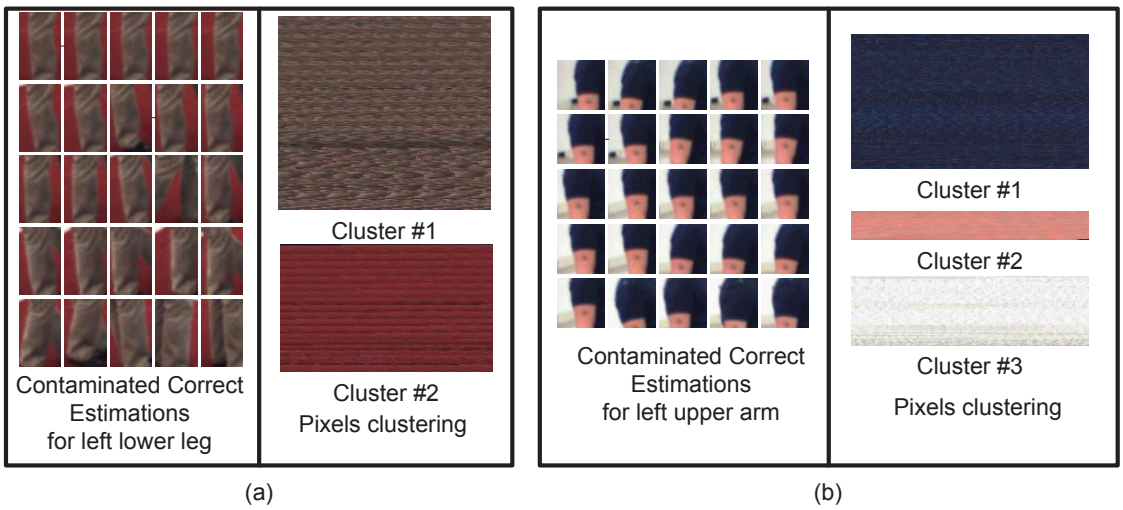


Figure 3.7: Two examples for pixel clustering. (a) shows the result of pixel clustering for the left lower leg (single-colour body part). (b) shows the result of pixel clustering for the left upper arm (multiple-colour body part).

a subset M_c^j, an average distance D_j can be defined as

$$D_j(M_c^j) = \frac{1}{S}\sum_{s=1}^{S} d_s, \tag{3.12}$$

where S is the size of set M_c^j.

If D_j is small and the size of M_c^j is big, M_c^j is more likely representing the target pixels. The elements in set $\{M_c^j\}_{j=1}^{l}$ can be sorted from small to large according to the average distances. A distance threshold d_{thres} is used to separate the positive subsets from the negative subsets. d_{thres} is set relative to the half body part width (d_m), i.e., $d_{thres} = \mu * d_m$. If the average distance D_j for M_c^j is bigger than d_{thres}, M_c^j is removed; otherwise it is retained. Among the positive subsets, the subsets whose number of elements is far less than the others are discarded. The remaining positive subsets $\{M_c^{j^r}\}_{r=1}^{R}$ are retained to be used for building the accurate appearance model.

Building the accurate appearance model

With the positive subsets (target pixels subsets) $\{M_c^{j^r}\}_{r=1}^{R}$ identified, the accurate/uncontaminated appearance model can be built through training a set of target-pixel classifiers. The number of classifiers is determined by the number of the positive

subsets. Although complex classifiers, such as SVM or quadratic logistic regression classifiers, can be learned, simpler Gaussian classifiers turn out to be sufficient for our system.

Given the data in $\{M_c^{j^r}\}_{r=1}^{R}$, each $M_c^{j^r}$ is characterized by a Gaussian and modelled into a single Gaussian. To classify an unknown vector x, its likelihood probability is defined as

$$p(x|M_c^{j^r}) = \mathcal{N}(x, \mu_m^{j^r}, \Sigma_m^{j^r}). \tag{3.13}$$

A threshold γ is used. If $p(x|M_c^{j^r})$ is more than γ, the vector x is identified as representing the target pixels. Otherwise, the vector x is classified as representing non-target pixels. In this way, all target pixels in each frame can be classified and labelled.

Labelling target pixels

As stated above, the target-pixel classifier or classifiers can be built for each body part, which can then be applied to check the pixels through all frames in a sequence. Consequently, for each body part, a binary image (called *mask image*) can be generated for every frame in the sequence. If a pixel in a frame with coordinate (x, y) is a target pixel, the corresponding pixel in the mask image with coordinate (x, y) is marked as 0, otherwise marked as 1.

An experiment is conducted on the Combo sequence from the HumanEva dataset (as shown in Figure 3.8 (a)) to test the proposed target-pixel classifiers. In this experiment, the learned target-pixel classifiers for each body part are used to mark the target pixels.

Sample results are shown in Figure 3.8. It can be seen that the body part pixels are clearly identified and labelled. Although a small number of noise pixels are wrongly marked as the target pixels, they have very small impact on localizing the part. Due to the symmetrical colouring of typical clothing, the symmetric body parts such as left and right arms/legs usually appear in similar colours. For example, the legs including left/right lower/upper legs have the same colour, so four different target-pixel classifiers actually represent the same colour, as shown in Figure 3.8 (c). In these cases, the pictorial structures is used to resolve the confusion between body parts. The spatial relations between the body parts are defined in the pictorial

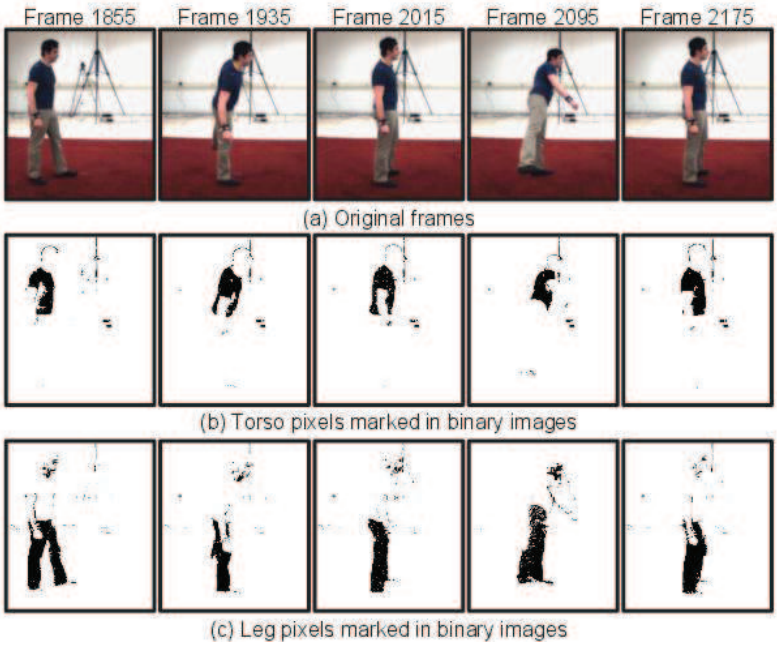


Figure 3.8: Several examples of marking target pixels for body part using the learned Gaussian classifiers. The frames shown in this figure are representative and typical in Combo sequence (HE_I_S2_Combo_2_C2) from HumanEva dataset Sigal *et al.* (2010).

structure, which helps to determine which target pixels should belong to which body part. Section 3.3 describes how to use these mask images during the tracking process.

The separation of target and non-target pixels in the *'correct'* estimates helps to improve the accuracy of the constructed specific appearance model.

3.3 Tracking with the Accurate Appearance Model

After the accurate appearance model is obtained using the algorithm of Section 3.2, poses are tracked by detecting this model in all frames. The pictorial structure framework as described in Section 3.1 is used.

For body part m in frame t, given a set of configurations, filtering is used to select the candidates which agree with the specific accurate appearance model and generate a set G_m^{*t} called *concentrated set*. Specific to this system, the trained appearance model (a set of target-pixel classifiers) can label the target (body part) pixels in each frame, thus the criterion for filtering can be defined as the correct rate of target pixels appearing in each patch of the candidates. Specifically, for a candidate detection (a bounding box a_m^t for body part m in frame t), given the width and length (w_m and l_m) and a mask image b_m^t obtained from the pixel labelling process described in section 3.2.2 (Figure 3.8), a mask patch p_m^t can be extracted by intersecting the mask image b_m^t with the bounding box a_m^t. The specific appearance feature is recorded in the mask patch. In order to score the candidate detection, a final response is set to evaluate the ratio of target pixels in this detection. If a pixel in the mask patch p_m^t is a target pixel (the pixel value is 0), a positive response ($r(i,j) = 1$) is obtained; otherwise a negative response ($r(i,j) = 0$) is obtained. Finally, the response of the specific appearance for a candidate detection is defined as

$$L_m(a_m^t) = \frac{\sum_{i=1}^{w_m} \sum_{j=1}^{l_m} r(i,j)}{w_m \times l_m}. \tag{3.14}$$

Given any detection a_m^t for body part m in frame t, if $L_m(a_m^t) > 50\%$, it is retained otherwise it is discarded thus generating a subset G_m^{*t} in which each element satisfies

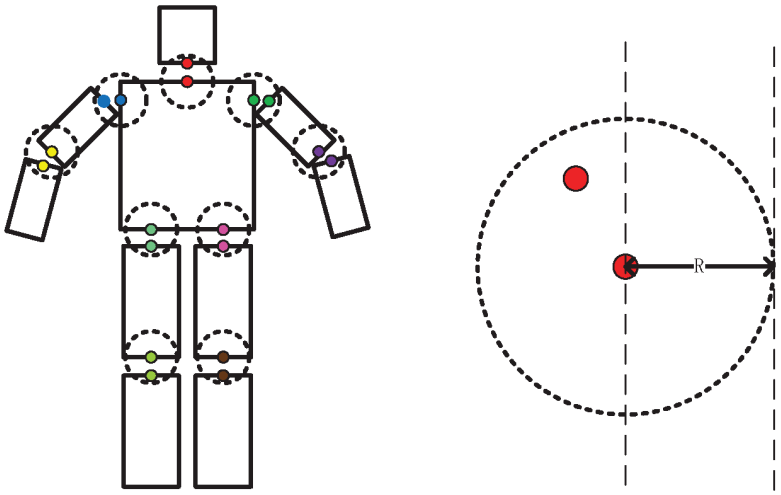


Figure 3.9: The endpoints for all body parts in the human body model.

the specific accurate appearance model. Each estimate in the *concentrated set* G_m^{*t} is a candidate for the final estimate with spatial search.

Spatial search is conducted to find an estimate from set G_m^{*t} that best fits with the neighbouring body parts. The estimate for part m is required to satisfy the spatial constraint of its parent part n. In the pictorial structures model, a tree-graphical model is adopted to model a human body consisting of a set of body parts, which has been described in Section 3.1. Figure 3.9 shows a ten-part human body connected by nine pairs of hinge points. The spatial constraint is defined as follows. For any pair of hinge points, the hinge point of the child part is required to exist within the circle of radius R centred at the hinge point of the parent part. Formally, given the coordinate of the hinge point $\{x_{h_n}, y_{h_n}\}$ in the parent (body part n), the coordinate of the hinge point $\{x_{h_m}, y_{h_m}\}$ in the child (body part m) is required to satisfy:

$$(x_{h_m} - x_{h_n})^2 + (y_{h_m} - y_{h_n})^2 < R. \tag{3.15}$$

Note that the final estimate of the torso is determined by choosing the candidate with the largest value of $L_m(a_m^t)$ in the *concentrated set*.

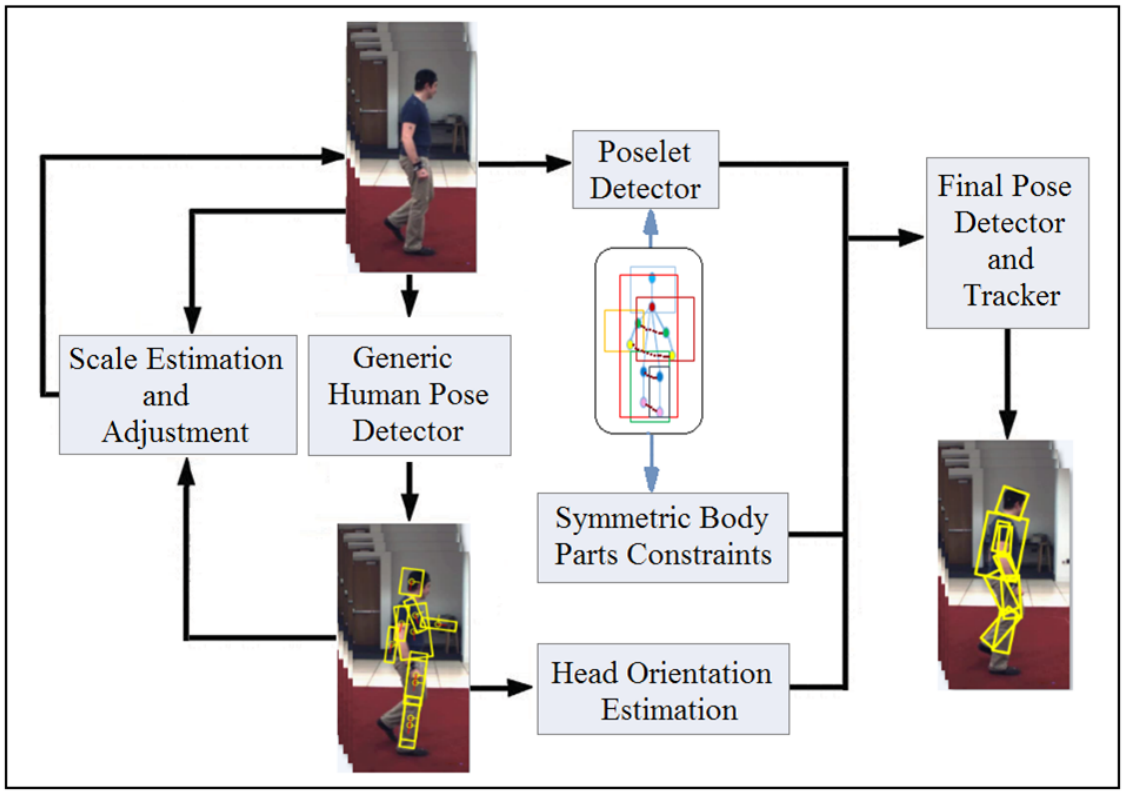


Figure 3.10: The overview of the proposed tracking system.

3.4 Overview of the Proposed Tracking System

The tracking system presented above is experimented and proved that it can track people in various motions from a single view and the performance is superior compared to several state-of-art 2D tracking systems (Lu *et al.* (2012b)). However, it is only for sequences in which the tracked target appears in the same or similar size. Moreover, tracking performance for the most active body part, such as the arms, is relatively low compared to the other parts of the body.

In order to address these issues, based on the framework described above (Lu *et al.* (2012b)), a new tracking system is proposed in this thesis and the overview of it is illustrated in Figure 3.10.

The main contributions of the whole system includes four parts: multi-scale strategy (MSS) for scale estimation and adjustment, poselet detector, additional constraints (AdCon) on symmetric body parts and head orientation estimation.

The first step of the proposed human pose tracking system is scale estimation and adjustment, which is implemented by the multi-scale strategy (MSS) described in Chapter 4. Two metrics for scale validation are proposed in this thesis depending on the motion types of the tracked target performing in sequences. One is the Height_Metric that is based on the height of the tracked target and suitable for sequences in which the tracked target is always upright. The other is PixelCount_Metric, which is implemented by computing the ratio between pixel counts of the foreground blobs and the detected body part bounding boxes and hence invariant to motion types. With the estimated scale value, each frame is then processed by the generic human pose detector based on the generic shape-based PicStr model and the mid-level poselet detectors separately. The poselet representations and detector implementation are presented in Chapter 5. The results obtained from the former are a set of estimates for every single body part. The latter provides a set of poselet detections for 'large parts' that cover a range of portions of the human body. The configurations of poselet detections can be combined to the generic PicStr model, which guide the search of smaller single parts by providing more image-conditioned information on dependencies of multiple body parts from a higher level while do not change the structure of the generic PicStr model. The head detection from the generic pose detector for each frame is processed to estimate the head orientation, and hence the orientation of the whole body. The goal of this component is to correct the confusion of the left and right limbs due to the overlapping and occlusion during tracking especially when the human is not facing front (detailed in Chapter 6). In addition to the inclusion of the mid-level poselet representations and the head orientation estimation, the final pose detector and tracker also considers more dependencies between symmetric body parts to constrain the left and right arms and legs, encoding the natural human distinction for balance and body coordination (see Chapter 6). With the final pose tracker, a series of human poses for all frames are obtained with high accuracy.

3.5 Chapter Summary

The goal of 2D human pose tracking is to track the articulations of people using 2D representations in video sequences. In this chapter we present a robust framework for human pose tracking with a specific accurate appearance model without motion

priors in 2D monocular images. One of the main contributions here is to propose an automatic process to identify the non-target pixels and excluding them from being used in building the appearance model. A specific appearance model has hence been proposed, which can be used to track the target's postures with much improved accuracy.

However, most of the existing systems including the one described in this chapter are for sequences in which the tracked target appears in the same or similar size. Moreover, tracking performance for the most active body part, such as the arms, is relatively low compared to the other parts of the body.

In order to address these issues, a new framework based on the pipeline of the system (Lu *et al.* (2012b)) is proposed in this thesis. Researches on tracking in various scales and improving the tracking performance for active limbs will become our main task in this thesis and will be discussed in more details in the subsequent chapters.

Chapter 4

Multi-Scale Strategy (MSS)

Many algorithms have been proposed for human motion tracking on 2D monocular videos. Most of 2D human postures tracking frameworks are inspired by the development of bottom-up pose estimation approaches and they tend to focus on building body models or deriving effective detectors. In most of these approaches, the tracked target in the video is moving with a rather fixed distance to the camera, resulting in the size of the human figure in the video to be constant or near constant, i.e., the perspective scale is fixed. In reality, videos often contain people appearing at any distance to the camera hence appeared in various scales in the videos. Often they are moving towards or away from the camera, resulting in their sizes (scales) to be changed within a video clip. In this chapter, we focus on the problem of tracking human motion in multiple scales in monocular image sequences.

A successful approach for 2D human pose tracking in video is to detect the human body and estimate body posture in each frame ('tracking by detection'). One example is by Ramanan *et al.* (2007), a colour-based specific appearance model based on the detections from a 'stylized pose' detector and then to track the person by detecting the model in each frame. Another system (Lu *et al.* (2012b)) proposes to combine a generic shape-based appearance model with a specific colour-based one for human motion tracking. Although the performance of Ramanan *et al.* (2007) and Lu *et al.* (2012b) is acceptable, a critical problem is that they do not implement the scale-variation issue. Lu *et al.* (2012b) state their tracking approach can only track a target at a single scale in a video clip. Ramanan *et al.* (2007) mention their system should work theoretically for the multiple scales by searching the pictorial structures over an image pyramid for each frame in the video. But no implementation detail is given in the paper on the idea. Since it is basically an exhaustive search it should be computationally inefficient.

Recently, there are a few approaches on pose estimation trying to address the scale-variation issue for still images. In Eichner *et al.* (2012), an upper-body detector and

a foreground high-lighting step are used to determine the approximate location and scale information of the person to be tracked. Although it is capable of estimating upper body pose in highly challenging images, the person to be tracked is required to be upright and seen from the front or the back (not the side). Andriluka *et al.* (2012) discuss the scale variation problem on pose estimation for still images. In their approach, the value for the scale parameter is changed within a fixed range at a fixed interval to estimate the human pose in the still images in a trial-and-error fashion. Even though this algorithm is quite generic, it is obviously not attractive for tracking in image sequences because it is cumbersome and computationally inefficient. In contrast, a more effective approach is proposed in this chapter for automatically evaluating and adjusting the perspective scales of the moving target during the tracking process, which enables the tracking for free-moving human motion with high efficiency and accuracy. Two strategies are proposed in this chapter depending on the motion that the target performs.

To track the human motion in a video sequence, it is required to detect the person's pose in each frame with a proper scale. A scale checking and adjusting step is incorporated into the tracking process. Two metrics are proposed for detecting and adjusting the scale change. One metric is from the height value of the tracked target (*Height_Metric*), which is suitable for some sequences where the tracked target has generally upright postures with no limbs stretching. For such sequences, a full body detector is proposed to estimate the height of the tracked target in each frame.

