Human pose estimation based on data-driven Monte Carlo hidden Markov models

Tao, Meng

2007

Tao, M. (2007). Human pose estimation based on data-driven Monte Carlo hidden Markov models. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/3416>

<https://doi.org/10.32657/10356/3416>

Nanyang Technological University

Downloaded on 26 Apr 2023 22:58:19 SGT

Human Pose Estimation Based on Data-Driven Monte Carlo Hidden Markov Models

Tao Meng

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Master of Engineering

2006

Acknowledgment

During my graduate study at Nanyang Technological University, I encountered a lot of difficulties but luckily I didn't walk alone through my life. Here I want to thank all the people who have helped me overcome the obstacles. Without them nothing would be possible.

First, I would like to thank my supervisor, Dr. Chua Chin Seng and my co-supervisor, Dr. Yau Wei Yun, for the guidance throughout the whole research project. Their way of thinking and invaluable suggestions inspired me to do a better job.

I also want to thank Institute for Infocomm Research, Singapore for offering me the research scholarship.

Special thanks to all the people in the Intelligent Machine Research Lab, including Li Jiang, Liu Xiao Hui, Gede, Tata and Yeo Yong Kiang, for their advice and assistance provided.

Finally I want to thank my parents and my friends. Their warm support and encouragement give me the faith to complete this project.

Contents

Acknowledgements	i
Contents	ii
Summary	v
List of Figures	vii
List of Tables	xi
1 Introduction	1
1.1 Background	1
1.2 Motivation	3
1.3 Difficulties	5
1.4 Objectives of the Research	11
1.5 Major Contributions	12
1.6 Outline of the Thesis	14
2 Literature Review	16

2.1	Review on Pose Representation Extraction	17
2.1.1	Train-based Method	17
2.1.2	Detection-based Method	19
2.2	Review on Model Fitting	22
2.2.1	Top-down Approach	23
2.2.2	Bottom-up approach	26
2.2.3	Integration of Top-down and Bottom-up	27
3	System Structure	31
3.1	Body-part Detection	32
3.2	Configuration Inference	34
4	Body-Part Detection	37
4.1	Head-shoulder Contour Matching	38
4.1.1	Face Detection	38
4.1.2	Learning Algorithm	46
4.1.3	Edge Detection	48
4.1.4	Gradient Descent Approach	54
4.1.5	Results for Locating Head and Shoulders	59
4.2	Limb Detection	61
4.2.1	Parallel Line Detection	62
4.2.2	Candidate Selection by Maximum Likelihood	63
4.2.3	Results for Locating Arms and Legs	66

5	Human Pose Inference	70
5.1	Data-Driven MCHMM in Human Pose Estimation	70
5.1.1	Hidden Markov Models	71
5.1.2	Density Approximation	74
5.1.3	Data-Driven MCHMM Algorithm	79
5.1.4	Computational Complexity Analysis	82
5.1.5	Implementation Issues	83
5.2	3D Human Model & Image Likelihood	86
6	Experimental Results	89
6.1	Image Data	89
6.2	3D Pose Result & Analysis	90
7	Conclusions and Recommendation	97
7.1	Conclusion	97
7.2	Future Work	100
	Bibliography	103

Summary

Estimating human poses from 2D images or video sequences can provide the moving trajectories of the body joints for the high level processing, human activity recognition, which is applicable in surveillance, human-computer interaction and clinical and sport analysis. The underlying concept of a human pose estimation system is to find the best hypothesis or human model that fits the representation of the body pose built from the observed image. Hence such a system normally consists of three parts,

1. human detection for finding the global location of the human inside the image;
2. human representation extraction;
3. model fitting.

All of them are critical to the performance of the pose estimation system. Many researches have been conducted for each part of the system in recent years but no optimal solutions were found. This thesis mainly focuses on the last two parts: human representation extraction and model fitting.