However, in general cases, the types of the motion performed by the tracked target are not known and the tracked target may not always maintain an upright posture. A metric is therefore needed to represent the scale changes that is invariant to motion types. An alternative metric is hence proposed that is more generic (named *PixelCount_Metric*). The illustration of our tracking system encoded with scale evaluation procedure based on PixelCount_Metric is shown in Figure 4.1. Specifically, the images are firstly processed with foreground segmentation which aims to obtain an approximate size of the body blob. This blob size is not used to determine the scale directly. Rather it is used to be compared with the size of the estimated human body (normally in the shape of bounding boxes) from pose estimation to determine whether the scale used for the pose estimation is appropriate. If the comparison shows that the scale value used satisfies the preset condition, the algorithm will proceed to the next frame using the same scale value. Otherwise, the scale value

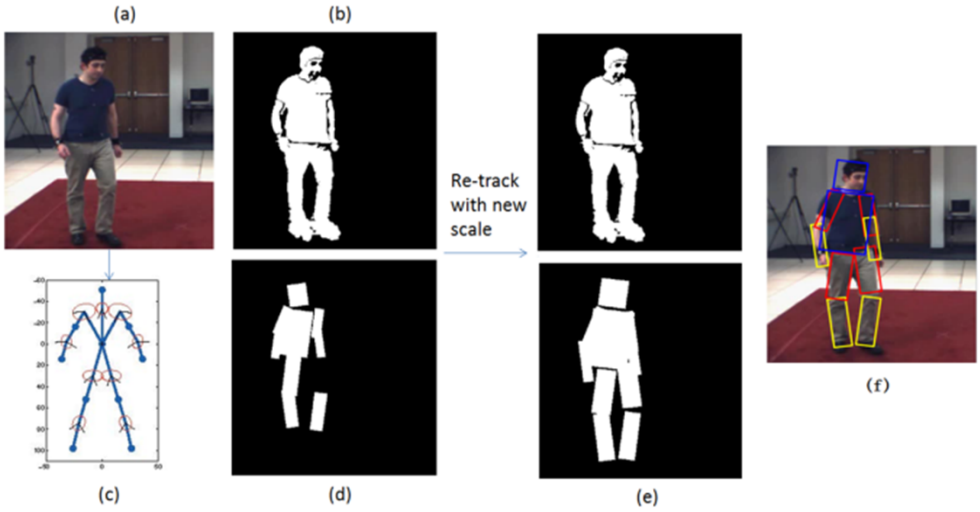


Figure 4.1: (a) original image. (b) foreground segmentation. (c) kinematic tree model. (d) area of the tracking body parts (represented using bounding boxes) for this frame. The pixel numbers from the foreground and from the tracked body parts are counted for scale evaluation. If the scale used is deemed inappropriate, the scale value will be changed and the frame reprocessed. When the scale value is satisfactory, the tracking results is accepted and shown as (f).

will be adjusted and the frame will be re-processed until the preset condition is met. The metrics and condition used for evaluating and adjusting the scale values are detailed in the next sections.

The structure of this chapter is as follows. Section 4.1 presents the Height_Metric for dealing with multi-scale issue during tracking of basically upright postures and the height of the tracked target is obtained from a full body detection. The Pixel-Count_Metric is described in Section 4.2, which provides the details of the method for obtaining the pixel numbers utilized on evaluating scale values. Finally, the performance of the proposed multi-scale algorithm with two metrics are evaluated separately in Section 4.3.

Figure 4.2: Some frames from the sequence HE_Walking_S1.

4.1 Metric from the Height of the Tracked Target

4.1.1 Full body detection

For certain human motions, such as the walking sequence shown in Figure 4.2, the change in the body height is a good representation for scale variation of the person. Before applying the temporal PicStr model for pose estimation and tracking, a pre-processing stage is incorporated to find the height of the full body in each frame, hence provide indication for scale evaluation and adjusting.

A full body detector is implemented with the deformable part-based model (DPM) framework. The model of the full body detector is defined by a coarse 'root' filter similar to the Dalal-Triggs filter on histogram of oriented gradients (HOG) features Dalal and Triggs (2005) which approximately covers the full body, and a series of higher resolution part filters that cover smaller parts of the human body. In implementation, the part filters capture features at twice the spatial resolution to the features of the root filter. The part filters are collected by a graphical model

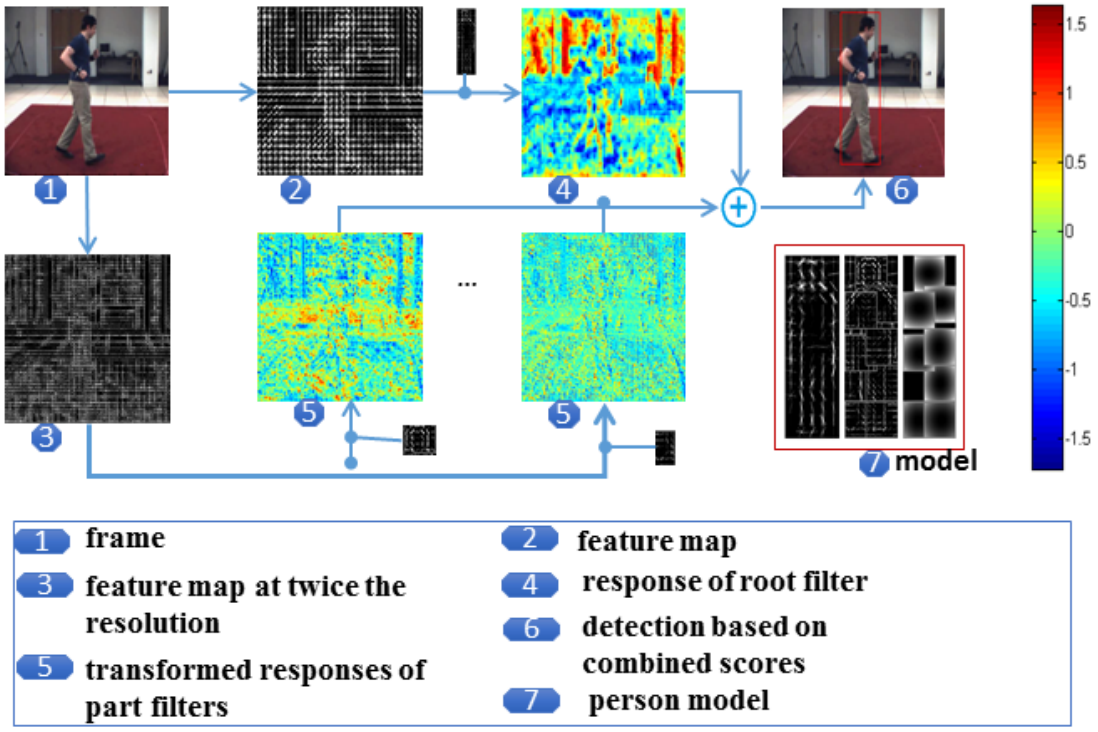


Figure 4.3: Full body detector. We only show the transformed responses for the head and left-lower leg. The model shown in (7) includes three components: a coarse root filter, several higher resolution part filters, and a spatial model for the location of each part relative to the root.

with deformation prior (Figure 4.3 (7)).

An hypothesis of the detection specifies the location of each filter in the model, $z = (p_0, ..., p_m)$, where p_i is the position for the i_{th} filter. At a particular position of an image, the score of a hypothesis is computed by the response of the root filter plus the sum of the transformed responses of each part filter (Figure 4.3 (1)-(6)). Note that the transformed responses are obtained from the responses of part filters minus a deformation cost that depends on the relative position of each part with respect to the root (the spatial prior). The score of a hypothesis z can be expressed in terms of a dot product between a vector of model parameters β and a feature vector $\psi(H, z)$ as:

$$score(z) = score(p_0, ..., p_m) = \beta \cdot \psi(H, z). \tag{4.1}$$

Here, β is obtained by concatenating the root filter, the part filters, and the deformation cost weights; H is a feature pyramid; and $\psi(H, z)$ is a concatenation of

subwindows from the feature pyramid and part deformation features. Detecting a person in an image means to find a root location with high score and the corresponding part locations with optimal displacements:

$$score(p_0) = \max_{p_1,...,p_m} score(p_0, ..., p_m). \tag{4.2}$$

The detection result in an image is defined by a bounding box (bbox) $B = (x_1, y_1, x_2, y_2)$ with the upper-left and lower-right corners being at (x_1, y_1) and (x_2, y_2) respectively. Then $h_i = (y_2 - y_1)$ is taken as the height value for the i_{th} frame.

4.1.2 ROI normalization

A straightforward metric as given in Equation 4.3 is proposed for estimating the scale value in motions where the tracked target keeps upright with no extreme limbs stretching,

$$s_i = h_i/\alpha, \tag{4.3}$$

where s_i is the scale value used for pose estimation for the i_{th} frame, h_i is the body height in pixels measured from the tracked target in the frame, and α is a reference coefficient, which corresponds to the height of the tracked person in pixels when the scale equals to 1. It is important to note that the scale here is defined to be relative to the value in the training set.

After the scale values for each frame are obtained, they are not directly used to estimate and track poses. Rather all these scale values are processed as indicators to normalize the height of target person in each frame. From the full body detector, a bounding box (bbox) containing all pixels representing the person is achieved. In order to avoid the possible impact of imperfect bounding box boundaries or the false positive detections, the bounding boxes are enlarged by 10 pixels vertically and 15 pixels horizontally in the original images. To ensure the tracking to be invariant to the size of the human body appeared in different images, the bounding box area is cropped out and resized to a patch with a normalized height h that is derived from the scale-normalized training set for PicStr. In this work $h = 210$. The normalized bounding boxes form the final ROIs.

Figure 4.4: Sample poses which are not upright, such as seating, bending or the ones with limbs stretching.

4.2 Metric from Pixel Counts

Although the body height is a straightforward metric for sequences such as walking, it is not suitable for sequences where the human body is not always upright such as the poses shown in Figure 4.4.

A simple fact of perspective projection is that the number of pixels a person projected onto an image always changes with respect to the distance between the person and the camera, regardless of the pose/motion of the person. Therefore, the number of pixels occupied by the bounding boxes representing the estimated human is a good indication of the scale value used for pose estimation. The pixel count hence provides a very good means for estimating and adjusting the scale value. As shown in Figure 4.5, if the two pixel counts are similar (as in row 1), it can be concluded that the scale used for the pose estimation is acceptable. Otherwise as shown in the second row, the scale value used for pose estimation is far from adequate and needs to be adjusted according to the difference between the two pixel counts.

In many situations, the tracked person could stretch his limbs or bend towards/away from the camera, resulting in some parts of the body with more scale changes than the others. Like most state-of-art motion tracking techniques, we do not distinguish the scale differences within body parts since their effect is rather insignificant under the current bounding box framework.

The pixels occupied by the tracked target in images can be obtained through image segmentation, while the pixels occupied by the bounding boxes can be easily identified after pose estimation. Assuming the pixel numbers counted from both operations are denoted as n_1 and n_2 respectively, the scale value used for pose esti-

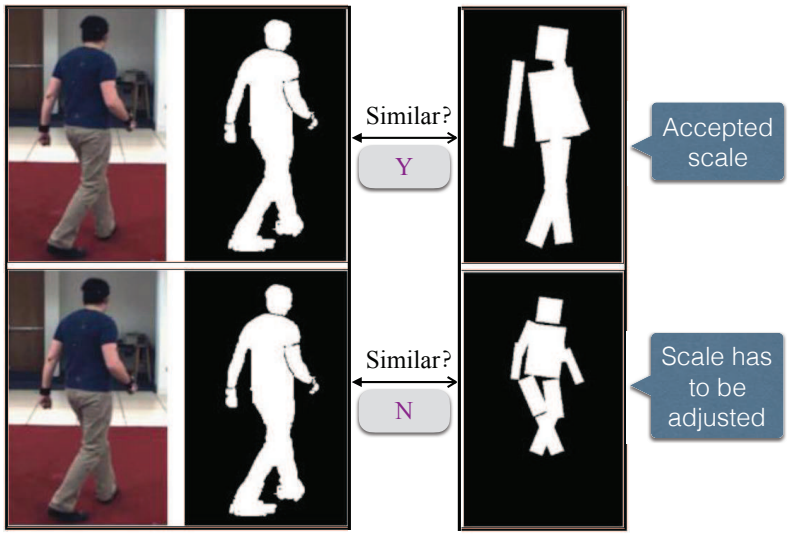


Figure 4.5: Examples of pixel area of the foreground blobs and the estimated bounding boxes for the body parts for a single image. In both rows, the images in the second column are the foreground blobs and the images in the third column show the bounding boxes(bbox) area. It can be seen that the area of foreground blob is larger than the bbox in Row 2 while the two areas in Row 1 are similar.

mation is evaluated and adjusted according to Equation 4.4

$$s_i = \begin{cases} s_i & \text{if } |r_i - 1| \leqslant \sigma \\ \sqrt{r_i} * s_i & \text{else} \end{cases} \tag{4.4}$$

where s_i is the scale value used for pose estimation for the i_{th} frame, r_i stands for the ratio between n_1 and n_2, and σ is the threshold, again set $\sigma = 0.1$. Since any changes in the scale value will apply to both the width and the height in a 2D image, square root of the ratio r_i is chosen as the coefficient in Equation 4.4.

An initial value needs to be given for s_i for the first frame. It does not have to be a proper scale, since the pixel numbers n_1 and n_2 counted after image segmentation and pose estimation will be compared to check whether s_1 is an appropriate value. It can then be adjusted according to Equation 4.4 and used for pose estimation until $r_1 \approx 1$. The updated scale will be used for tracking the second frame, and the same procedure will apply to all remaining frames.

Our framework focuses on situations using a fixed camera, so background subtraction is a proper method for image segmentation. In our implementation, an extend

version of background subtraction (Stauffer and Grimson (1999); KaewTraKulPong and Bowden (2002)) is selected to provide the blobs of the foreground. Although in general, image segmentation is unable to provide accurate image blobs for representing the tracked target, it is sufficient to provide the approximate pixel count of the human body projection, and can be easily implemented in our approach.

The pixel number n_2 can be easily obtained by considering the vertices of the resulting bounding box for each body part. The pixels bounded by them can be easily counted with overlapping areas counted only once.

4.3 Experiments and Discussion

In this section we evaluate the performance of the proposed multi-scale algorithm. **PCP Metric:** To numerically evaluate the performance, we use the well-known PCP metric proposed in Ferrari *et al.* (2008). Specifically, a body part l_m is considered correctly localized if the endpoints of its segment lie within 50% of the ground-truth segment length. This is commonly used as an evaluation metric in bottom-up human pose tracking.

4.3.1 Evaluation of the proposed metrics

We begin the experimental part of this chapter with the evaluation of each proposed metric.

Tracking using the Height_Metric (M_I).

The proposed multi-scale tracking framework is first applied on two sequences from the well-known HumanEva dataset (Sigal *et al.* (2010)), hereby named as HE_Walking_S1 and HE_Jogging_S1. In both sequences, the tracked person walks or jogs in a circle, thereby generates image frames with different scales and shows different body orientations including frontal, back, and sideways, but is basically upright in all frames. Figure 4.2 and Figure 4.6 shows some sample frames of both sequences. We also apply the Height_Metric on a walking sequence (named Walking_S2) that we collect

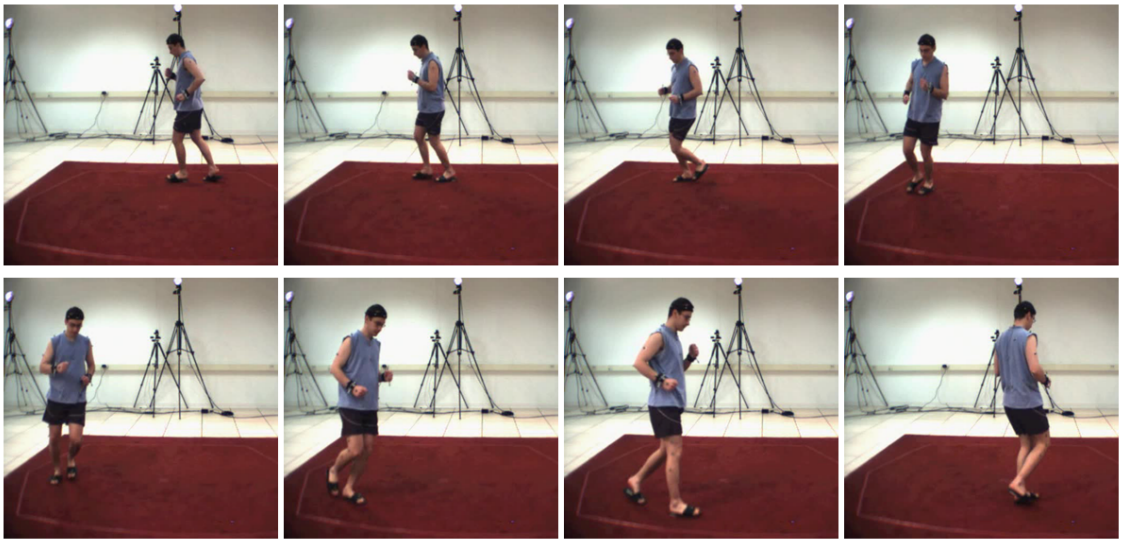


Figure 4.6: Some frames from the sequence HE_Jogging_S1.

Figure 4.7: Some frames from the sequence Walking_S2.

Figure 4.8: Some normalized ROI results processed from the full body detector for jogging and walking sequences.

from a free environment and some sample frames are shown in Figure 4.7. The tracking targets recorded in these sequences are all shown in different sizes because their distances to the fixed camera vary throughout the sequences.

These sequences are firstly processed by the full body detector that is described in section 4.1.1 and some sample normalized ROI results are illustrated in Figure 4.8. The height of them are normalized to the reference value, which also greatly decreases the searching expense by removing most of the background area.

Table 4.1: Tracking results based on the Height_Metric in percentage.

Sequence	Tor	Head	U.L.	L.L.	U.A.	F.A.	Total
HE_Walking_S1	99.2	95.5	85.1 86.5	81.6 79.3	86.4 84.5	82.3 84.1	86.5
HE_Jogging_S1	96.9	93.8	93.8 87.5	90.6 78.1	59.4 59.4	31.3 25.0	71.6

Figure 4.9: Sample results on jogging and walking sequences from the proposed tracking system.

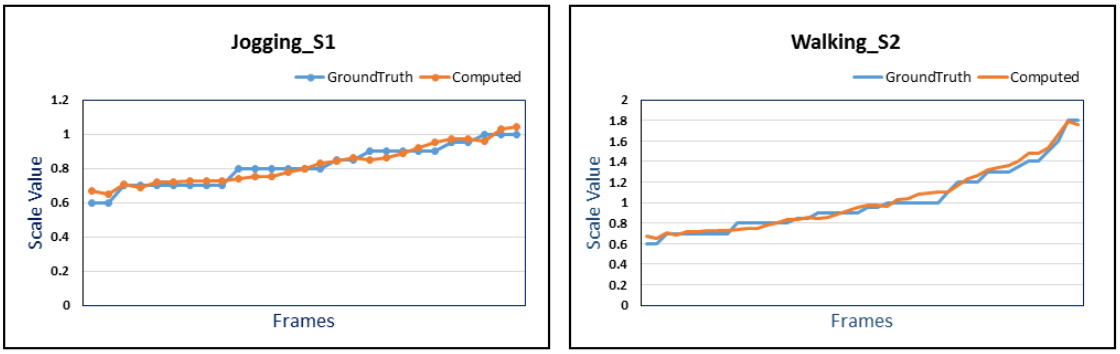


Figure 4.10: Comparison of scale values between the ground_truth and the ones achieved from the Height_Metric.

Figure 4.9 and Table 4.1 show the quantitative and qualitative tracking results on jogging and walking sequences from the proposed tracking system. It demonstrates that, for upright poses, the height is a suitable metric for scale evaluating and the normalization depending on the height of the tracked target is a good approach to deal with scale-variation issue during tracking.