The main contribution of this thesis is the use of a new statistical formulation

called Monte Carlo hidden Markov model (MCHMM). Estimating the human pose in such a Bayesian framework is equivalent to computing the *posterior* distribution in a high dimensional space. The hidden Markov model (HMM) is employed since it provides an efficient Bayesian inference for estimating the *posterior* distribution. Another issue related to the pose estimation is the integration over a high-dimensional probability distribution. In such a case, analytic evaluation is computationally intractable. Therefore the Monte Carlo method is incorporated with the HMM for density approximations. The novelty of the MCHMM algorithm is not only the combination of the Monte Carlo and HMM but also the approximation techniques employed. The conventional Monte Carlo methods only use the sample-based approximation while the MCHMM algorithm uses both sample-based and tree-based approximations. As a result, the MCHMM algorithm is able to converge to the local maximum from random initializations.

Another contribution of this thesis is the extension of the MCHMM algorithm by integrating data-driven method. The original MCHMM algorithm provides a top-down approach for the model matching. It is incapable to guide the model searching process to concentrate more on the local maximum or hot spots of the solution space. Utilizing the data-driven (bottom-up) method is to compute the image likelihood as the weight for each sampled pose parameters. This approach drastically improves the efficiency of the Bayesian inference of the MCHMM algorithm and results in a better convergence.

List of Figures

1.1	The structure of a typical pose estimation System	6
3.1	The structure of proposed pose estimation system	31
4.1	Examples of an upright and 45° rotated rectangles	40
4.2	The rectangles in the first row represent edge features. The second and third rows shows line features and the last one shows center- surround features	41
4.3	(a)Integral image and (b)calculation scheme of the pixel sum for up- right rectangles	44
4.4	(a)Integral image and (b)calculation scheme of the pixel sum for 45° rotated rectangles	45
4.5	(a)Grey-level transition of an ideal edge (b)Grey-level transition of a practical edge	48
4.6	The $\rho\eta$ -diagram	50
4.7	Nonmaxima suppression in the $\rho\eta$ -diagram (a)The two virtual neigh- bors (b)The pixel is not an edge point (c)The pixel is an edge point .	51

4.8	(a)Decision boundary for nonmaxima suppression in $\rho\eta$ -diagram (b)Decision boundary for hysteresis thresholding	53
4.9	Noisy image: (a)Original image (b)Binary image by Canny detector (c)Binary image by the edge detector with confidence measure	54
4.10	Controlled room: (a)Original image (b)Binary image by Canny detector (c)Binary image by the edge detector with confidence measure	55
4.11	Image with grass patch: (a)Original image (b)Binary image by Canny detector (c)Binary image by the edge detector with confidence measure	55
4.12	Image with sandy ground: (a)Original image (b)Binary image by Canny detector (c)Binary image by the edge detector with confidence measure	56
4.13	Eight head-shoulder contour models	57
4.14	(a), (f), (k)&(p)Original image; (b), (g), (l)&(q)Face detection; (c), (h), (m)&(r)Binary edge image; (d), (i), (n)&(s)Edge-distance transformation map; (e), (j), (o)&(t)Head-shoulder contour matched . . .	61
4.15	(a)Original image; (b)Face detection; (c)Binary edge image; (d)Edge-distance transformation map; (e)Head-shoulder contour matched . . .	62
4.16	Head-shoulder sampling regions	62
4.17	(a)Straight lines by Hough transform; (b)Parallel-line sets; (c)Segmented image; (d)Candidate parallel-line sets for limbs; (e)Candidate sets represented by ellipse; (f)Limbs detected	67

4.18	Sampling region for (a)left elbow; (b)left wrist; (c)right elbow; (d)right wrist	68
4.19	Sampling region for (a)left knee; (b)left ankle; (c)right knee; (d)right ankle	69
5.1	The structure and grow process of the density tree. The domain of each node is marked by grey color.	77
5.2	A distribution with an isolated mode.	77
5.3	A data-driven MCHMM framework for human pose estimation. . . .	84
5.4	3D human model (a)front view; (b)profile view	86
5.5	Orientation in 3D space	87
5.6	Three angles defined for checking kinematic constraint	87
6.1	3D pose models estimated from 2D images. First row: original images; Second row: projection on 2D images; Third row: profile views .	91
6.2	Sample distribution over the width of the right-knee sampling region for the image in Figure 6.1a for (a)first MCHMM iteration; (b)first three MCHMM iterations; (c)first five MCHMM iterations; (d)first six MCHMM iterations; (e)first eight MCHMM iterations; (f) first ten MCHMM iterations. The horizontal axis indicates the x coordinates of the samples while the vertical axis indicates the normalized sample distribution.	94