To further evaluate the performance of the Height_Metric, we try to obtain the 'ground_truth' of the scale values for every frame in sequences HE_Jogging_S1 and Walking_S2 by doing pose estimation for each frame with the trial-and-error idea proposed by Andriluka *et al.* (2012). The scale parameter is changed at a fixed interval within a range and the best scale value is selected as the 'true' scale value depending on the estimation score for each frame. The comparison of the 'true' scale values obtained this way and derived from the Height_Metric for both sequences are shown in Figure 4.10. The computed scale values from the Height_Metric match well with the 'true' scale values.

Tracking using the PixelCount_Metric (M_II).

The proposed multi-scale tracking framework with PixelCount_Metric is firstly applied on the sequence Skating_S1, which is extracted from a video recording the skater Michelle Kwan performed at Olympics Sports 1998 and consisting of 116 frames. The skater in the sequence Skating_S1 performs different actions with different scales and various poses. The second row of Figure 4.11 shows several sample frames of this sequence.

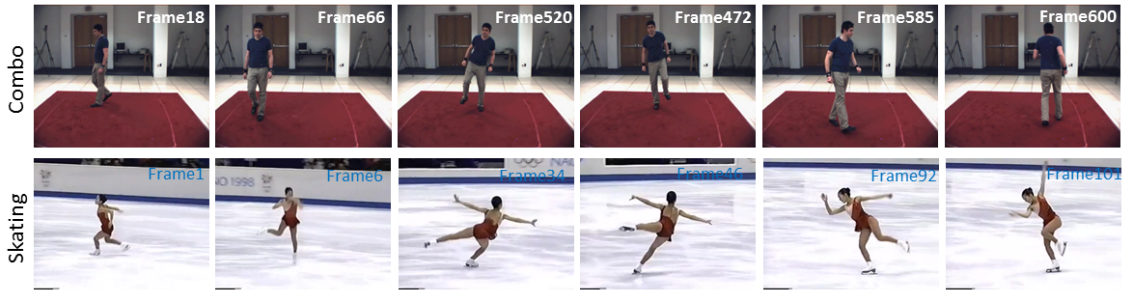


Figure 4.11: Some frames from sequences Skating_S1 and HE_Combo_S1.

Figure 4.12: Sample results on the sequence Skating_S1.

Many poses are not upright but bending or stretching, thus changes on the height of the tracking target do not reflect the corresponding scale variation. Rather the ratio between the foreground blobs and the estimated bounding-boxes better reflects whether the utilized scale is acceptable or not. Therefore, the PixelCount_Metric can deal with the scale variation issue regardless of the type of poses and motions.

Several screenshots of the tracking results on the sequence Skating_S1 based on the PixelCount_Metric are illustrated in Figure 4.12 and the performance is quantitatively illustrated in the first row of Table 4.2. The results prove that the system can handle scale variation issue during tracking.

Table 4.2: Tracking results based on the PixelCount_Metric in percentage.

Sequence	Tor	Head	U.L.	L.L.	U.A.	F.A.	Total
Skating_S1	94	91.4	78.5 81.0	77.6 78.5	76.7 71.6	61.2 59.5	77.0
HE_Combo_S1	98.4	94.5	84.7 83.2	81.8 78.0	81.0 82.5	73.1 72.8	83.0

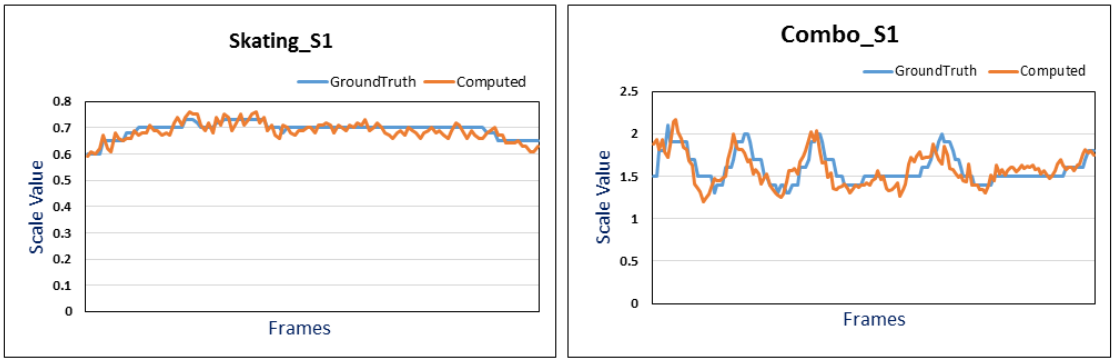


Figure 4.13: Comparison of scale values between the ground_truth and the ones achieved from the PixelCount_Metric.

Besides special poses like skating, the PixelCount_Metric is also definitely suitable for the sequences including upright poses. In order to analyze the performance on sequences with upright poses, the system with the PixelCount_Metric is applied on the sequence HE_Combo_S1, which is a 600 frame sequence containing a person performing different motions in a circle and showing different scales and body orientations. In addition, the sequence contains several non-lateral motions, such as jumping, kicking, leaning and stretching. The numerical analysis is presented in the second row of Table 4.2.

Same as the analysis in the previous part, for further evaluating the performance of the PixelCount_Metric, we also compare the estimated scale values with the 'ground_truth' for sequences Skating_S1 and HE_Combo_S1. Similarly, the 'ground_truth' is obtained by performing pose estimation for each frame in a trial-and-error fashion proposed by Andriluka *et al.* (2012). The comparison between the computed scale values from the PixelCount_Metric and the 'ground_truth' for both sequences are shown in Figure 4.13. It is clearly seen that the computed scale values from the PixelCount_Metric match well with the 'ground_truth'.

4.3.2 Combing the multi-scale strategy (MSS) to the Classifier and Cluster algorithms

For further testing the application of the proposed multi-scale strategy (MSS), we incorporate MSS to both the Classifier algorithm (described in Chapter 3) and the

Figure 4.14: The screenshots of tracking results for HE_Combo_S1 sequence from Cluster (CLS) and Classifier(CLF) without and with MSS.

Cluster algorithm (Lu *et al.* (2012b)) separately and compare their performance qualitatively and quantitatively with the tracking results from the same algorithms without the MSS. For this experiment, we still utilize the sequence HE_Combo_S1, which combines different motion together.

The performance of the tracking results from these algorithms (Classifier and Cluster with and without the MSS are shown in Figure 4.14 and Table 4.3.

In Figure 4.14 Row 1 and Row 3 are the results by Classifier and Cluster with a fixed scale value, while the other two rows show the results of the two algorithms with MSS incorporated. The results of tracking without MSS are clearly much worse because these algorithms cannot handle scale variations. When the scale value used is not appropriate for a frame, some or even all body parts cannot be correctly located. However, with the proposed multi-scale strategy incorporated, both algorithms can produce satisfactory tracking performance for the sequence with a wide range of motion types. It can be seen from Table 4.3 that the performance of the systems with MSS actually improves for every body part.

Table 4.3: The performance of CLUSTER and CLASSIFIER systems with and without multi-scale strategy (MSS)

Dataset	Method	Tor	Head	U.L.	L.L.	U.A.	F.A.	Total
HE_Combo_S1	CLUSTER	86.8	81.6	**56.8 56.8**	**53.7 52.5**	**44.3 48.7**	**41.8 40.2**	56.3
	CLUSTER+MSS	100	98.0	**83.8 81.8**	**82.6 77.9**	**83.7 78.4**	**78.7 75.9**	84.1
	CLASSIFIER	85.3	82.7	**57.4 55.8**	**55.6 53.5**	**48.3 44.2**	**43.4 41.2**	56.7
	CLASSIFIER+MSS	100	98.4	**87.1 86.8**	**85.0 81.3**	**86.3 70.7**	**85.5 70.0**	**85.1**

Table 4.4: Comparison of tracking results on Combo sequence with scale variation.

Method	Tor	Head	U.L.	L.L.	U.A.	F.A.	Total
Ramanan	52.5	36.3	45.8 43.4	55.8 52.0	24.3 27.5	32.8 29.7	40.0
Yao	85.3	82.7	57.4 55.8	55.6 53.5	48.3 44.2	43.4 41.2	56.7
Proposed	98.4	94.5	84.7 83.2	81.8 78.0	81.0 82.5	73.1 72.8	83.0

4.3.3 Comparison to state-of-the-art approaches

Comparison with Lu *et al.* (2012a) (Yao) and Ramanan *et al.* (2007) (Ramanan).

To further illustrate the performance of the proposed algorithm, we compare it with Lu *et al.* (2012a) (Yao) and Ramanan *et al.* (2007) (Ramanan) on sequences where the tracked person appears with scale variation. The reason for choosing these two systems as the benchmark since the basic tracking ideas in them are similar to ours, such as pictorial structures model, baseline of tracking by detection, etc.. We implement the two approaches based on their provided source code.

The three methods are applied on the sequence HE_Combo_S1, in which the tracked person appears at different scales, and the tracking results are compared. Sample screenshots of the tracking results from these three frameworks are shown in Figure 4.15. The first row is the results by Ramanan *et al.* (2007) (Ramanan) and the second row shows the results by Lu *et al.* (2012a) (Yao). The results from our approach are shown in row 3. It is obvious that our method can produce satisfactory tracking performance for sequences not only with a wide range of motion types but also with the significant scale variations. The other two approaches fail for frames in which the assumed fixed scale does not provide a reasonable approximation.

The quantitative comparison is given in Table 4.4, where the accuracy is evaluated

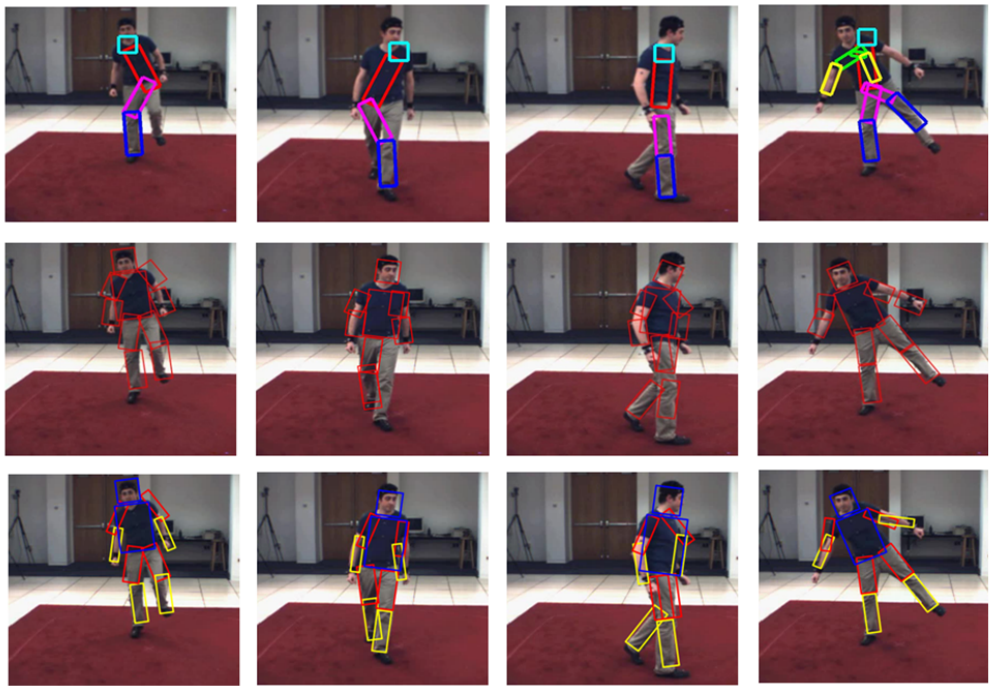


Figure 4.15: Screenshots of tracking results on sequences with scale variations.

based on the tracking results for all frames in the sequence. Clearly, the tracking performance of our approach surpasses the other two with the tracking for all body parts remarkably improved. It appears that the method proposed by Ramanan *et al.* (2007) (Ramanan) performs quite poorly when there are significant scale variations in the image sequence. This clearly demonstrates the importance of including scale adjustment in the tracking process, since the overall performance can be greatly improved.

4.4 Summary

In this chapter, we attempt to address the scale variation problem in a human motion tracking framework for 2D monocular images. An automatic multi-scale strategy (MSS) is proposed to adaptively change the scale values during the tracking process. Two metrics are proposed to be used in the MSS. One is the Height_Metric, which is a simple and straightforward metric suitable for motions where the tracked target remains basically upright. The other is the PixelCount_Metric, which is implemented by computing the ratio between pixel counts of the foreground blobs

and the detected body part bounding boxes. This metric is more complicated yet is more generic and invariant to motion types. The efficacy of the proposed MSS is demonstrated through experiments on the publicly available HumanEva dataset and videos taken from uncontrolled environment, where the proposed algorithm can produce significantly improved tracking results.

Chapter 5

Modelling Non-connected Body Part Dependencies by Poselets

In the generic Pictorial Structures (PicStr) model, part detectors are trained invariant to poses. In order to compute the likelihood and inference efficiently, the body parts appearances are often assumed to be mutually independent (Andriluka *et al.* (2012)) and the prior over body parts connections are assumed to be a Gaussian with a tree structure independent to image evidence.

However, there generally exist a great number of strong dependencies between/among many, even all body parts in human activities, such as walking, dancing or playing ball games. In other words, human motions and activities often make the positions of multiple body parts correlated (see Figure 5.1). Such property has not been reflected within the generic PicStr approach, which limits the accuracy of pose estimation and tracking.

Furthermore, there is another problem often occurring within the generic PicStr model: double counting, i.e., both body parts are arranged at the same location in

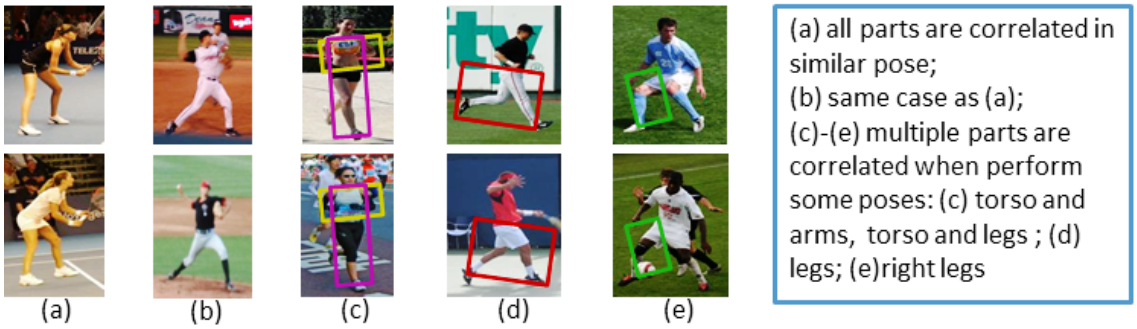


Figure 5.1: Illustrations of the correlation among appearance of multiple body parts. Given similar poses such as in (a)-(e), multiple even all body parts are dependent even if they are not directly connected.

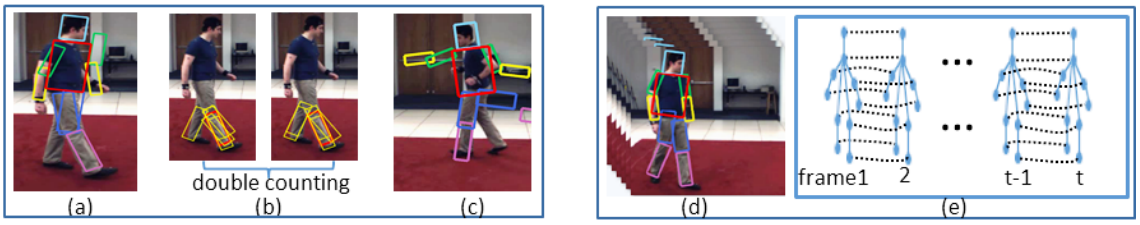


Figure 5.2: The possible erroneous cases with the generic PicStr (a)-(c) and the tracking process with temporal PicStr model (d)-(e).

the image. Double counting often happens during pose estimation when symmetric part pairs share the high detection score at the same image evidence (see Figure 5.2 (a) and (b)). We have found out that modelling the dependencies between non-adjacent body parts is highly effective in handling such detection errors during the tracking process.

In this chapter, we propose to incorporate some more expressive spatial constraints by defining a mixture of mid-level image conditioned spatial representations that model higher order information between body parts (named poselets, which is introduced by Bourdev *et al.* (2010)). We employ them to capture multiple body parts configurations and dependencies.

The generic tree-based PicStr methods used for pose estimation and tracking generally work well for images/sequences where all limbs of the person are visible. However, they are easily affected by cluttered background and suffer from double counting issue especially when self-occlusions occur. Several recent approaches augment the tree-structure to capture cues such as appearance similarity between limbs not connected in the tree (Karlinsky and Ullman (2012); Tran and Forsyth (2010)). Wang *et al.* (2011) propose an approach relying on a complex hierarchical model that requires approximating inference with loopy belief propagation. A recent work proposes an occlusion-aware algorithm for tracking human pose in image sequences in order to address the problem of double counting (Ramakrishna *et al.* (2013)). They address the problem of tracking human pose using an iteration process and employing multi-target tracking algorithm for the symmetric parts.

To effectively model dependencies between non-adjacent body parts while still allowing for efficient and exact inference in a tractable PicStr model, the mid-level features (poselets) are chosen and incorporated into a conditional model in which

all parts are connected *a priori*. The poselet representation increases the flexibility of the PicStr approach by collecting a set of image specific part appearance and dependencies. Similar to Desai and Ramanan (2012), our approach can incorporate dependencies between body parts that go beyond adjacent pairwise interactions while allow efficient inference at test time.

The structure of this chapter is as follows. Section 5.1 describes the details of the poselet representation including poselet descriptions, poselet detectors training and testing. The unary and pairwise terms of poselets conditioned PicStr model are detailed in Section 5.2. The final tracking system is described in Section 5.3. Section 5.4 describes the experiments conducted to analyze the performance of the poselets conditioned PicStr model presented in this chapter and compare it against other state-of-the-art approaches. Finally, some concluding marks and discussions are given in Section 5.5.

5.1 Poselet Representation

Poselets are pieces of human poses that are tightly similar in both appearance and configuration spaces, which aim to capture common dependencies of multiple body parts.

In the standard PicStr model, the human body is modelled with a set of rigid parts corresponding to body segmentation, e.g. torso, head, upper/lower arms/legs. This definition of parts is natural, but it excludes interactions of non-connected body parts and is easily confused by similar shape existing in the background.

In reality, some action patterns are visually distinctive, such as 'a torso with left arm raising up', 'right leg kicking' or 'legs in lateral pose'. The group of these kind of 'parts' convey more motion information and provide more useful constraints on part dependencies. Therefore, the 'parts' can be re-defined to cover more pieces of human poses at various levels.

Apart from the original 10 rigid parts defined in the generic PicStr model, another 11 body part configurations are defined in our system to serve as the mid-level poselets,

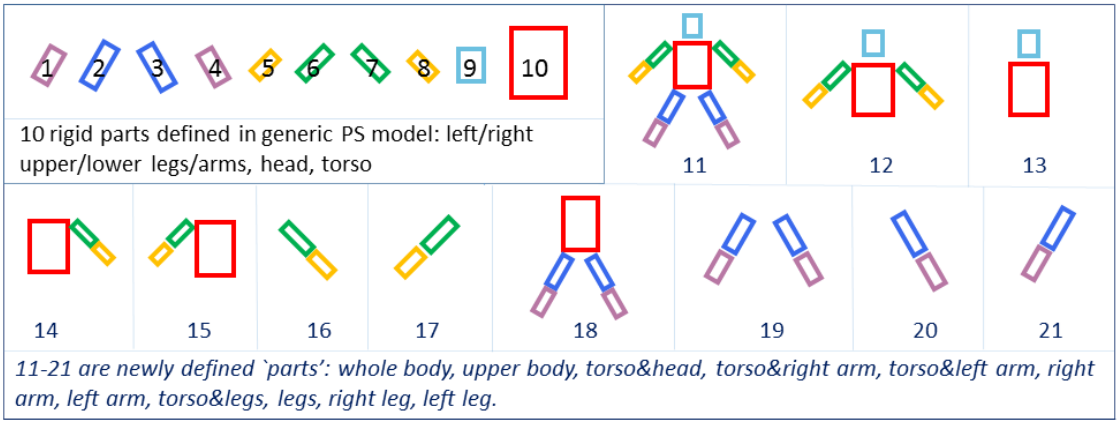


Figure 5.3: An illustration of all parts utilized in our tracking system. $1-10$ are original rigid body parts defined in the generic PicStr model. $11-21$ are newly defined parts for presenting poselets information.

similar to Pishchulin *et al.* (2013), namely full body, upper body, torso and head, right arm and torso, left arm and torso, right arm alone, left arm alone, torso with legs, legs, right leg alone, and left leg alone (shown in Figure 5.3).