6.3	Sample distribution over the height of the right-knee sampling region for the image in Figure 6.1a for (a)first MCHMM iteration; (b)first three MCHMM iterations; (c)first five MCHMM iterations; (d)first six MCHMM iterations; (e)first eight MCHMM iterations; (f) first ten MCHMM iterations. The horizontal axis indicates the y coordinates of the samples while the vertical axis indicates the normalized sample distribution.	95
6.4	Sample distribution over the 2D space of the right-knee sampling region for the image in Figure 6.1a for (a)first MCHMM iteration; (b)first three MCHMM iterations; (c)first five MCHMM iterations; (d)first six MCHMM iterations; (e)first eight MCHMM iterations; (f) first ten MCHMM iterations. The two horizontal axes indicates the x and y coordinates of the samples while the vertical axis indicates the normalized sample distribution.	96

List of Tables

4.1	Number of features inside of a 24×24 window for each prototype . . .	43
-----	---	----

Chapter 1

Introduction

1.1 Background

Increasing attention has been paid to the computer vision area since the late 1970s when computers are able to process large data sets such as images. The focus of this area is on the theory and technology for building artificial systems that can obtain information from images or videos and understand the information. The state of the art in computer vision can be characterized as immature and diverse. Although various computer vision systems have been proposed for different applications in the scope of process control, event detection, information organization, object modeling and interaction, there is no standard formulation or universal approach for solving computer vision problems. Instead existing systems are designed specifically for certain tasks such as autonomous vehicle, visual surveillance, medical image processing, and human-computer interaction, and seldom can be reused for other

tasks. Moreover, many applications are still in their infant stage and far from being actually applied in real life, though some commercial products based on computer vision technologies have already shown its effectiveness in reducing human work load and increasing productivity. For example, fingerprint recognition helps raise the security standard and ease crime investigation. Some inspection machines in the semi-conductor industry also employ computer vision technologies to check the pin connection and help save manpower. Such promising prospects of computer vision have encouraged people to explore the usage of the computer vision technologies in more and more application domains.

One significant part of computer vision is human activity recognition. Since human beings play a dominant role in the world, it is important for machines or computers to understand human activities that have occurred in images or videos. The term human activity not only refers to the actions performed by a single person like the whole body movement [32, 96], gait [90, 39] and arbitrary articulated motions [51, 50, 80] but also the interaction among people, i.e., multi-agent event [36, 88]. The driving force behind this is the wide spectrum of promising application areas, which can be categorized into three major aspects, surveillance, control and analysis.

The *surveillance* aspect involves applications where people are being tracked and monitored consistently over time so that certain actions can be predicted and recognized. Such applications can be used to build an ideal security system, which has the ability to understand the events which have occurred with the monitoring areas and send alarms automatically when unusual events happen. As a result,

people can be alleviated from watching long and boring security video streams. Except for event detection, the human gait is even used as biometrics for human identification [89] since it is non-invasive and effective at a distance.

Secondly, the *control* aspect covers applications where it provides interfaces between man and machines. For example, in industrial areas, human beings send different commands to robots by different poses.

Finally, the *analysis* aspect enables people to retrieve information from human activities. These applications can assist us in clinical studies and sports training. For example, if an activity recognition system is able to extract the movement of human body joints, then a medical doctor could use this system to monitor the recovery process of a leg-injured people. Also a football coach could use it to analyze the skills of a player. Moreover, a system for the multi-agent activity recognition could assist the football coach to understand the formation strategy his opponent team uses.

1.2 Motivation

This thesis focuses on analyzing human motions with the aim of providing low level information for the human activity recognition. According to the degree of deformation of body parts, the human motion is classified as an articulated motion, which has the following characteristics: individual rigid parts of an object move independently of one another and the motion of the whole object is non-rigid in nature [3]. Since the articulated motion results in a wide range of possible motion