The following two steps are implemented to train a set of poselet detectors. The first step is to prepare the training data, i.e., to select proper poselets clusters for each poselet-part. The second step is to learn detectors for all poselets.

Preparing the poselet clusters.

Figure 5.5 (a) shows all joint positions $(0-15)$ of the rigid body segmentations. These joints are also utilized as reference positions for poselets. Each poselet-part is assigned with a joint as its reference position. For example, the reference position of the *full body* part is joint 8 in Figure 5.5 (a). Similarly, joint 12 is the reference position for *left arm alone*, and joint 6 is for *legs* part, etc..

Firstly, the joints on each poselet-part are clustered into several clusters based on their offsets (relative x and y coordinates) of all related rigid body parts with respect to the reference position using Euclidean distance. For example, for the part 'upper body', joint 8 is chosen as the reference point and the relative coordinates of all upper body parts are computed with respect to this reference joint, which is concatenated to form a vector used for clustering.

Figure 5.4: Examples of three poselets for the 'legs' part and each row corresponds to a poselet.

K-means is selected for clustering on the vectors collected from all training images. The clusters that have less than 10 members are removed and the remaining ones are utilized as poselets. In such method, we obtain a set of K ($K \leqslant 200$) clusters and each cluster contains similar pose type. These clusters are used as poselets. Similarly, the clusters for all other poselet-parts are obtained by picking different reference points and different subsets of related rigid body parts and finally a total of M clusters are obtained.

Based on the clustering, the corresponding patches from the images can be cropped (examples are shown in Figure 5.4). Note that, in this work, the annotated training images are from the 'Leeds Sports Poses'(LSP) dataset Johnson and Everingham (2010) that includes 2000 images showing people involved in various sports (see Figure 5.1).

Learning the poselet detectors.

With the clusters collected, the next step is to learn detectors for all poselets, i.e., a separate AdaBoost detector that is trained based on the dense shape context features for every poselet cluster.

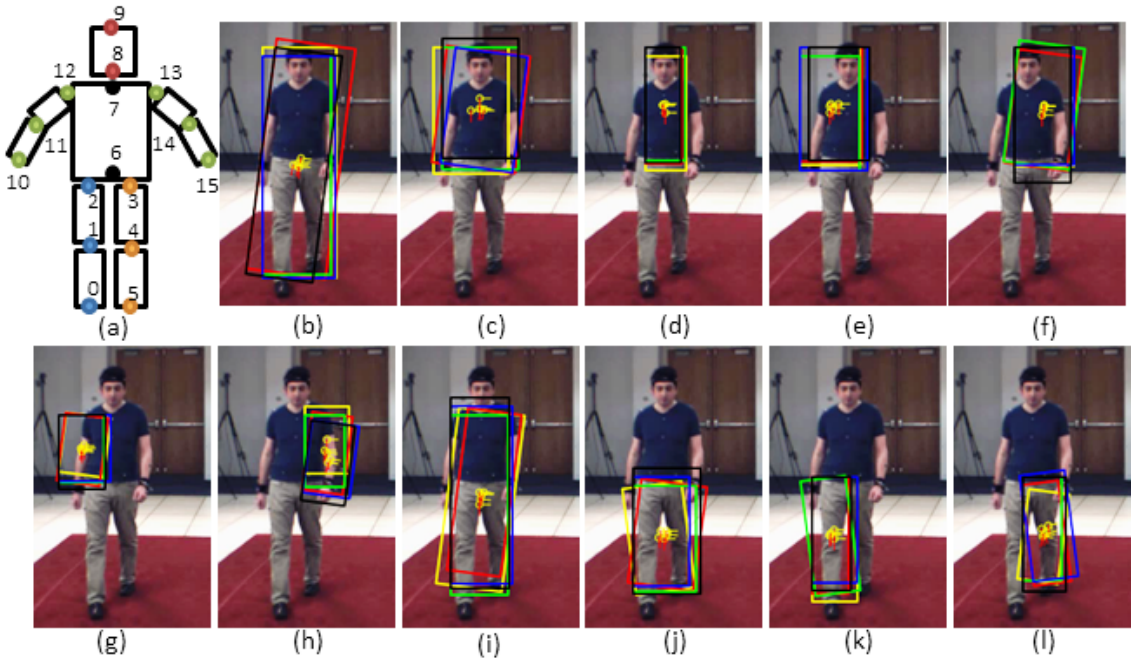


Figure 5.5: (a) shows hinge points of single body parts. In our model, each part is controlled by two hinge points, hence the ground truth of all body parts is defined by 16 hinge points. (b)-(l) illustrate the poselet detection outcomes for one sample frame. Note only results for some not all poselet configurations are shown here.

For each poselet, shape context features are constructed from patches in the corresponding cluster. Then a set of Adaboost detectors are trained for poselets using all training images.

Given a test frame, we can compute the maximum response for every poselet detector which represents the modalities of human poses and use them as mid-level feature representations. Example poselet detection results for one sample frame are shown in Figure 5.5, in which the top scoring poselets are sampled and denoted with bounding boxes.

The parts considered in this chapter are 'large parts' that cover a wide range of portions of human bodies. The reason for choosing this kind of representation is mainly due to the fact that large body parts are easily found at the coarse level and the configurations of large pieces of human bodies can guide the search of smaller parts. For example, an upright torso with arms raising up is coarse-level information, which is a very good indicator of where the arms (fine-level details) might be.

Therefore, poselets are chosen to capture distinctive appearance patterns of various parts. These poselets have better discriminative powers than the traditional rigid part detectors.

5.2 Poselet-conditioned Pictorial Structures Model

As discussed in Chapter 3 and Chapter 4, in each frame, the body parts are represented with a PicStr model and the generic PicStr model is formulated by

$$E(L; D) = \sum_{i=1}^{N} E^u(l_i; D) + \sum_{i \sim j} E^p(l_i, l_j). \tag{5.1}$$

The unary terms E^u represent the image likelihood, and the pairwise relationships between body parts are denoted by $i \sim j$. They are spatial priors encoding the kinematic dependencies of body parts. In order to simplify the inference, an assumption is introduced in Felzenszwalb and Huttenlocher (2005) that the part d_i only depends on its own configuration l_i and different part evidences are conditionally independent given the configuration L. Furthermore, the pairwise term is *a prior* tree-based structure, which is image independent and modelling the displacement relations between *adjacent* body parts only.

To generate optimal proposals for part locations in each frame, we need to take a closer look on the terms E^u and E^p in Equation (5.1). In this section we focus on improving the model representation to make it more flexible and also capable for encoding *non-adjacent* part dependencies. Note that we do not intend to change the base PicStr model structure, but aim to include a mid-level stage to reflect the dependencies of multiple body parts, inspired by the image-conditioned poselets idea introduced in the work Pishchulin *et al.* (2013).

The location and rotation for each body part are first predicted separately using the poselet features. For instance, to predict the position of part i, during training we cluster the relative offsets between the reference point and the part into K clusters. For each cluster we compute the mean offset μ and the variance Σ. If the mid-level poselet feature f is obtained, prediction will be treated as a multi-class classification

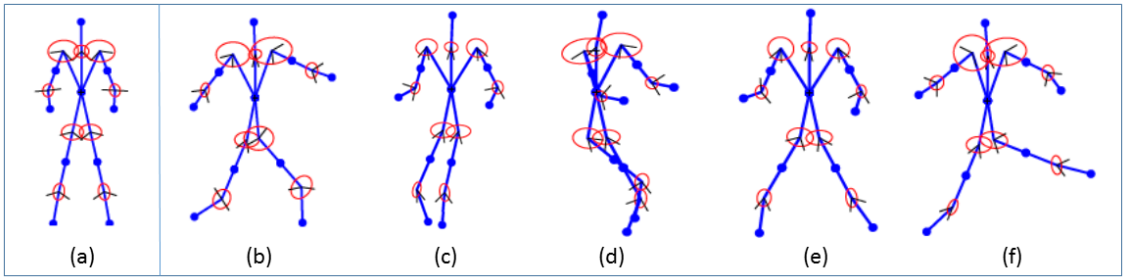


Figure 5.6: (a) shows the generic tree-based PicStr model. (b)-(f) illustrate the samples of the poselets conditioned PicStr model with deformable pairwise terms. (reproduced from Pishchulin *et al.* (2013))

problem, from the poselet response f into the set of K clusters. A classifier can now be trained based on the sparse linear discriminant analysis (sLDA) Clemmensen *et al.* (2011). The mean offset μ and the variance Σ can be predicted from the poselet feature f using the learned classifier. Here the mean offset μ and the variance Σ are subsequently used as a Gaussian unary potential for the part. Prediction of the absolute body part orientation is done in the same way. A sLDA classifier is learned to predict the absolute part rotation based on the poselet responses. The Gaussian unary terms for both the location and the orientation of a part also form a Gaussian potential denoted by $E^{u,m}$. The complete unary term of our model is then defined by

$$E^{u}(l_i; D) = E^{u,s}(l_i; D) + \omega \cdot E^{u,m}(l_i; D), \tag{5.2}$$

where $E^{u,s}$ is the original unary term for a single body part given in the generic PicStr model as denoted in Equation (5.1); ω is the weighting parameter of the poselets-based unary term estimated on the validation set, which defines the influence extent of the poselet features in the final solution. Following Pishchulin *et al.* (2013), ω is set as 0.05.

The generic PicStr model has a limitation that the spatial prior of the body parts is modelled as a tree-structured Gaussian and independent to image evidence, which cannot properly represent the multi-modalities of human poses. Here we extend the pairwise terms in Equation (5.1) with image conditioned factors and then make them image dependent and more flexible (see Figure 5.6).

For each pair of parts (l_i, l_j), the training data with respect to the relative part rotations is clustered into K clusters. Similar to the unary terms, a sLDA classifier

is trained to predict the type of the multiple pairwise terms based on the image conditioned poselet feature f. The new pairwise terms is defined by

$$E^p(l_i, l_j) = E^{p,m}(l_i, l_j; D). \tag{5.3}$$

The original image-independent pairwise term defined in Equation (5.1) is then replaced by image-conditioned term $E^{p,m}(l_i, l_j; D)$ that is embedded with multimodal of body poses.

With the improved image conditioned unary and pairwise terms, predictions for single parts in each frame can be obtained.

5.3 Tracking

In addition to the single-frame inference (see the left part of Figure 5.7), we extend the model to include dependencies between body parts over time (see the right part of Figure 5.7), which becomes a temporal PicStr model and is defined by

$$p(L_{1:T}^{1:N} \mid D_{1:T}^{1:N}) \propto \sum_{t=1}^{T} \sum_{i=1}^{N} \left(\phi(l_t^i, l_{t-1}^i) + E^u(l_t^i; D) + E^p(l_t^i, l_t^j; D) \right). \tag{5.4}$$

Here superscripts are used to represent body parts and subscripts for frames, *e.g.*, l_t^i is the configuration of part i at time t.

The first term in the right-hand side of Equation (5.4) is the motion model capturing temporal continuity of body parts between frames. Since the position of body parts generally changes smoothly within an image sequence, we exploit this temporal continuity using a simple velocity threshold, i.e., the Euclidean distance between the part positions in successive frames should not exceed a preset threshold:

$$\phi(l_t^i, l_{t-1}^i) \propto \mathcal{I}(\mathcal{D}(l_t^i, l_{t-1}^i) < d_{max}), \tag{5.5}$$

where $\mathcal{I}$ is the standard identity function and $\mathcal{D}(l_t^i, l_{t-1}^i)$ is the Euclidean distance between the part positions in consecutive frames.

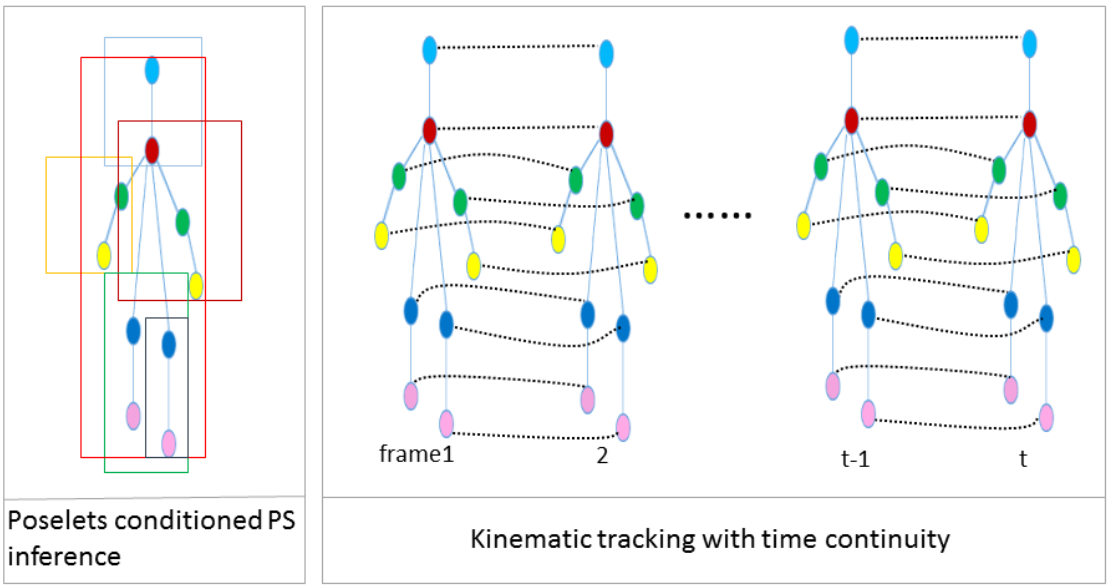


Figure 5.7: (a) shows the pose inference for one frame. The kinematic tracking applying the learned poselets and temporal continuity is shown in (b).

5.3.1 Full body detection

It is noted that, before applying the temporal and poselet-conditioned PicStr models for pose estimation and tracking, a pre-processing stage is incorporated to crop the region of interest (ROI) in each frame, hence reduces the search space for the PicStr models. A full body detector with part-based models is chosen to deal with this pre-processing step, which has been described in Section 2.1.2. For the reader's convenience, we recap here briefly how the detector works and how it is implemented.

The model of the full body detector is defined by a coarse 'root' filter similar to the Dalal-Triggs filter on histogram of oriented gradients (HOG) features Dalal and Triggs (2005) which approximately covers the full body, and a series of higher resolution part filters that cover smaller parts of the human body. In implementation, the part filters capture features at twice the spatial resolution compared to the features of the root filter. The part filters are collected by a graphical model with deformation prior (Figure 5.8 (7)).

An hypothesis of the detection specifies the location of each filter in the model, $z = (p_0, ..., p_m)$, where p_i is the position for the i_{th} filter. At a particular position of

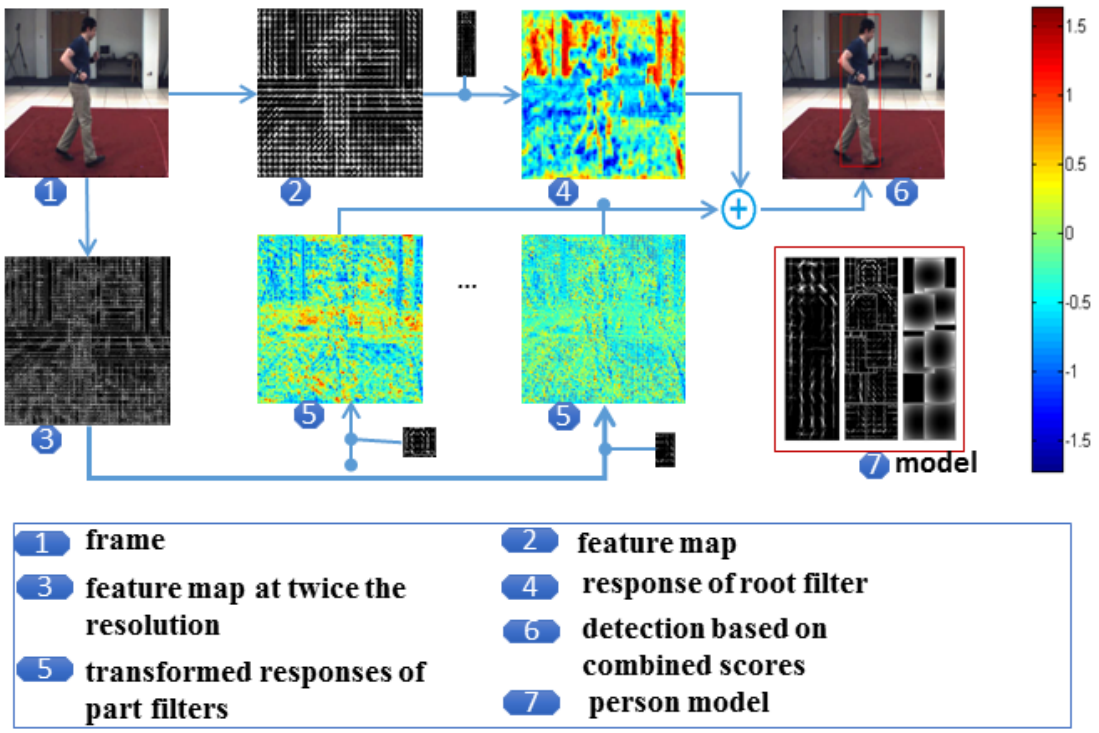


Figure 5.8: Detection process. Here only the transformed responses for the head and left-lower leg are shown. The model shown in (7) includes three components: a coarse root filter, several higher resolution part filters, and a spatial model for the location of each part relative to the root.

an image, the score of a hypothesis is computed by the response of the root filter plus the sum of the transformed responses of each part filter (Figure 5.8 (1)-(6)). Note that the transformed responses are obtained from the responses of part filters minus a deformation cost that depends on the relative position of each part with respect to the root (the spatial prior).

The score of a hypothesis z can be expressed in terms of a dot product between a vector of model parameters β and a feature vector $\psi(H, z)$ as:

$$score(z) = score(p_0, ..., p_m) = \beta \cdot \psi(H, z). \tag{5.6}$$

Here, β is obtained by concatenating the root filter, the part filters, and the deformation cost weights; H is a feature pyramid; $\psi(H, z)$ is a concatenation of subwindows from the feature pyramid and part deformation features. Detecting a person in an image means to find a root location with high score and the corresponding part locations with optimal displacements:

$$score(p_0) = \max_{p_1,...,p_m} score(p_0, ..., p_m). \tag{5.7}$$

The detector is implemented with the deformable part-based model (DPM) framework and the publicly available software Girshick *et al.* (2012) is utilized. Further implementation details can be found from Felzenszwalb *et al.* (2010).

The detection result in an image is defined by a bounding box (bbox) $B = (x_1, y_1, x_2, y_2)$ with the upper-left and lower-right corners being at (x_1, y_1) and (x_2, y_2) respectively. In order to avoid the possible impact of imperfect bounding box boundaries or the false positive detections, the bounding boxes are enlarged by 10 pixels vertically and 15 pixels horizontally in the original images. To ensure the tracking to be invariant to the size of the human body appeared in different images, the bounding box area is cropped out and resized to a patch with a normalized height h that is derived from the scale-normalized training set for PicStr. In this work $h = 200$. The normalized bounding boxes form the final ROIs.

5.4 Experiments and Discussion

In this section we evaluate the performance of the proposed framework on two well-known datasets and report the comparison results with other approaches from the literature.

Datasets and Evaluation Metric.

Datasets: We first evaluate our method on a sequence (HE_combo_S1) from the HumanEva dataset Sigal *et al.* (2010). It shows a person moving in a circle and contains several non-lateral motions, such as jumping, kicking, leaning and stretching. They are used to demonstrate that the proposed system can be widely applied to different motions, with different viewing angles and different number of frames. Note that here we choose 320 frames from the Combo sequence (600 frames) for the experiment in this chapter. We also evaluate our approach on the 'Baseball' dataset from Ramanan *et al.* (2007). The Baseball dataset is a sequence of 200 frames that records a pitcher throwing out a ball.

PCP Metric: As discussed in the previous chapter, to numerically evaluate the performance, we use the well-known PCP metric proposed in Ferrari *et al.* (2008). Specifically, a body part l_m is considered correctly localized if the endpoints of its segment lie within 50% of the ground-truth segment length. This is commonly used as an evaluation metric in bottom-up human pose tracking.

The experimental results are evaluated qualitatively and quantitatively.

5.4.1 Framework components evaluation

There are two essential parts involved in our temporal tracking framework, *i.e.,* the pre-process (ROI selection, abbreviated as ROI) and the poselets constraints (shortened as PL).

We implement a series of experiments to evaluate the performance of various components in our framework using a step-by-step manner. Specifically, we first implement

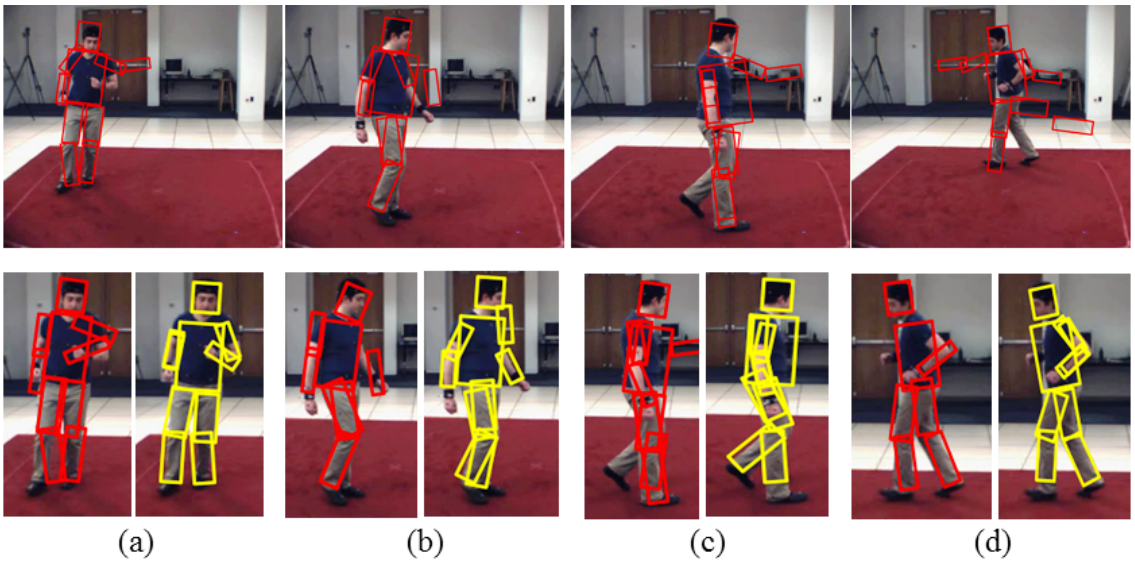


Figure 5.9: The screenshots of tracking results for HE_Combo_S1 sequence from the proposed framework with different components. Row 1 shows the results from the temporal tracking framework with PicStr only. Bounding boxes coloured with red and yellow shown in Row 2 are the tracking results from PicStr+ROI and the whole framework PicStr+ROI+PL, respectively.

the temporal tracking based on the generic tree-structured pictorial structures (PicStr) model only. Then the ROI selection component is combined with the temporal PicStr-based tracking (i.e., PicStr+ROI). Finally, the temporal tracking framework (i.e., PicStr+ROI) is augmented with poselets constraints to form the complete tracking framework (i.e., PicStr+ROI+PL).

Several screenshots of the tracking results from the frameworks with different components on Combo sequence are shown in Figure 5.9. Row 1 shows results from the temporal tracking framework with generic PicStr model only and the selected screenshots show that the results are affected severely by background clutter. When combined with the ROI selection before temporal tracking, some errors from the background clutter such as the arm-like shape in the door or the straight line in the floor can be eliminated in a large extent, which is clearly seen from the red-coloured bounding boxes shown in the second row of Figure 5.9. Moreover, the follow-up temporal PicStr-based tracking process only detects and searches the pose from the ROI area, which largely decreases the search space and improves the computational efficiency. The tracking performance from the whole proposed framework (Pic-

Table 5.1: The performance comparison for Combo sequence (in percentage).

Method	Tor	Head	U.L.		L.L.		U.A.		F.A.		Total
PicStr	98.4	94.5	84.7	83.2	81.8	78.0	81.0	82.5	73.1	72.8	83.0
PicStr+ROI	100	100	86.3	85.5	82.2	81.1	87.1	84.7	85.5	83.0	87.5
PicStr+ROI+PL	100	100	99.3	98.7	93.8	90.0	94.2	94.0	93.1	89.8	95.3

Str+ROI+PL) is demonstrated in the second row of Figure 5.9 with yellow colour. In addition to further error corrections from background clutters, as shown in column (c), the combined poselets constraints also increase the accuracy by eliminating most of the double counting problems, e.g., the lower-leg cases shown in column (b) and (c) of Figure 5.9, which is benefited from the multiple parts dependencies modelled with the poselet representation.

The quantitative performance of the frameworks with different components on Combo sequence are shown in Table 5.1. Compared with the performance of the temporal generic PicStr-based tracking framework (PicStr), the PicStr+ROI system improves the total accuracy by 4.5% and the framework combined with PL component (PicStr+ROI+PL) further increases it by 7.8%. The PicStr+ROI system improves the accuracy significantly for forearms by more than 10%. The whole system (PicStr+ROI+PL) increases the accuracy for all body parts.

The ROI part is an essential pre-processing in our work, which can decrease the search space significantly, thus improve computational efficiency. The ROI component alone even outperforms the effort to improve the tracking accuracy by perfecting the appearance model, such as in Lu *et al.* (2012a), the results of which on Combo sequence are shown in Table 5.2.

5.4.2 Comparison to state-of-the-art approaches

To further evaluate the performance of the proposed framework, we also implement a series of experiments to compare its performance against other state-of-art tracking-by-detection frameworks. The first one is from Ramanan *et al.* (2007). The second one is the specific (colour-based) appearance model from Lu *et al.* (2012b), which is built by clustering the generic (edge-based) part detections across all frames. We

Figure 5.10: The screenshots of tracking results for HE_Combo_S1 and Baseball sequences from the frameworks (Ramanan *et al.* (2007), *Cluster* Lu *et al.* (2012b), *Classifier* Lu *et al.* (2012a)) and the framework proposed in this work.

refer to this algorithm as *Cluster*. The last one is referred to as *Classifier*, which is an accurate specific appearance model from Lu *et al.* (2012a). The source code packages implemented are from the respective research teams.

The performance of the tracking results from these algorithms (*Ramanan*, *Cluster*, *Classifier* and the proposed framework with all components in this work) are shown in Figure 5.10 and Table 5.2.

In Figure 5.10 Row 1 to Row 3 show visually some screenshots for tracking by *Ramanan*, *Cluster* and *Classifier* respectively, and the last row shows the results from the proposed framework. The performance of the proposed approach is clearly much superior. It corrects most of the double counting problems, e.g., the lower-leg cases shown in column (a) and (b) of Figure 5.10, which is mainly benefited from the multiple part dependencies modelled with the poselet component. Moreover, the proposed system can deal with some errors from the background clutter such as the arm-like shape in the door, as illustrated in column (c) and (d), due to the ROI-selection.

Table 5.2 shows the performance comparison of the frameworks on two datasets. The

Table 5.2: The performance comparison based on PCP-metric in percentage.

Dataset	Method	Tracking Performance						
		Tor	Head	U.L.	L.L.	U.A.	F.A.	Total
Baseball	*Ramanan*	98.5	70	58.3 55.2	77.9 71.1	76 35.8	77.9 48.9	67
	Cluster	100	97	93.4 92.1	89.4 88.1	93.8 92.4	91.5 90.8	92.85
	Classifier	100	100	94.1 93.4	92.9 90.6	94.7 93.5	94.2 92.6	94.6
	PicStr+ROI+PL	100	100	96.6 95.1	94.0 92.7	94.5 93.8	91.5 90.1	94.8
HE_Combo_S1	*Ramanan*	52.5	36.3	53.8 65.0	68.8 63.4	24.3 26.7	33.8 42.5	46.7
	Cluster	100	98.0	83.8 81.8	82.6 77.9	83.7 78.4	78.7 75.9	84.1
	Classifier	100	98.4	87.1 86.8	85.0 81.3	86.3 70.7	85.5 70.0	85.1
	PicStr+ROI+PL	100	100	99.3 98.7	93.8 90.0	94.2 94.0	93.1 89.8	95.3

proposed system outperforms the others in every single case. For both sequences, all frameworks achieve good performance in tracking the torso part. However, for the other body parts, i.e., the limbs, the correct rates of the proposed system (more than 90%) is much higher than those of the other systems. It clearly demonstrates that the multiple parts dependencies modelled in this work can significantly improve the performance of tracking and is much more robust. The proposed framework works much better for the detection of smaller and more active body parts, such as arms and legs. The motion in the Combo sequence is much more complex than those in the Baseball sequence, hence the existing methods (*Ramanan*, *Cluster* and *Classifier*) perform worse for this sequence. However our proposed framework is able to produce highly satisfactory results for both sequences, which clearly demonstrates the superiority of the proposed system.

5.5 Summary

In this chapter, we propose a robust framework for human pose tracking in 2D monocular image sequences. A model is proposed to incorporate higher order dependencies of multiple body parts, even if they are not directly connected, which allows the body part connections to be more flexible and specific to image evidences. In order to establish the image-conditioned variables, the effective poselet features are employed. Based on a series of detectors, the poselet descriptors can be computed for each frame. These image-specific terms can be combined to the pictorial structures model, which leads to highly accurate and efficient computation and inference. A simple motion constraint is also incorporated to capture temporal continuity of body parts between frames, which makes the positions of body parts

change smoothly across all frames. In addition, a full body detector is introduced as the first step of our framework to reduce the search space for pose tracking. The proposed framework is evaluated on two challenging image sequences and compared against existing state-of-the-art approaches. The results illustrate that the proposed framework outperforms the state-of-the-art 2D pose tracking systems.

In the next chapter, we will continue to exploit the non-connected body part dependencies by augmenting the PicStr model from the same layer. Moreover, the confusion between the left and right limbs will also be discussed in the next chapter.

Chapter 6

Symmetric Body Part Dependencies Modelling and Head Orientation Estimation

Most approaches in 2D human pose tracking rely on the simple tree-based pictorial structures (PicStr) model, in which the appearances of body parts are modelled with a set of unary terms and the spatial arrangement between adjacent body parts is captured by a group of pairwise terms.

The generic tree-based PicStr methods used for pose estimation and tracking are successful on images where all the limbs of the person are visible. However, as described in Chapter 5, the main drawback of such approaches is that the simple tree structure completely excludes all the dependencies among non-adjacent body parts. As a consequence, the detections for some limbs from these approaches can be inaccurate especially in scenarios involving a cluttered background or self-occlusion. Moreover, double counting is another common problem with the tree-based PicStr model especially when the tracked targets show sideways poses. Symmetric body part pairs often appear in close proximity in images hence share a high detection score with the same image evidence.

In Chapter 5, a model incorporated higher order dependencies of multiple body parts has been proposed, which provides more conditions onto the generic PicStr model from a higher level and make the PicStr model more specific to images. In this chapter we propose methods to further improve the tracking performance by adding more flexibility to the generic PicStr model. One of the methods proposed in this chapter augments the tree structured PicStr model by adding more dependencies between symmetric and non-adjacent limbs, which are in fact important factors for balancing and coordination. In other words, we propose a framework based on pictorial structure that not only encodes the information based on relations between

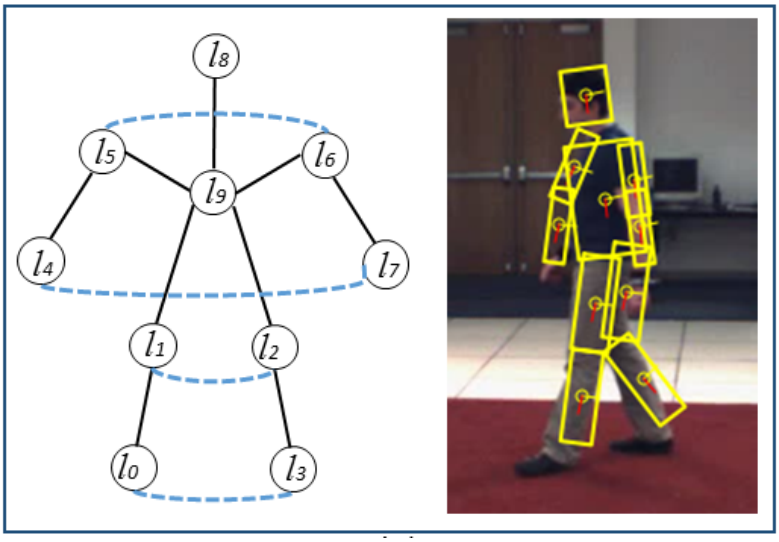


Figure 6.1: The proposed model integrates the relation information between symmetric legs and arms, i.e., the blue dash lines illustrate the augmented dependencies between symmetric body parts.

connected body parts, but also integrates additional constraints (AdCon) between symmetric limbs. The model incorporated AdCon utilized in this chapter is shown in Figure 6.1. It can be seen from the tracking results shown in the frame on the right of Figure 6.1 that the proposed approach can effectively deal with the problem associated with the double counting.

As shown in Figure 6.1, it is clear that the new model is no longer a tree structure since loops are introduced. To effectively model dependencies between non-adjacent body parts while still allowing for efficient inference, we use the factor graph Kschischang *et al.* (2001) to represent the whole human body model and libDAI Mooij (2010) is employed to implement inference for this graphical model.

A factor graph is an easy and straightforward way to add more variable and factor nodes which can be used to represent more dependencies between different body parts. We exploit the inference relying on the factor graph approach, in which the whole model is encoded by a set of variable and factor nodes. Section 6.1.1 describes the factor graph utilized for this work in detail.

Another serious problem occurring frequently during 2D human motion tracking is the confusion between the left and right limbs. For example, Figure 6.2 shows several

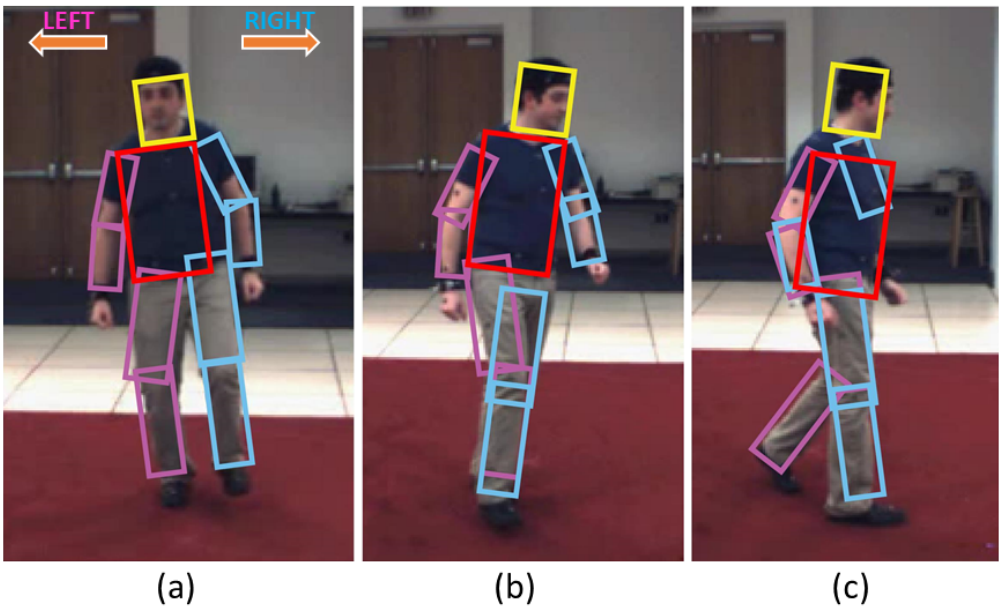


Figure 6.2: Several tracking results. In (b) and (c), the detections of the left and right legs are clearly inconsistent with those in (a) because of occlusion and the double counting issues.

tracking results for frames in a walking sequence (named HE_walking_S1) from the well-known tracking dataset HumanEva Sigal *et al.* (2010). In this sequence, the tracked target walks around a circle. Although the frontal (or back) poses can be accurately detected especially when all body parts are visible, the left and right limbs are often confused especially when the human body is side-faced. For example, in Figure 6.2 (a), the pink colour bounds the left limbs while the bounding boxes in light blue are for the right limbs. The left and right legs are incorrectly identified for the side-faced poses shown in Figure 6.2 (b) and (c). The trajectories of the right/left upper legs (lul/rul) for the sequence HE_walking_S1 are also illustrated in Figure 6.3. From the frontal poses to back-facing, it is clearly noted that the right and left limbs shown in these trajectories only represent the relative locations of limbs in images without considering the consistency issue, which is a big problem during tracking. For example, the left leg in the frontal pose is encoded with blue colour in the trajectory, but the same left part in some back-facing poses is recognized as right-side leg and coloured with orange.

The reason for the left/right inconsistency problem is that the left and right body parts in the generic PicStr model are defined only depending on their relative locations in the image coordinate system because the PicStr is originally designed

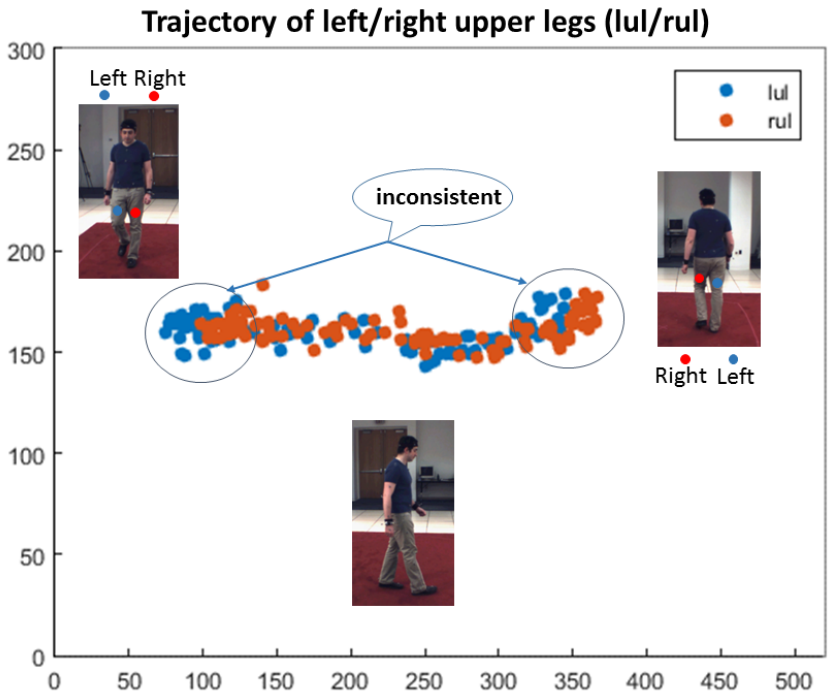


Figure 6.3: Tracking trajectories of upper legs from frontal poses to back-facing for the sequence HE_walking_S1. It is clearly noted that the real right and left limbs are not consistent with the recognized left and right shown in the trajectory. For example, the left leg in the frontal pose is encoded with blue in the trajectory, but the same left part in backwards poses is recognized as right-side leg and coloured with orange.

to infer body poses for one frame. When dealing with a video sequence, confusion between the left and right limbs often occurs especially when poses from different viewpoints exist in the sequence since their appearances are similar.