transformations, it is nearly impossible to develop a general solution for interpreting all articulated structures of the human body. Some researchers try to solve this problem from an overall concept by using the binary shape of the human body. Based on the binary shape, several features can be extracted to model different human activities. These features include the evenly spaced points on the B-spline that approximates the shape boundary [28], temporal variations of the extracted binary shapes in nearby frames [35, 22], “star” skeletonization [27] and Hu-moment [11]. The models of human activities are then be learnt from the training features and used for recognition. However it is insufficient to describe the complex articulated human motion using an overall feature. The activity recognition systems built based on the above features are only capable of classifying specific or periodic motions, such as gait and are unable to be generalized over the recognition of other activities. Due to the 2D binary images used, they are not good at handling self-occlusion either. To overcome the limitation of the 2D images, some researchers attempt to use stereo images to obtain depth information, for example, the Spfinder (“stereo person finder”) system proposed in [8] and the work in [52, 23]. The stereo systems bring the camera calibration problem, which is not easy to be solved in a less-controlled environment such as a surveillance system. To analyze arbitrary human motions in 2D images, the ideal approach is to estimate the positions or trajectories of human body joints from images or video sequences. The underlying philosophy is based on the motion perception behavior of human beings found by Johansson [46]. In his experiment, a set of bright spots are attached to the body joints of an actor who is

dressed in black and moving in front of a dark background. People who are watching the actor can recognize different activities like walking, running, dancing, etc., based solely on the movement of the spots. Therefore it can be expected that computers can identify human activities if provided with the position or trajectories of human body joints. Early works that follow this approach employ markers such as glass bead reflectors [76] and LED [85] to indicate the positions of human body joints. However it is impractical for human beings to wear markers in daily life. Hence these works are only suitable for the applications under the well controlled environment like the clinical analysis or industrial control. The effort of analyzing the articulated configuration of the human body in normal conditions leads to the rise of a new research area, vision-based pose estimation. It aims to determine the human pose, i.e., the 2D or 3D positions of body joints from the normal images where no markers are attached to the human body. Over the past decade, a lot of researchers have devoted themselves to this area and some well-known pose estimation systems such as the Pfinder (“person finder”) system [91] are proposed.

1.3 Difficulties

To build a pose estimation system is not trivial. Generally speaking, such a system can be divided into three parts, human detection, pose representation extraction and model fitting as shown in Figure 1.1. The basic concept is as follows: (1) find the position of the human inside the image; (2) build a pose representation of the human detected; (3) find the best hypothesis or human model to fit the

Figure 1.1: The structure of a typical pose estimation System

representation. Each part forms a critical component for the system and none of them have been successfully solved.

The motivation of integrating the human detection techniques is to eliminate the effect of the background of the image as much as possible. In real life, an image always contains a complex content so that it will be tedious and difficult to estimate the human pose in the entire image. Thus the human detection is required to constrain the work in the certain portion of the image. However to find the human object inside the image is not straightforward [29]. For the moving human body, the current approach usually uses the background subtraction or motion analysis methods [62, 82]. The problem is that other moving objects like cars and animals are also detected. Therefore an object classification is necessary to determine which moving object(s) is(are) human. The classification can be achieved using some simple features such as the size of the foreground [59] with the assumption that only human objects and vehicles are present and the size of human objects is smaller than that of vehicles. The detection of the moving persons can be reinforced by the appearance model of the human body [18]. The appearance model is built based on

the moving object detected in the initial stage. To detect human objects in a more general case, some complex features like active shape and contour models [9, 94] are employed. However since the human shape and size vary dramatically with the human pose and camera view, these approaches are often application-dependent. For the static human objects or static images, the detection of the human body relies on a model or template matching basis. For example, the work in [69, 70] use wavelet or Haar-like features to define the properties of the human objects and detect people through machine learning techniques. This kind of approaches generally has no assumption or constraints for the application environment but their accuracy still needs to be improved. The main reason is that it is hard to extract robust features for the human body from the image. The focus of this thesis is not on the stage of human detection but rather on the following two stages.

After the position of human objects is located inside the images or video frames, the pose estimation system concentrates the work on a portion of the entire image. A pose representation is then built so that the likelihood of the body pose can be measured quantitatively. The pose representation can be extracted based on the motion trajectories of each body part [64], shapes [34], silhouettes [50] or user-defined features [81, 25]. This technique is named as the train-based pose representation, which normally contains trained or pre-determined models with the information of joint or body-part positions. Another approach for the pose representation is called the detection-based pose representation. The basic idea is to detect candidate positions of body joints based on a set of image cues, such as contour, edge and color.

2D or 3D human models are then generated given the locations of each body joint for the next stage of the pose estimation system. The strength and weakness for each approach will be discussed later. One common problem is that both approaches are based on low level image cues, which are noisy and weak. Therefore the performance of the pose representation is easily to be affected by the following factors.

- **Background:** Although the human detection stage has pruned most of the background portion, its influence still can not be ignored. Especially for detection-based methods, which normally relies on the edge detection, a complex background may introduce many trivial edges and decrease the detection accuracy.
- **Lighting:** For the pose representations built on the shapes, the changes of lighting may cause inaccurate foreground segmentation. Also the shadow of the human body will give abnormal shapes. Approaches like the detection of abrupt lighting change, Gaussian based segmentation and morphological operation are proposed to solve this problem but their reliability has to be further improved.
- **Clothing:** The cloth normally makes the extracted body shape inconsistent with their original size and shape. Moreover, the pleat on the cloth makes the outline of the body look jaggy and results in noisy edge detection.
- **Occlusion:** Both the environmental and self occlusions block the visual information of some parts of the body. The environmental occlusion can even

block the whole body. It is hard to retrieve the positions or trajectories of the invisible body parts. Eventually, the loss of track will lead to inaccurate pose estimation. In this case, video indexing methods are usually employed to find the appropriate moving trajectory of the invisible body parts.

Due to these difficulties, the pose estimation system is often built based on certain assumptions, such as neat background, constant lighting and tight clothing. This thesis is not going to provide a powerful solution that can handle all the problems mentioned above but rather it attempts to find a way that is able to build good pose representations when the image cues are less-noisy and strong.

Given the pose representations built in the previous stage, the following work is to find the best fit models. The models could be some trained models with body configuration information or could be some 2D or 3D human models constructed from the positions of the body joints detected. To find appropriate models, there are two usual strategies, top-down or bottom-up approaches. The first strategy, top-down approach, compares the pre-defined models with the pose representation and select the model with the highest likelihood. Early works employ template matching [60] and learning algorithms like hidden Markov Models (HMMs) [65] for this approach. However, since the articulated structure are capable of forming a number of different body configurations, the model matching strategy is insufficient to estimate all possible human poses and is only preferred in some limited applications like gait recognition. Other researcher choose the stochastic search algorithm such as particle filter [42, 24] to find the best model in the solution space. The

exploration is normally taken in an iterative procedure, where the search direction of the current iteration is determined by the models or hypothesis generated in the previous iteration. The problem associated with the top-down approach is that the solution space is usually high dimensional. This is partly due to the articulated body structure. In 2D space, a human body is normally treated as a stick figure, which consists of 14 main segments, including head, torso and limbs. These segments are linked by 14 joints. Suppose one degree of freedom is assigned to the torso and limbs to define the length of the segment whereas maximum two degrees of freedom are assigned to the 14 joints to indicate the direction of the rotation. As a result, a body structure may have 37 degrees of freedom in total. Therefore a robust search algorithm should be able to explore the entire space and converge to the best solution but such an approach will impose significant computation load on the system and in addition, the currently proposed algorithms only support local convergence. On the other hand, the bottom-up approach doesn't assume pre-defined models or follow the requirement of searching a high dimensional space. Instead it finds the best fitting model by grouping the detected body parts. Ideally, this approach is able to interpret all human configurations but it is highly dependent on the accuracy for the candidate body parts or joints detected. Moreover, too many false alarms also increases the difficulties of the grouping procedure and result in wrong body configurations. Details of the advantages and difficulties for both top-down and bottom-up approaches will be discussed later. In this thesis, the top-down and bottom-up approaches are integrated in order to build a system that possesses the

advantages of each approach. The work focuses on the problem of how to search the best model in a high dimensional space.

1.4 Objectives of the Research

As mentioned above, the pose estimation system described in this thesis mainly concentrates on the work of the last two stages, pose representation and model fitting. However, both stages have their own difficulties and none of them are trivial. This project does not aim to provide an ultimate solution that can solve all the problems. Instead, it offers a new framework for the human pose estimation problem from a different point of view. The objective of this thesis is to develop a pose estimation system that can recover 3D human configurations from static images. The system should satisfy the following requirements:

- self-start or initialize automatically;
- able to find arbitrary postures;
- no constraints on background and clothing.