In this work, we propose a very simple method based on estimation of the head orientation (looking left or right) to provide instructive information to address such a problem. A head-yaw-estimation step is introduced into the tracking framework to serve as a complementary tool to assist the human pose estimation. Yaw rotation of the head is one important type of head poses and it attracts much attention because its estimation has many potential applications Ma *et al.* (2015). In this work, accurate estimation of the head yaw angle is not necessary. We only need a brief indication on whether the human body is roughly facing left or right. A simple skin colour detector and a set of threshold templates is hence used to indicatively identify the head orientation of the tracked target. Such information is used to determine the visible side of the body, hence in this work we only deal with side-facing poses in which the orientations of the head and body are consistent. We believe this covers most side poses in most human motion. Postures which do not satisfy such an assumption are not considered in this work.

Generally, self-occlusions are more likely to happen when the tracked targets are side-facing. With the head orientation determined, if the human figure is not frontal or backward facing, the system can first determine whether the left or right side of the body is certainly visible and then deduce the situation of the other side based on image evidences. For example, for the side-facing pose in Figure 6.2, the head pose detector indicates that the pose is facing right. The system will then determine that the left side of the body is definitely visible. Accordingly, the system will grant higher priority for the left body parts and assign the posterior with higher score to them. The posteriors for the right side limbs will then be searched and located with reference to their left counterparts. If all posteriors for a certain right body part score very low, this part will be regarded as being occluded. Therefore, the double counting problem can be effectively avoided and the confusion between left and right body parts is cleared.

The structure of this chapter is as follows. In Section 6.1, the repulsive factors, i.e., additional constraints (AdCon), between symmetric body parts are introduced to the PicStr model. The details on how to factor the terms of the whole model

mathematically is also presented in this section. Then the proposed head facing orientation (HeadOri) estimation approach is described in Section 6.2. After that, the complete tracking process with the proposed tracking framework is presented in Section 6.3. The components included in the complete tracking framework are the improved PicStr model with constraints on symmetric body parts, head facing orientation estimation and the poselets conditions proposed in previous chapter. The experiment settings and experimental results are shown in Section 6.4. Finally we conclude this chapter in Section 6.5.

6.1 Repulsive Factors

6.1.1 The PicStr model with additional constraints (AdCon) between symmetric body parts

While the generic PicStr model often leads to competitive results, there are situations in practice that the tree structure for the human body cannot be clearly observed such as in the case of body parts occluding each other. In order to improve the tracking performance especially for the challenging limbs, we propose a framework in this chapter that augments the tree structure model by considering dependencies between symmetric and non-adjacent body parts, which are actually important in real life for balancing and coordination. Specifically, in the proposed framework,

$$p(L_t|D_t) = \sum_{i=1}^{N} E^u(l_t^i; D) + \sum_{m \sim n} E^p(l_t^m, l_t^n; D). \tag{6.1}$$

where, E^u is the unary term and E^p is the pairwise term of the proposed model, and $m \sim n$ denotes the relationship between the body parts m and n. Same as the generic PicStr model, the unary term denotes the image likelihood based on a set of pre-trained shape-based appearance models for all body parts. However, the pairwise term here is different from the generic PicStr model. Specifically, the pairwise term $E^p(l_t^m, l_t^n; D)$ represents relations between the connected and non-connected body parts while the pairwise term in the generic PicStr model only represents information on relations between the connected body parts.

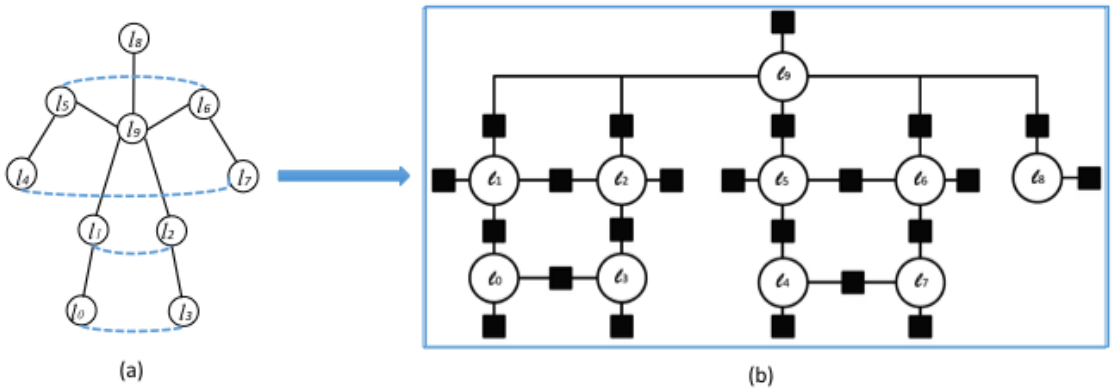


Figure 6.4: The model shown in (a) is the generic tree-based PicStr model (drawn in black colour) integrated with constraints between symmetric legs and arms (the blue dash lines). The whole model is transferred to a factor graph shown in (b). Each part l_i is represented by a variable node (empty circle), a factor node (solid square) denoting each local function f_j, and an edge connecting a variable node l_i to a factor node if and only if l_i is an argument of f_j.

Obviously loops are introduced when constraints are added between symmetric and non-adjacent body parts. Thus the original inference approach for the tree based model cannot be utilized for the proposed model. In order to guarantee high computational efficiency while taking into account the dependencies between body parts as much as possible, the factor graph method is used in this research.

Figure 6.4 shows a transferred factor graph from a PicStr model augmented with additional constraints between symmetric legs and arms. Each body part is represented by a variable node (empty circle), a factor node (solid square) denoting a local function (f_j), and an edge connecting a variable node to a factor node if and only if the variable node is an argument of the factor node. All the unary terms and the dependencies of the proposed model are factorized in Figure 6.4.

6.1.2 Factorization of the model

A factor graph is a bipartite graph representing the factorization of a function. In probability theory and its applications, factor graphs are used to represent factorization of a probability distribution function, enabling efficient computations such as the computation of marginal distributions through the sum-product algorithm.

Factor graph provides an easy and straightforward way to add more variable and factor nodes representing more dependencies between different body parts. Both the unary term and the dependencies modelled in the pairwise term of Equation (6.1) should be factorized in a factor graph.

First of all, the unary term represents the image likelihood and is written as

$$E^u = \prod_{i=1}^{N} f(l_i) = \prod_{i=1}^{N} p(d_i|l_i) \tag{6.2}$$

which is resulted from a set of appearance models for all body parts. In Figure 6.4, unary terms correspond to variable vertices l_i, the factor vertices (solid squares) f connecting only one variable, and undirected edges connecting these variable vertices to factor vertices.

Besides the dependencies between the connected parts, another four repulsive factors are incorporated in the proposed model to represent the dependencies between symmetric and disconnected limbs, i.e., the factors between left/right upper/lower arms and legs. Thus the pairwise term in Equation (6.1) can be written as

$$E^p = f(l_0, l_3)f(l_1, l_2)f(l_4, l_7)f(l_5, l_6) \prod_{(l_i,l_j)\in E} f(l_i, l_j) \tag{6.3}$$

where E is the edge set containing all connections between adjacent body parts and $f(l_i, l_j)$ is the kinematic dependencies between connected body parts, and all body parts arrangement is shown in Figure 6.4.

The additional dependency factors in the proposed model is defined as

$$f(l_m, l_n) = \begin{cases} exp(-\alpha) & IoU(l_m, l_n) > \gamma \\ 1 & otherwise \end{cases} \tag{6.4}$$

where $IoU(l_m, l_n)$ is the ratio of intersection and union of the bounding boxes of part m and n, γ is a threshold controlling when the defined factor should take effect. The additional factors tend to push two symmetric limbs away from each other, which intend to serve the purpose of avoiding double counting in the same image region. The parameter α defines how strong the parts are pushed away from each other. In this research, we set $\gamma = 0.3, \alpha = 1.5$ for legs and $\gamma = 0.2, \alpha = 1.5$ for arms.

Factorizing all factors in Equation (6.1), the transferred factor graph is illustrated in Figure 6.4 (b), where each part l_i is represented by a variable node (empty circle),

each local function f_j is represented by a factor node (solid square), and an edge connects a variable node l_i to a factor node if and only if l_i is an argument of f_j.

Different from the tree structure, loops are introduced in the proposed structure. The whole inference process thus consists of two steps: firstly inference is performed using the model with only tree-based kinematic constraints, and then the posterior marginals of each part are sampled and inference is performed with the full model using the samples as the new state-space. The inference for the proposed model utilizes an open source C++ library libDAI Mooij (2010) that implements various (approximate) inference methods for discrete graphical models. LibDAI supports arbitrary factor graphs with discrete variables.

6.2 Estimating the Head Facing Orientation

The goal of this step in our work is merely to roughly decide whether the tracked target is facing left/right/front/back. It is worth noting that estimating the accurate yaw angle of the head is not necessary for this purpose. Although there are many algorithms published on the accurate estimation of the head pose (Demirkus *et al.* (2014); Zhu and Ramanan (2012)), they are considered unnecessary and computationally overkill for our purpose. Rather, a very simple algorithm based on colour detection is utilized to identify the rough head orientation of the tracked target, namely, whether he/she is roughly facing front, back, left or right.

Given the head bounding box for each frame shown in the first column of Figure 6.5, it is noted that the absolute location of the face area (C) or the relative location of the face and hair regions (ρ) are essential clues for the head orientation estimation.

Considering the generally small size of the head area in this kind of applications and to ensure simplicity of the algorithm, we utilize a skin colour detector to select the face area which can produce a binary skin-map and highlight patches of skin-like pixels for a given image (see Figure 6.5 (a)). The hair region is not detected separately. The head image is firstly transformed from RGB colour space to YCbCr colour space and the resultant image is comprised of intensity component (Y) and

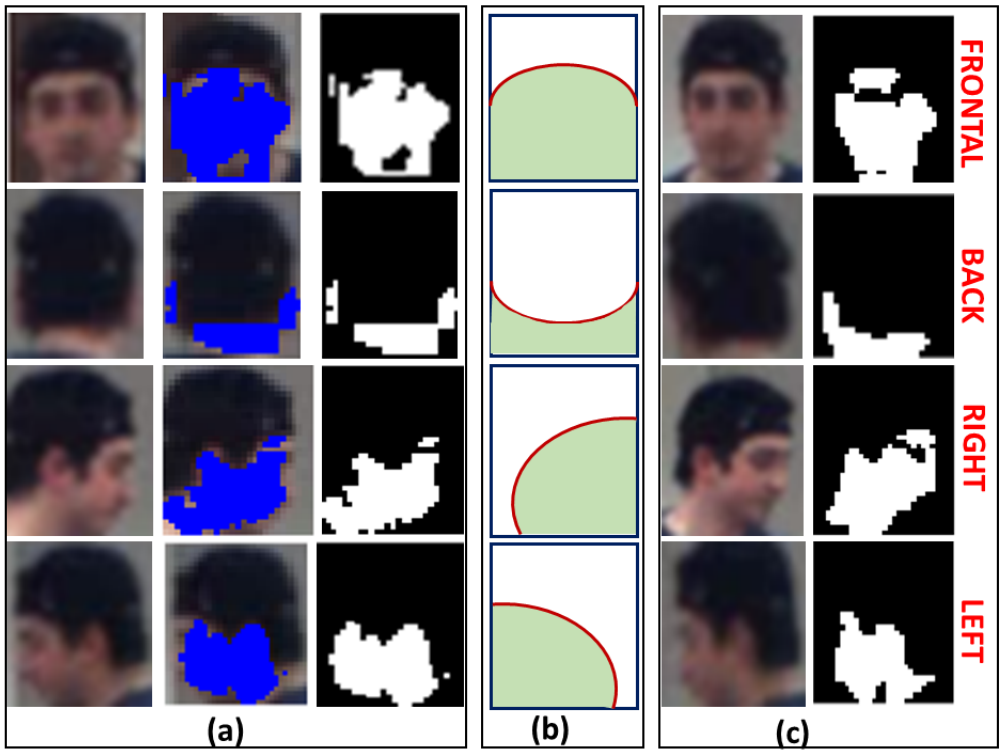


Figure 6.5: The face detector is shown in (a). The skin-map is overlayed onto the image marked in blue. The binary skin-maps are shown in the third column. (b) shows the set of face templates. Examples of estimation results with the face templates are shown in (c).

chrominance components (Cb and Cr).

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{6.5}$$

The YCbCr colour space is chosen since it is effective and efficient for the separation of image pixels in terms of colour and can be applied for complex colour images with uneven illumination.

For determining the head orientation $H(C, \rho)$, a set of templates for different head orientations are pre-set and illustrated in Figure 6.5 (b). The boundary of the face area is assumed to be roughly elliptic and the region intersecting with the head bounding box is considered the face area, which is marked in green. Intuitively, in the head bounding box, if the skin-coloured region accounts for a great part of the

bounding box and is locates in the central, it is considered that the target is front facing. In contrast, if there are only a small number of skin-like pixels and the region they represent locates at the bottom of the head bounding box, the tracked target is considered to be back-facing. If the skin area is located only in the right-bottom or the left-bottom area of the bounding box, we consider the head is looking right or left and hence the tracked target is assumed to be in a sideway pose, facing right or left accordingly. As mentioned before, in this work, it is assumed that the pose orientations are always consistent with the head orientation during tracking.

6.3 Human Pose Tracking

Given an image sequence, our task is to infer the posterior $p(L_t|D_t)$ across all frames, i.e., to estimate the optimal tracks of each part, which corresponds to find the maximum *a posteriori*

$$L^* = \underset{L}{argmax}(p(L_{1:T}^{1:N} \mid D_{1:T}^{1:N})). \tag{6.6}$$

Taking into consideration the time coherence and the head orientation, the posterior in the proposed complete model for pose estimation in each frame can be formulated as

$$p(L_{1:T}^{1:N} \mid D_{1:T}^{1:N}) \propto \sum_{t=1}^{T} \sum_{i=1}^{N} \left(\phi(l_t^i, l_{t-1}^i) + E^u(l_t^i; D, H) + E^p(l_t^m, l_t^n; D, H) \right). \tag{6.7}$$

Here superscripts are used to represent body parts and subscripts for frames, e.g., l_t^i is the configuration of the part i at the time instance t. The first term in the right-hand side of Equation 6.7 captures temporal continuity of body parts between frames. The last two terms are the unary term and pairwise term of the proposed model. Here the pairwise term represents the relationships between both the connected and non-connected body parts. H is the head orientation indicator.

In the proposed framework, we first perform inference using the model with only tree-based kinematic constraints, and then sample from the posterior marginals of each part according to Gaussian prior, temporal filter and head orientation information.

Through experiments, we observe that the posterior marginals of each part in the

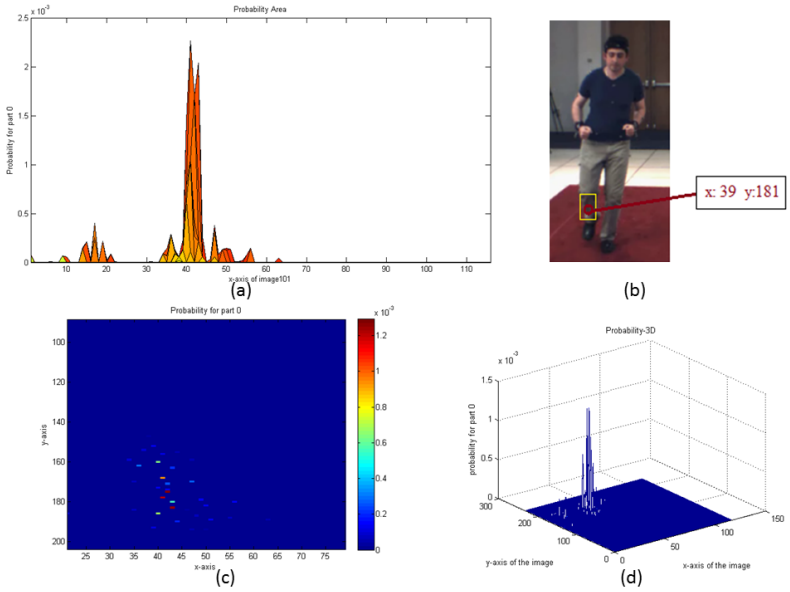


Figure 6.6: The yellow bounding box in(b) is the ground-truth location of the left lower leg (LLL). (d) shows the posterior probabilities of LLL in image coordinates, which is mapped to image (c) in the 2D space. It can be clearly seen that the posteriors form a Gaussian distribution.

tree structure model approximately satisfy a Gaussian distribution, which means that the closer the candidate location is to the ground truth of the target location, the more possible it is to be chosen as the final tracked location. An example for left lower leg is illustrated in Figure 6.6.

It is obvious that the most probable posteriors are around the ground-truth location of the body part. Therefore, the posterior marginal set (named M_t^i) can be sampled based on this fact, i.e., more posteriors are sampled at the locations near the centre of the tracked body part in the previous frame. The marginal set sampled this way is named $M_t^i\_g$.

Since the position/orientation of each body part generally changes smoothly within an image sequence, a simple motion model is exploited, taking advantage of this temporal continuity, by setting a simple velocity threshold, similar to the one utilized in Chapter 5. That is, the Euclidean distance between the part positions in successive frames should not exceed a pre-set threshold:

$$\phi(l_t^i, l_{t-1}^i) \propto \mathcal{I}(\mathcal{D}(l_t^i, l_{t-1}^i) < d_{max}), \tag{6.8}$$

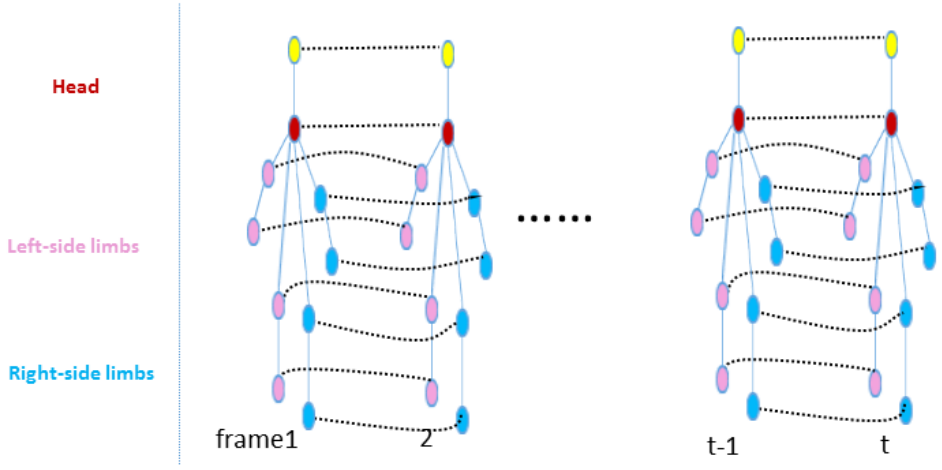


Figure 6.7: The tracking process. For each frame, the tracking system first determines whether the left or right side of the human body is certainly visible according to the head orientation and then grants higher priority to the visible side for the subsequent pose inference.

where $\mathcal{I}$ is the standard identity function and $\mathcal{D}(l_t^i, l_{t-1}^i)$ is the Euclidean distance between the part positions in consecutive frames. Besides, the orientation difference of each body part in two successive frames should not exceed a pre-set threshold θ_max.

To avoid left/right inconsistency during tracking, the proposed head detector is implemented to roughly estimate the head (and subsequently the overall body) orientation (represented as $HeadOri$). During tracking, the head orientation is firstly estimated using the approach described in Section 6.2 and the body facing direction is then determined based on the estimated head orientation. If the body pose is determined as not being frontal or back-facing, the system can first determine whether the left or right side of the human body is certainly visible according to the head orientation and grant higher priority to that side for the subsequent posterior sampling and pose inference. For example, if the head pose detector indicates that the pose is facing right, the system can identify that the left side is definitely visible. Accordingly, the system will grant higher priority for the left side and sample posteriors of body parts in the order $[l_0, l_1, l_4, l_5, l_8, l_9]$. The next step is to select the sampled posteriors for the right side limbs $[l_2, l_3, l_6, l_7]$ and allocate them to different places from their respective left counterparts. It is noted that if all posteriors for a certain right body part are small, this part will be regarded as being occluded. In

the contrary, if the pose is facing left, the order will be $[l_3, l_2, l_7, l_6, l_8, l_9]$ and then $[l_0, l_1, l_4, l_5]$. The arrangement of the part order is demonstrated in Figure 6.1.