To infer the human pose in a high dimensional space, a new Bayesian framework, called data-driven Monte Carlo hidden Markov models (MCHMMs), is proposed. The underlying concept is as follows:

1. Detect different body parts based on the image cues to find the candidate sampling region for each body joint.

2. Sample the image positions of the body joints and convert them to pose parameters. Build 3D human models given the pose parameters
3. Apply data-driven MCHMMs to find the best 3D model that fits the image observation.

1.5 Major Contributions

This thesis presents a new system for the human pose estimation. It employs a detection-based method for the pose representation and integrates the top-down and bottom-up methods to search the most likely 3D human model given the pose representation.

The main contribution of this thesis is the use of a new statistical formulation, Monte Carlo hidden Markov models. Estimating the human pose in such a Bayesian framework is equivalent to compute the *posterior* distribution in a high dimensional space. The hidden Markov models (HMMs) are employed since it provides an efficient Bayesian inference for estimating the *posterior* distribution. Another issue related to the pose estimation is the integration over a high-dimensional probability distribution. In such a case, analytic evaluation is computationally intractable. Therefore the Monte Carlo method is incorporated with the HMM for density approximations.

The novelty of the MCHMMs algorithm is not only the combination of the Monte Carlo and HMM but also the approximation techniques employed. The conven-

tional Monte Carlo methods only use the sample-based approximation, e.g., MCMC [5, 15, 16] and particle filter [7]. Since the number of samples is quite small compared with the size of the sampling space, the sample-based approximation can only provide a sparse representation of a complex distribution over a large space. In order to explore the solution space efficiently, the conventional sample-based methods draw the samples from a proposal distribution, which can be any form of distributions for the solution space. Due to the sparse sample set and weak convergence of the Bayesian inference, the proposal distribution needs to be wisely constructed beforehand to ensure good convergence. This is not an easy task in a high-dimensional space. In contrast, the MCHMMs algorithm compensates the drawback of the sample-based approximation by integrating the sample-based and tree-based representations for density approximation. The underlying concept is that even though the randomly drawn samples is unrepresentative, it still contains certain information about the relative likelihood of other unsampled space [49]. Thus, it is possible to transform the samples into a probability distribution through a density tree, which forms a recursive representation of a multivariate distribution. As a result, the MCHMMs are able to converge to the local maximum from random initializations.

Another contribution of this thesis is the extension of the MCHMMs algorithm by integrating the data-driven method. The original MCHMMs algorithm provides a top-down approach for the model matching. It is incapable of guiding the model searching process to concentrate more on the local maximum or hot spots of the

solution space. The data-driven (bottom-up) method is integrated by offering constrained sampling space through the detection of different body parts. Moreover, instead of assigning equal weight for each sample, the image likelihood is used as the weight for each set of sampled pose parameters. This approach drastically improves the efficiency of the Bayesian inference of the MCHMMs algorithm and results in a better convergence.

These contributions will be discussed in greater details in the succeeding chapters.

1.6 Outline of the Thesis

The outline for the rest of this thesis is as follows:

Chapter 2 reviews the works that have been proposed so far related to the human pose estimation system. The revision mainly concentrates on the second and third stages of the system, i.e., pose representation extraction and model fitting. Different approaches for each stage are investigated and their advantages and disadvantages are discussed.

Chapter 3 briefly describes the structure of the pose estimation system proposed in this thesis. Generally speaking, the system consists of two stages, body-part detection and pose inference. The underlying concepts are presented and the methods used in each stage are introduced.

Chapter 4 presents the details of the techniques employed in the body-part detection stage. Results for the head-shoulder detection and limb detection are shown

separately.

Chapter 5 presents the MCHMMs algorithm used for the pose inference. The integration with the data-driven method is described. Moreover, the framework for estimating the human pose based on this algorithm is given in details. Finally, the generation of the 3D human model and image likelihood measurement are presented.

Chapter 6 shows the results of the 3D human models estimated from the image observations. Both frontal and profile views of the 3D models are given to provide qualitative measurement. The convergence of the data-driven MCHMMs is also discussed based on the sample distributions extracted.

Finally, Chapter 7 presents the conclusion as well as the future directions of this work.

Chapter 2

Literature Review

A lot of works related to the human pose estimation in 2D images have been published. As mentioned above, most of them follow a similar approach: (1) locate the position of the human object inside the image; (2) build a representation of the body pose inside the image; (3) find the best hypothesis or the human model to fit the representation. Since this thesis mainly concentrates on the second and third stages, this chapter presents a literature survey on the existing works of the last two stages. In some works, the second stage may not be necessary for the pose estimation system. For example, in [89], the pose estimation is limited to calculate the joint-angle trajectories of the gait activity when people are walking parallel to the image plane. Due to the periodicity of the gait activity, a motion model can be constructed to describe the distribution of the positions for each joint at any time phase in a walking cycle. Thus based on the prediction of the global position by detecting the moving human object, the articulated movement is interpreted under

the Condensation framework [42] that finds the best human model for the image observations.

2.1 Review on Pose Representation Extraction

Based on the difference in the second stage, the existing approaches can be categorized into two groups, train-based and detection-based.

2.1.1 Train-based Method

The train-based methods extract features such as silhouettes [90], contours [33, 92, 93], stick figures [4] and Hu moments [11] from 2D images and build pose representations which can represent different human poses based on the extracted features. Initially, the train-based methods will learn models from the pose representations for training. The trained models are often integrated with the information of joint or body-part positions in order to derive human body configurations. Therefore, by finding the model that matches with the pose representation extracted, the corresponding human posture can be obtained. For example, the *Ghost* system proposed in [33] use the silhouettes to represent different human postures with the idea that each body part should corresponds to a hull vertex detected along the boundary. In [90], the silhouette is described by the coordinates of a set of points on the boundary with the origin allocated at its shape centroid. As a result, a simple pattern classification method is employed to identify different postures by measuring the distance between the extracted shape and the mean shape. Locating the joints along

the contours or boundary of the silhouette is a bit tedious. Instead, the work in [10] converts the binary image into stick figures via the medial axis transformation [79]. As a result, the human pose is represented by a set of rigid lines connected with articulated joints. The methods based on the contours or silhouettes build the pose representations in the spatial domain in the sense that the contours or silhouettes contain information about the joint positions or articulated structures. Bobik and Davis [11] on the other hand model the patterns of the human pose based on the temporal variation of the binary image. Two features, motion-energy images (MEI) and motion-history images (MHI) are extracted by cumulating a sequence of binary motion images to represent the human motion in video index. The train-based methods mentioned so far can't adapt to the deformability of the human body shape caused by non-rigid transformation. Hence it becomes difficult to find an optimal model that matches with the image observation. Recently, more complicated representations have emerged with the belief that they are able to simplify the computation and reach a high-level interpretation of human configuration. In [93], the binary human shape is represented triangulated polygons, which are decomposed into a set of connected deformable triangles [25]. The boundary of the polygon approximates that of the human shape and it actually consists of piecewise linear contours. This special scheme is able to model any deformable shapes and can provide good matching qualities. The train-based methods have the ability to construct accurate representation for in-class postures but they can't generalize out-class postures that are not dealt with. Thus the applications based on this method are often under

certain restrictions such as special clothing [10], neat background [24] and parallel movements with respect to camera plane [93].

2.1.2 Detection-based Method

On the other hand, the detection-based methods have the potential to represent all possible postures. The idea is to find candidate positions for different body joints or body parts based on image cues such as edge, color and shape. Subsequently, these detected candidates can be used to build 2D or 3D human configurations so that the image likelihood can be measured. A simple approach can refer to Fleck's work [26], which uses the skin color and edge information to find cylindrical regions, which are formed by finding two parallel edges. However the system is implemented with the purpose of retrieving the image content to determine whether naked people is presented. The requirement for the accuracy of the pose representation is not high. Ioffe [40] adopted a similar approach to estimate the postures of naked people. The detected cylindrical regions are possible to represent the human pose by grouping nine candidate segments. The problem is that the proposed system is unable to accurately locate the body segments since the detected segments may not represent the body parts precisely. In addition, the implementation is limited to the images with clean background and naked people present. Such an approach is definitely infeasible in a real world. In [64], the wavelet feature is employed for detecting three body parts, the head, the lower part of the right leg and the right foot. The training and localization of the body parts in an image sequence follows the system described