The tracking process is shown in Figure 6.7.

With the time coherence and the head orientation information, the marginal set $M_t^i\_g$ is then sampled, which further shrinks the candidate marginal set. Here the final marginal set is named as $M_t^i\_g\_s$.

With the marginal set $M_t^i\_g\_s$, inference for the proposed model is performed again to obtain the final pose estimation for each frame.

Algorithm 6.1 Tracking human poses with head orientation estimation

```
Generate all part proposals with part detectors.
Generate head orientation from head yaw estimation step.
for all frames do
  Sample posterior candidates for all body parts
  if is_right − facing(pose) then
    Prioritize the left-side body parts and then infer the right side pose with
    reference to the left side parts.
  else if is_left − facing(pose) then
    Prioritize the right-side body parts and then infer the left side pose with
    reference to the right side parts.
  else
    Directly infer the body pose with 10-part Pictorial Structures Model.
  end if
end for
```

6.4 Experiments and Discussion

In this section we evaluate the performance of the proposed complete framework on several datasets and compare its performance with the framework based on the tree structure PicStr model.

Datasets: Two sequences are used in the experiments. One is the sequence HE_Combo_S1,

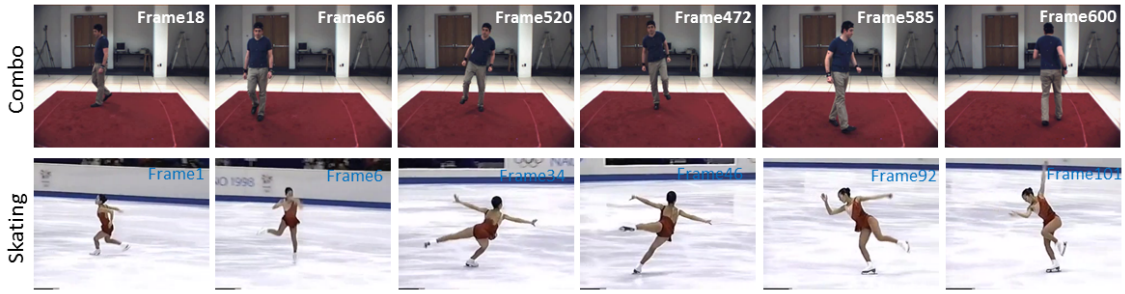


Figure 6.8: Some frames from sequences Skating_S1 and HE_Combo_S1.

which is a 600 frames sequence and the tracked person in it performs different motions in a circle and shows different scales and body orientations. In addition, the sequence contains several non-lateral motions, such as jumping, kicking, leaning and stretching. The other is the sequence Skating_S1, which is selected from a video recording the skater Michelle Kwan performed at the 1998 Olympics Game and consisting of 116 frames. The skater in the sequence Skating_S1 performs different actions with various poses. Several sample frames of both sequences are illustrated in Figure 6.8. They are used to demonstrate that the proposed system can be widely applied to different motions with different viewing angles .

We conduct a series of experiments to evaluate the performance of the proposed framework qualitatively and quantitatively.

6.4.1 Experiments with additional constraints (AdCon)

We first implement a pose tracking framework on the sequence HE_Combo_S1 based on the generic tree-structured pictorial structures model. Then, we augment the tracking framework with the proposed additional constraints, i.e., dependency information between symmetric limbs, and the system is named $PicStr + AdCon$.

To improve computational efficiency and ensure fairness of the comparisons, for the sequence HE_Combo_S1, all frameworks implemented in this chapter use the same region-of-interest (ROI) selection technique to normalize the scale value of all frames.

Several screenshots of the tracking results from both frameworks with different com-

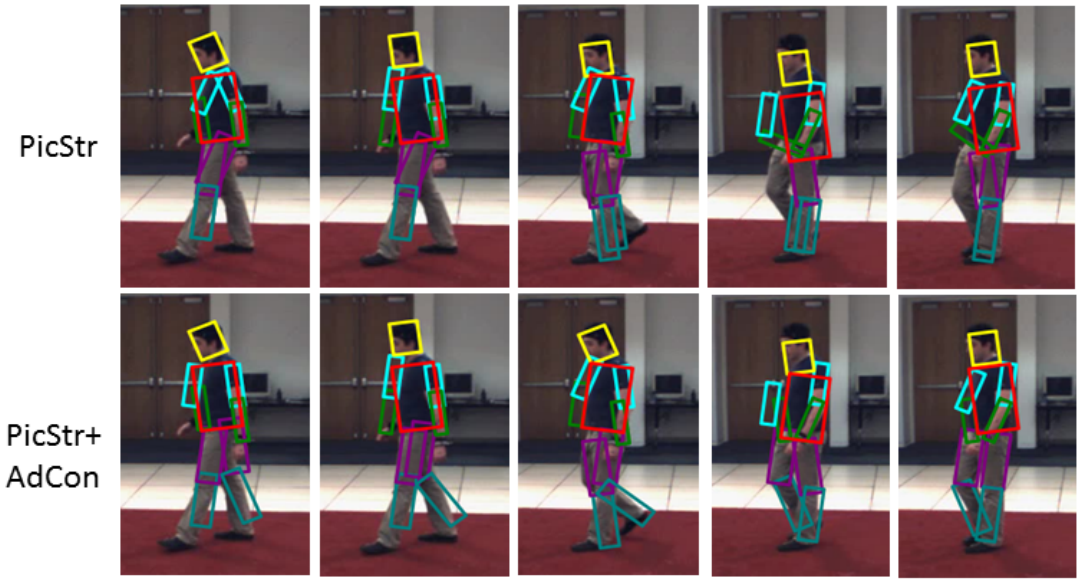


Figure 6.9: Row 1 shows the selected screenshots from the basic pose tracking framework with the generic *PicStr* model. The tracking performance with constraints on symmetric body parts (*PicStr* + *AdCon*) is shown in the second row.

ponents on HE_combo_S1 sequence are shown in Figure 6.9. The selected screenshots show that the results using PicStr only are affected severely by the double counting problem especially for legs.. When augmented with repulsive factors on symmetric limbs (*AdCon*) during tracking, errors from the double counting can be corrected in a large extent, which is clear in the screenshots shown in the second row of Figure 6.9. In other words, the combined constraints on limbs increase the accuracy by overcoming most of the double counting problems benefited from the additional factors between symmetric legs and arms.

The quantitative performance of the tracking frameworks with *PicStr* only and *PicStr* + *AdCon* components on the Combo sequence is shown in first and second rows of Table 6.1. Compared with the *PicStr* system, the framework combined with additional dependencies (*PicStr* + *AdCon*) increases the total accuracy by 6%. It is thus evident that the additional constraints between non-connected body parts is important for improving the tracking performance. More specifically, the inclusion of dependencies between symmetric body parts increase the accuracy for limb tracking by overcome the double counting errors. It is clearly seen from the quantitative comparison that, with the *AdCon* component, the tracking accuracy

Table 6.1: The performance comparison in percentage for HE_combo_S1 sequence.

Framework	Tor	Head	U.L.	L.L.	U.A.	F.A.	Total
PicStr	100	100	86.3 85.5	82.2 81.1	87.1 84.7	85.5 83.0	87.5
PicStr+AdCon	100	100	**100 98.5**	**95.9 94.7**	87.8 86.9	86.0 85.1	93.5
PicStr+PL	100	100	99.3 98.7	93.8 90.0	**94.2 94.0**	**93.1 89.8**	95.3
PicStr+AdCon+PL	100	100	**99.6 98.7**	**95.9 94.7**	**95.8 94.0**	**93.1 90.0**	96.2

for the legs are improved significantly.

It can also be noted through our experiments that the ROI part is an essential pre-processing in this kind of pose tracking, which can decrease the search space significantly, thus improve computational efficiency. The simple ROI component even outperforms the effort to improve the tracking accuracy by perfecting the appearance model, such as the one described in Lu *et al.* (2012a).

In this chapter, we also implement the mid-level poselet representation (described in Chapter 5) onto the tracking framework $PicStr + AdCon$. From the experiment results in Chapter 5, the tracking performance of the system with poselets for arms is more significant than $AdCon$ experimented in this chapter while for legs, the $AdCon$ component is much more effective. Therefore, all parameters on poselets and $AdCon$ are tuned and combined here to play their role as much as possible. That is, the group of poselets related arms are selected to constrain the configuration of arms and the repulsive factors between symmetric legs are utilized to inference the leg configurations.

In implementation, 7 poselets parts $(11 - 17)$ and 2 repulsive factors between left and right upper/lower legs are selected. The quantitative and qualitative comparison between all frameworks are shown in Table 6.1 and Figure 6.10.

The results proved that the system with PL component performs better on tracking arms while the system with $AdCon$ provides higher accuracy on legs. This is due to the fact that the mid-level poselets representation can guide the search for body parts even for small limbs such as arms. It should be true for legs. However in fact, the legs are not only with similar appearance but also adjacent and often crossing or occluded by each other, which limits the effectiveness of the PL component. Compared with the PL, the $AdCon$ on the symmetric body parts are more effective

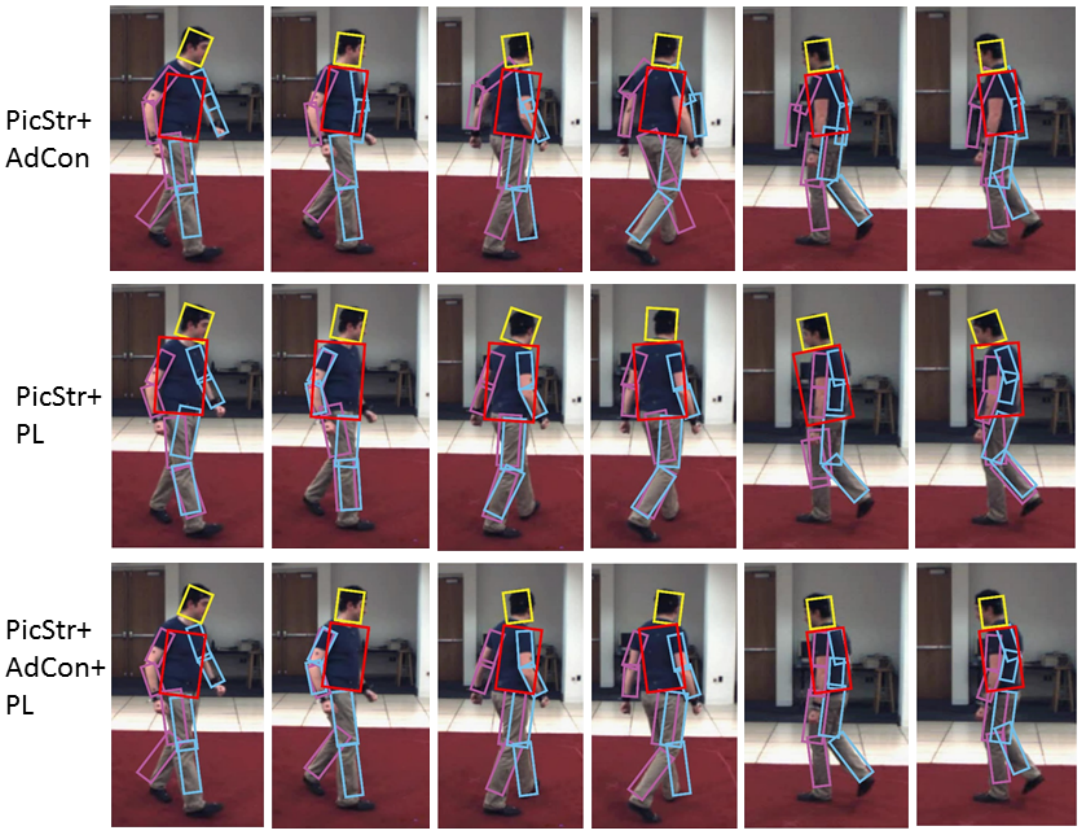


Figure 6.10: Row 1 shows the selected screenshots from the human pose tracking framework with constraints on symmetric body parts ($PicStr + AdCon$). The tracking performance with mid-level poselets constrains ($PicStr + PL$) is shown in the second row. The third row is the performance of the tracking system with components $PicStr + AdCon + PL$.

for legs because they tend to push two symmetric limbs away from each other. When both components PL and $AdCon$ are combined, the accuracy for legs and arms is simultaneously improved, which is clearly shown in Table 6.1 and Figure 6.10.

6.4.2 Experiments with head orientation estimation ($HeadOri$)

In addition to the inclusion of the poselets and the additional constraints on symmetric body parts, the head orientation information is augmented to form the complete tracking framework $PicStr + AdCon + PL + H$.

The complete tracking system is tested with the sequences HE_Combo_S1 and Skating_S1. As mentioned before, for the sequence HE_Combo_S1, the region-of-interest (ROI) selection technique is applied to normalize the scale value of all frames. For the sequence Skating_S1, the Pixel_Count metric is utilized to deal with the scale variation issue during tracking because the ROI strategy is not suitable for the sequence in which not all poses are upright.

Several screenshots of tracking results for both sequences without and with head orientation information ($PicStr$+$AdCon$+PL vs. $PicStr$+$AdCon$+PL+$HeadOri$) are shown in Figure 6.11 and Figure 6.12. With the body orientation changing within the video sequences, if the head orientation information are estimated and utilized as shown in the bottom row of both Figures, the limbs are tracked consistently. On the contrary, without the head orientation information, the left and right limbs are often confused during tracking as shown in the top row.

The improvements is especially evident in the correct tracking of legs for the sequence Skating_S1, in which the body turns frequently. For both sequences, the proposed method reduces the confusion between the symmetric left and right limbs and thus enhances temporal smoothness for each body part, which subsequently results in smoother and more accurate tracking.

The quantitative performance of both the frameworks ($PicStr + AdCon + PL$ vs. $PicStr + AdCon + PL + HeadOri$) on the two sequences are shown in Table 6.2 and Table 6.3. For both sequences, the proposed approach improves the tracking performance consistently for every limb. The left/right consistency in the proposed

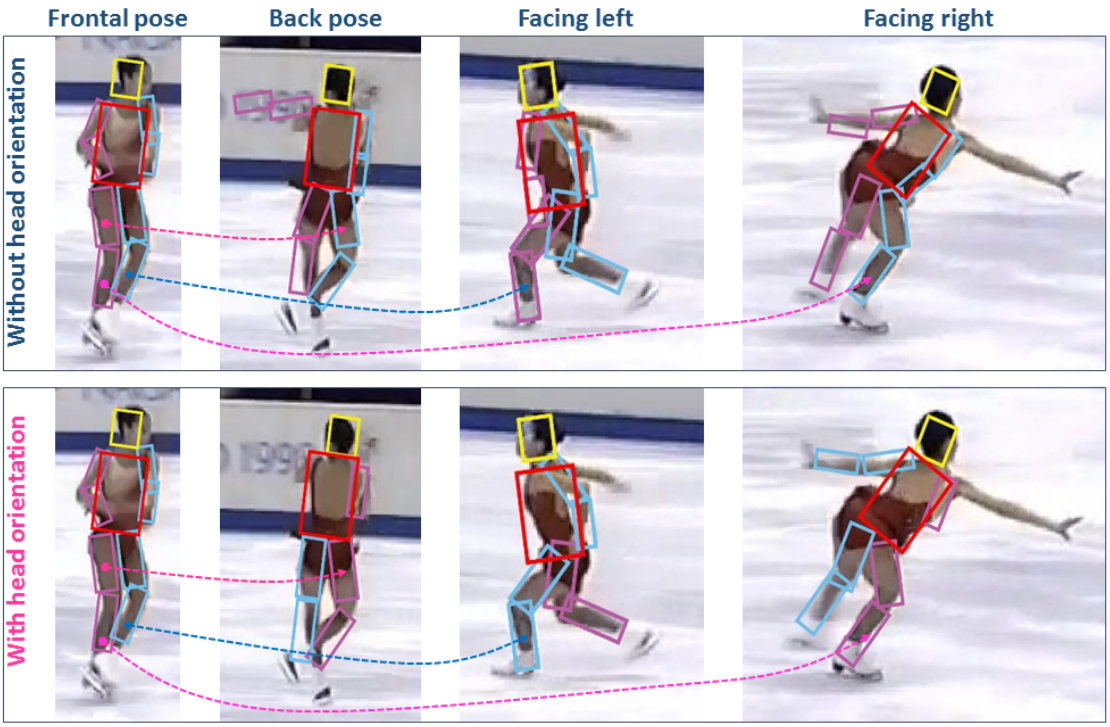


Figure 6.11: Several screenshots show tracking results for the sequence Skating_S1 without and with head orientation information. The screenshots are selected with poses facing front/back/left/right separately. The left and right limb confusions as shown in the top row are corrected when the body orientation information is utilized during tracking as shown in the bottom row. For example, the left upper leg in the frontal pose (coloured in pink) is mistaken as the right upper leg in the back pose (coloured in blue) when tracking without orientation information. With the orientation information, the left upper leg is always recognized as the left-side part no matter which side the body is facing, thus the limbs are tracked consistently.

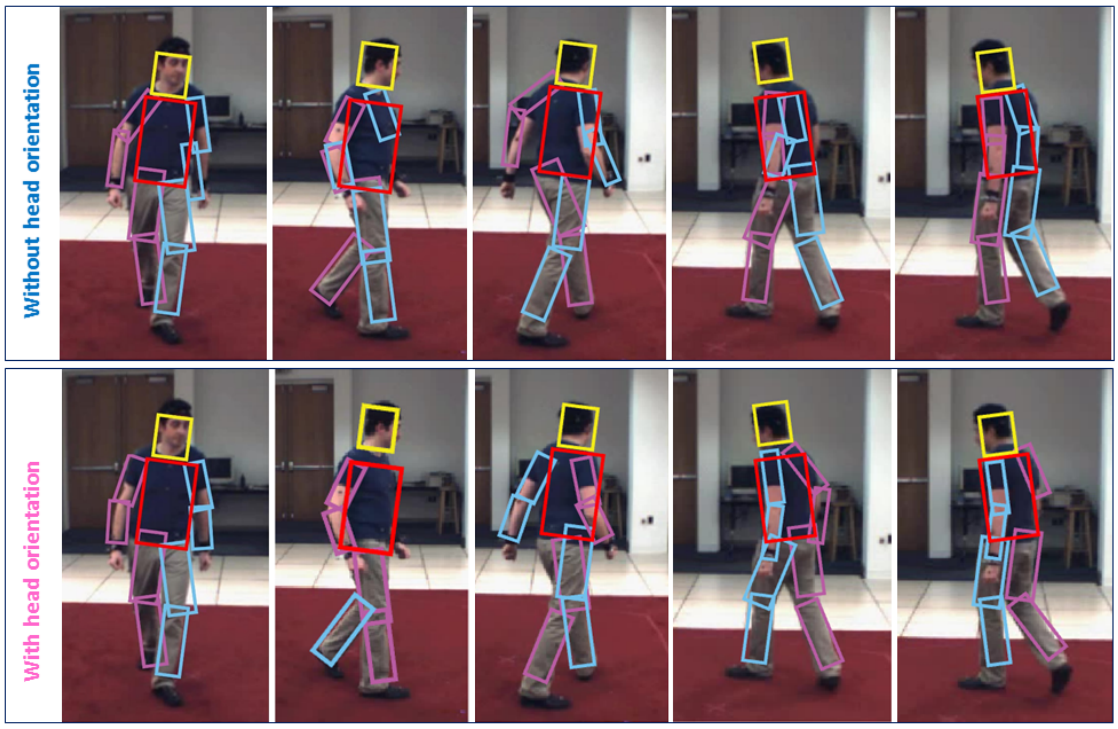


Figure 6.12: Several screenshots show tracking results for the sequence HE_Combo_S1 without and with head orientation information. The left and right limb confusions as shown in the top row are corrected when combing with the body orientation information during tracking as shown in the bottom row.

Table 6.2: The performance comparison in percentage for Skating_S1 sequence.

Framework	Tor	Head	U.L.	L.L.	U.A.	F.A.	Total
PicStr+AdCon+PL	96.1	94.6	88.3 87.5	83.7 84.1	78.2 79.6	75.4 74.3	84.2
PicStr+AdCon+PL+HeadOri	96.1	94.6	89.9 89.8	**88.9 89.7**	79.4 81.0	79.1 78.0	86.7

Table 6.3: The performance comparison in percentage for HE_combo_S1 sequence.

Framework	Tor	Head	U.L.	L.L.	U.A.	F.A.	Total
PicStr+AdCon+PL	100	100	99.6 98.7	95.9 94.7	95.8 94.0	93.1 90.0	96.2
PicStr+AdCon+PL+HeadOri	100	100	99.0 99.0	**97.1 96.8**	96.0 95.4	**94.3 92.5**	97.0

whole framework is improved, thus decreasing the detection errors in the previous framework. Moreover, the proposed inclusion of head orientation estimation into the tracking system increases the accuracy for limb tracking by eliminating the double counting errors caused by self-occlusions.

6.5 Summary

In this chapter we propose a complete framework for human pose tracking in 2D monocular image sequences, which incorporate additional dependencies of non-connected body parts and head orientation information.

A model encoded with additional relationships between symmetric limbs is analysed in this chapter to constrain the left and right arms and legs, encoding the natural human distinction for balance and body coordination. In order to implement the inference efficiently for the proposed model, the factor graph approach is utilized to factorize the proposed model. All the unary term and all dependencies modelled in the pairwise term of the proposed model, both between the connecting and non-connecting body parts, are factorized in a factor graph.

A head orientation detector is augmented into the human pose tracking framework in this chapter. It is a very simple and efficient method, but effectively addresses one of the biggest problems in 2D human pose tracking, i.e., the confusion of the left and right body parts due to the overlapping and occlusion when the human is not facing front or back. A simple head orientation detector based on skin colour

detection is proposed to roughly estimate the head facing direction, and hence the orientation of the whole body. The side of the body that is definitely visible can then be determined and given higher priority for the subsequent inference. The other side will then be inferred with reference to their visible counterparts, with some body parts confidently determined to be occluded.

The proposed complete framework is evaluated on two challenging image sequences and compared against the framework with the tree-based PicStr model. Experimental results show that the proposed framework is able to achieve very high detection rate for very complicated video sequences involving large variations of motions and orientation. Another key advantage of the proposed framework is that it has a tunable model structure, i.e., we can select a certain group of the poselets representation and certain parameters of additional dependencies between symmetric limbs depending on the specific sequence, which enables more flexibility for the model to be simplified or added more constraints.

Chapter 7

Conclusions and Future Directions

7.1 Summary

This thesis has proposed a system for 2D human pose tracking on monocular videos. Monocular cameras are the most widely and easily available sources that record all kinds of human activities. In the past few decades, vision-based tracking using cameras rather than wearable sensors has received increasing attention because it is cheaper and more convenient.

Many algorithms have been proposed but building a robust human motion tracking framework is still a challenging task.

One of the challenges is the variations of body size in one video sequence. Videos in reality often contain people appearing at any distance to the camera hence appeared in various scales in the videos. To detect and track the human pose in each frame with a proper scale, a scale checking and adjusting module is developed in this thesis and incorporated into the tracking process. Chapter 4 presented this module. Two metrics are proposed and proved effective for detecting and adjusting the scale change within a sequence. The first metric is from the estimated height value of the tracked target (*Height_Metric*), which is suitable for some sequences where the tracked target has generally upright postures with no limbs stretching. The other metric is named *PixelCount_Metric* and implemented by computing the ratio between pixel counts of the foreground blobs and the estimated body part bounding boxes, which is invariant to motion types, thus is more generic.

Another challenge for 2D human pose tracking is resulted from cluttered background and double counting of body parts. The shapes similar to limbs in the image background and the similar appearances of symmetric body parts often affect the per-

formance of tracking frameworks based on the generic Pictorial Structures (PicStr) model. This is due to the fact that the part detectors in the generic PicStr model are trained based on shape feature only and the prior over body part connections is assumed to be a tree structure independent to image evidence. In order to address this challenge and improve the accuracy of the tracking framework, especially when dealing with rare and complex poses, the representation of body parts and the human body model utilized in the tracking framework is re-explored in this thesis.

In addition to the basic rigid body parts defined in the PicStr model, a series of more expressive spatial constraints are incorporated in this thesis by defining a mixture of mid-level spatial representations. Chapter 5 described these representations, named poselets. Each poselet captures multiple body parts configurations and dependencies, which is image conditioned and used to model higher order information among multiple body parts.

In Chapter 6, the human body model used for pose estimation is researched to further improve the tracking performance, especially for active body parts. Specifically, the generic tree structured PicStr model is augmented with more dependencies between symmetric and non-adjacent body parts, i.e., left and right upper/lower arms and legs. A new framework is proposed based on PicStr model to include additional constraints (AdCon) between symmetric limbs. These AdCons are in fact important factors for body balancing and coordination, which tend to force the left and right limbs separate and hence make the tracking to follow the motion biologically.

The confusion between similar-looking body parts, especially the left and right limbs, is also a challenge during human pose tracking. With the generic PicStr model, both sides of limbs are often confused especially when estimating side-faced poses due to the similarities of their appearance and the overlapping of these body parts. To overcome this problem and ensure the left and right limbs of being recognized correctly during tracking, a novel approach based on a simple head orientation (looking left or right) estimation is also proposed in Chapter 6, which serves as a complementary tool to assist the human pose estimation.

The complete tracking system can produce satisfactory tracking performance for video sequences with scale variations and a wide range of motion types, even rare and complex motions. It is highly effective in dealing with problems such as limbs

drifting due to cluttered background and double counting during the tracking process. Additionally, the proposed tracking system can distinguish the left and right limbs consistently during tracking process. The experimental results demonstrate that the proposed framework outperforms existing systems significantly, especially for the active body parts such as forearms and lower-legs.

7.2 Future Work

Based on the findings described in this thesis, some future directions could be possible and the components of our framework could be further improved.

One of the future directions is to explore more methods for handling the self-occlusion problem during tracking. Self-occlusion means that one body part is occluded/overlapped by another in the image, which often occurs in human pose tracking. The left/right confusion correction module proposed in this thesis works well for some self-occlusion cases because it can distinguish the visible body side and hence is able to deduce the depth order of body parts correctly. Future work will aim to learn more adaptable templates for determining head orientation and hence improve the accuracy for detecting the occlusion cases. We believe that other more effective methods for dealing with self-occlusion cases are deserved to be researched. For example, modelling the probability of an occlusion state of body parts may be a good way to implement an adaptive occlusion-sensitive model. In addition, dealing with inference for the occluded body parts is also an aspect to be further researched.

More robust and versatile higher level features and other approaches should also be explored to constrain more image observations into the basic human body model. In addition to the higher level features (poselets representations) used in this thesis, a variety of other cues based on body part correlations in poses could also be used to adapt the model with images. One limitation of the poselet representations is that they are independent from each other, which could limit their full functionality. Dependencies between poselets could be explored to establish the spatial correlation between poselets. Moreover, the factor graph structure used in this thesis enables the human body model being tunable, which means we can simplify or add more constraints between body parts by easily changing the factors. This excellent char-

acteristics suggests a possibility of future work on the human body model to research dependencies between different groups of non-connect body parts.

Furthermore, the proposed framework is currently only implemented for tracking of a single person. Future work could make some effort to track multiple persons within the same video sequence. This could be achieved by analysing the type mode of the appearance model. One possible method is to distinguish the appearance mode for torso, hence to distinguish between multiple people because torso is a large and consistently-detected body part. Once the torsos are defined, the scale of each person and the other body parts could be inferred by exploiting some motion constraints. The modules proposed in this thesis are also suitable if we make some adjustments depending on specific cases. For example, the head orientation module could provide not only information for body orientation but also each person's location even their motion direction. When a person mode is determined, the higher order dependencies among multiple body parts and constraints between symmetric parts can then be applied to infer the pose of each person.

Bibliography

Amit, Y. and Trouvé, A. (2007). Pop: Patchwork of parts models for object recognition. *International Journal of Computer Vision*, **75**(2), 267–282.

Andriluka, M., Roth, S., and Schiele, B. (2009). Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, pages 1014–1021. IEEE.

Andriluka, M., Roth, S., and Schiele, B. (2012). Discriminative appearance models for pictorial structures. *International journal of computer vision*, pages 1–22.

Bai, T. and Li, Y. (2012). Robust visual tracking with structured sparse representation appearance model. *Pattern recognition*, **45**(6), 2390–2404.

Bar-Shalom, Y. and Li, X.-R. (1993). Estimation and tracking- principles, techniques, and software. *Norwood, MA: Artech House, Inc, 1993.*

Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.

Barrón, C. and Kakadiaris, I. A. (2004). Monocular human motion tracking. *Multimedia systems*, **10**(2), 118–130.

Belongie, S., Malik, J., and Puzicha, J. (2000). Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, volume 2, page 3.

Binford, T. O. (1971). Visual perception by computer. In *IEEE conference on Systems and Control*, volume 261, page 262.

B.Leibe, A. L. and Schiele, B. (2007). Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*.

Bourdev, L., Maji, S., Brox, T., and Malik, J. (2010). Detecting people using mutually consistent poselet activations. In *ECCV*, pages 168–181. Springer.

Boykov, Y. and Funka-Lea, G. (2006). Graph cuts and efficient nd image segmentation. *International journal of computer vision*, **70**(2), 109–131.

Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **26**(9), 1124–1137.

Boykov, Y. Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE.

Bregler, C., Malik, J., and Pullen, K. (2004). Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, **56**(3), 179–194.

Brookner, E. (1998). Kalman filter. *Tracking and Kalman Filtering Made Easy*, pages 64–110.

Burl, M. C., Weber, M., and Perona, P. (1998). A probabilistic approach to object recognition using local photometry and global geometry. In *Computer VisionECCV98*, pages 628–641. Springer.

Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6), 679–698.

Clemmensen, L., Hastie, T., Witten, D., and Ersbøll, B. (2011). Sparse discriminant analysis. *Technometrics*, **53**(4), 406–413.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **24**(5), 603–619.

Crandall, D., Felzenszwalb, P., and Hutten, D. (2005). Spatial priors for part-based recognition using statistical models. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 10–17. IEEE.

Cucchiara, R., Grana, C., Piccardi, M., and Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **25**(10), 1337–1342.

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

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE.

Delamarre, Q. and Faugeras, O. (2001). 3d articulated models and multiview tracking with physical forces. *Computer Vision and Image Understanding*, **81**(3), 328–357.

Demirkus, M., Precup, D., Clark, J. J., and Arbel, T. (2014). Probabilistic temporal head pose estimation using a hierarchical graphical model. In *Computer Vision–ECCV 2014*, pages 328–344. Springer.

Deng, J., Berg, A., Satheesh, S., Su, H., Khosla, A., and Fei-Fei, L. (2012). Imagenet large scale visual recognition competition 2012 (ilsvrc2012).

Desai, C. and Ramanan, D. (2012). Detecting actions, poses, and objects with relational phraselets. *ECCV*, pages 158–172.

Deutscher, J., Blake, A., and Reid, I. (2000). Articulated body motion capture by annealed particle filtering. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 126–133. IEEE.

Drummond, T. and Cipolla, R. (2001). Real-time tracking of highly articulated structures in the presence of noisy measurements. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 315–320. IEEE.

Efe, M. and Bonvin, D. (2002). Data association in clutter with an adaptive filter. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, volume 2, pages 1243–1248. IEEE.

Eichner, M., Marin-Jimenez, M., Zisserman, A., and Ferrari, V. (2012). 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *International journal of computer vision*, **99**(2), 190–214.

Endres, I. and Hoiem, D. (2010). Category independent object proposals. In *Computer Vision–ECCV 2010*, pages 575–588. Springer.

Fablet, R. and Black, M. J. (2002). Automatic detection and tracking of human motion with a view-based representation. In *ECCV*, pages 476–491. Springer.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2000). Efficient matching of pictorial structures. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 66–73. IEEE.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, **61**(1), 55–79.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **32**(9), 1627–1645.

Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–264. IEEE.

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

Ferrari, V., Marín-Jiménez, M., and Zisserman, A. (2009a). 2d human pose estimation in tv shows. *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 128–147.

Ferrari, V., Marin-Jimenez, M., and Zisserman, A. (2009b). Pose search: retrieving people using their pose. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1–8. IEEE.

Fischler, M. A. and Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Transactions on computers*, (1), 67–92.

Forsyth, D. A. and Fleck, M. M. (1997). Body plans. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 678–683. IEEE.

Friedman, N. and Russell, S. (1997). Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 175–181. Morgan Kaufmann Publishers Inc.

Gavrila, D. and Davis, L. (1996). Tracking of humans in action: A 3d model-based approach. In *Proc. ARPA Image Understanding Workshop, Palm Springs*, pages 737–746. Citeseer.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Girshick, R. B., Felzenszwalb, P. F., and McAllester, D. (2012). Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/.

Ioffe, S. and Forsyth, D. A. (2001). Probabilistic methods for finding people. *International Journal of Computer Vision*, **43**(1), 45–68.

Isard, M. and Blake, A. (1998). Condensationconditional density propagation for visual tracking. *International journal of computer vision*, **29**(1), 5–28.

Johnson, S. and Everingham, M. (2009). Combining discriminative appearance and segmentation cues for articulated human pose estimation. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 405–412. IEEE.

Johnson, S. and Everingham, M. (2010). Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*. doi:10.5244/C.24.12.

KaewTraKulPong, P. and Bowden, R. (2002). An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-based surveillance systems*, pages 135–144. Springer.

Kaliamoorthi, P. and Kakarala, R. (2013). Parametric annealing: A stochastic search method for human pose tracking. *Pattern Recognition*, **46**(5), 1501 – 1510.

Karlinsky, L. and Ullman, S. (2012). Using linking features in learning non-parametric part models. In *ECCV*, pages 326–339. Springer.

Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B., and Russell, S. (1994). Towards robust automatic traffic scene analysis in real-time. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 126–131. IEEE.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, **47**(2), 498–519.

Lan, X. and Huttenlocher, D. P. (2005). Beyond trees: Common-factor models for 2d human pose recovery. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 470–477. IEEE.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, **60**(2), 91–110.

Lu, Y., Li, L., and Peursum, P. (2012a). Background suppression for building accurate appearance models in human motion tracking. *DICTA*.

Lu, Y., Li, L., and Peursum, P. (2012b). Human pose tracking based on both generic and specific appearance models. *ICARCV*.

Ma, B., Huang, R., and Qin, L. (2015). Vod: A novel image representation for head yaw estimation. *Neurocomputing*, **148**, 455–466.

McKenna, S. J., Jabri, S., Duric, Z., Rosenfeld, A., and Wechsler, H. (2000). Tracking groups of people. *Computer Vision and Image Understanding*, **80**(1), 42–56.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **27**(10), 1615–1630.

Mittal, A. and Davis, L. S. (2003). M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, **51**(3), 189–203.

Mooij, J. M. (2010). libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, **11**, 2169–2173.

Mori, G., Ren, X., Efros, A. A., and Malik, J. (2004). Recovering human body configurations: Combining segmentation and recognition. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–326. IEEE.

Murphy, K. P. (1998). Switching kalman filters. Technical report, Citeseer.

Neal, R. and Hinton, G. (1998). A new view of the em algorithm that justifies incremental, sparse and other variants.

Niyogi, S. A. and Adelson, E. H. (1994). Analyzing and recognizing walking figures in xyt. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 469–474. IEEE.

Okuma, K., Taleghani, A., De Freitas, N., Little, J. J., and Lowe, D. G. (2004). A boosted particle filter: Multitarget detection and tracking. In *ECCV*, pages 28–39. Springer.

Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., and Poggio, T. (1997). Pedestrian detection using wavelet templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 193–199. IEEE.

Pishchulin, L., Andriluka, M., Gehler, P., and Schiele, B. (2013). Poselet conditioned pictorial structures. In *CVPR*, pages 588–595. IEEE.

Ponce, J., Forsyth, D., Willow, E.-p., Antipolis-Méditerranée, S., d'activité RAweb, R., Inria, L., and Alumni, I. (2011). Computer vision: a modern approach. *Computer*, **16**, 11.

Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, **108**(1C2), 4 – 18.

Power, P. W. and Schoonees, J. A. (2002). Understanding background mixture models for foreground segmentation. In *Proceedings image and vision computing New Zealand*, volume 2002, pages 10–11.

Ramakrishna, V., Kanade, T., and Sheikh, Y. (2013). Tracking human pose by tracking symmetric parts. In *CVPR*, pages 3728–3735. IEEE.

Ramanan, D. (2006). Learning to parse images of articulated bodies. In *Advances in neural information processing systems*, pages 1129–1136.

Ramanan, D. (2007). Learning to parse images of articulated bodies. *Advances in Neural Information Processing Systems*, **19**, 1129.

Ramanan, D. and Forsyth, D. A. (2003). Finding and tracking people from the bottom up. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–467. IEEE.

Ramanan, D., Forsyth, D. A., and Zisserman, A. (2005). Strike a pose: Tracking people by finding stylized poses. In *CVPR*, volume 1, pages 271–278. IEEE.

Ramanan, D., Forsyth, D. A., and Zisserman, A. (2007). Tracking people by learning their appearance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **29**(1), 65–81.

Ridder, C., Munkelt, O., and Kirchner, H. (1995). Adaptive background estimation and foreground detection using kalman-filtering. In *Proceedings of International Conference on recent Advances in Mechatronics*, pages 193–199. Citeseer.

Ronfard, R., Schmid, C., and Triggs, B. (2002). Learning to parse pictures of people. In *Computer VisionECCV 2002*, pages 700–714. Springer.

Rother, C., Kolmogorov, V., and Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM.

Seki, M., Wada, T., Fujiwara, H., and Sumi, K. (2003). Background subtraction based on cooccurrence of image variations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–65. IEEE.

Sidenbladh, H., Black, M., and Fleet, D. (2000). Stochastic tracking of 3d human figures using 2d image motion. *ECCV*, pages 702–718.

Sidenbladh, H., Black, M. J., and Sigal, L. (2002). Implicit probabilistic models of human motion for synthesis and tracking. In *Computer VisionECCV 2002*, pages 784–800. Springer.

Sigal, L. and Black, M. J. (2006a). Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2041–2048. IEEE.

Sigal, L. and Black, M. J. (2006b). Predicting 3d people from 2d pictures. In *Articulated Motion and Deformable Objects*, pages 185–195. Springer.

Sigal, L., Isard, M., Sigelman, B. H., and Black, M. J. (2003). Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *Advances in neural information processing systems*, page None.

Sigal, L., Balan, A. O., and Black, M. J. (2010). Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, **87**(1), 4–27.

Song, Y., Feng, X., and Perona, P. (2000). Towards detection of human motion. In *CVPR*, volume 1, pages 810–817. IEEE.

Song, Y., Goncalves, L., and Perona, P. (2003). Unsupervised learning of human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **25**(7), 814–827.

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

Stauffer, C. and Grimson, W. E. L. (2000). Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **22**(8), 747–757.

Sullivan, J. and Carlsson, S. (2002). Recognizing and tracking human action. In *ECCV*, pages 629–644. Springer.

Tian, J., Li, L., and Liu, W. (2015). Monocular human motion tracking with non-connected body part dependency. In *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on*, pages 1–7. IEEE.

Tran, D. and Forsyth, D. (2010). Improved human parsing with a full relational model. In *Computer Vision–ECCV 2010*, pages 227–240. Springer.

Viola, P., Jones, M. J., and Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, **63**(2), 153–161.

Wachter, S. and Nagel, H.-H. (1997). Tracking of persons in monocular image sequences. In *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pages 2–9. IEEE.

Wang, Y., Tran, D., and Liao, Z. (2011). Learning hierarchical poselets for human parsing. In *CVPR*, pages 1705–1712. IEEE.

Weber, M., Welling, M., and Perona, P. (2000a). Towards automatic discovery of object categories. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 101–108. IEEE.

Weber, M., Welling, M., and Perona, P. (2000b). *Unsupervised learning of models for recognition*. Springer.

Wiklund, J. and Granlund, G. H. (1986). Tracking of multiple moving objects.

Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **19**(7), 780–785.

Zhou, H. and Hu, H. (2008). Human motion tracking for rehabilitationa survey. *Biomedical Signal Processing and Control*, **3**(1), 1 – 18.

Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.