in [72]. However, the object models of the body parts for training have to be manually initialized by placing marked rectangles in the first frame. Moreover, the training for the lower body parts needs to be repeated for different people since they normally wear different color of clothes. A more advanced detection technique based on the color cue is done by Ramanan and Forsyth [74]. They use a rectangular edge template to detect the torso and limbs since the body parts are in cylindrical shapes. The basic idea is to convolute the image with the rectangular template and find the image patches that are roughly in rectangle shapes. The image patches overlapped with the moving objects by background subtraction are extracted. The patches with consistent appearance in nearby frames are detected as candidate human body parts. Once the human pose is derived from these candidates, the appearance models for the corresponding body parts are built and used for later detection of the possible body parts. The approach may fail when there are too many rectangular-shaped objects in the background or when the appearance of the human objects varies with the change of the orientation. To eliminate the false detection that may appear in the above approach, Ramanan [75] proposed an approach based on the template matching. In his approach, a 2D human model for the lateral-walking pose is defined and represented by a set of connected rectangles. Each rectangle corresponds to one body part. The human model is also convoluted with a sequence of images and the human object and its corresponding body parts are detected by finding the image patch with the maximum image likelihood. Finally the appearance model for each body part is built and used to detect the body parts in other frames.

The pose representation extracted based on this detection method is accurate and robust but it only works under the assumption that the lateral-walking pose will surely emerge in a long video sequence. Both the approaches mentioned above are able to provide accurate detection for the human body parts. However, the two approaches can only work with a enough long video sequence. To find the human pose representation in a single frame or static images, it is not easy to build the appearance model for each body part. Instead a set of candidate segments for different body parts or candidate positions for different body joints are detected by different techniques based on one image feature or on the integration of multiple image cues. For example, in [56], three detection techniques are employed to find the positions of head, hands and torso given the silhouette of the moving human. The head position is determined by matching a predefined contour model along the contour boundary of the extracted silhouette while the positions of both hands are detected along the boundary by extracting the peaks of convex curvature. The positions of the head and hands can be further refined based on the human body structure and tracking information. Finally, the torso position is retrieved by the medial axis transformation. To detect the body parts in static images, Hua [38] combines four image cues including appearance, edge, color and shape. In this approach, face detection is employed to locate the position of the head. The lower arms and upper legs are detected using the skin color [61]. On the other hand, the detection of the torso, upper arms and lower legs is not straight forward since they don't have salient image cues. Therefore, geometric positions of the head, lower arms

and upper legs are combined with the edge and segmented blob information to detect the rest body parts. The problem with this application is that it is limited to images where the skin of the lower arms and upper legs is visible. Lee [55, 57] proposed another approach that lifts away the clothing constraint. The arms are allowed to be covered with sleeves but the hands still need to be exposed. The skin color and clothing model are then used to generate hypothesis for the position of arms [37]. Since the skin of legs are usually not visible, the medial axis is employed to represent the legs with the assumption that they are seldom horizontal. Compared with the train-based pose representation methods, the detection-based methods can work under less restrictions. Normally, they can cope with complex background, different human orientation and clothes. However, due to the weakness of the low-level image cues employed, the detection accuracy still needs to be improved. Another problem is that the detection of different body parts is often not independent. The failure of one body-part detection may lead to the failure of other body parts. Hence the system built is vulnerable.

2.2 Review on Model Fitting

After obtained the pose representations, the human postures are estimated by finding the best model that fits with the pose representation. In general, these approaches can be divided into two categories, namely, *top-down* approach and *bottom-up* approach. The *top-down* approach is normally implemented together with the train-based pose representation while the *bottom-up* approach is implemented with

the detection-based method.

2.2.1 Top-down Approach

The top-down approach performs the pose estimation task based on a comparison between a predefined model and the pose representation given. The best fitting model can be found by the model matching or model searching methods. The first method often tries to optimize an objective function, which measures the feature correspondence between the object model and the image features extracted such as optical flow [12] and edges [47]. This method makes the estimation system be able to handle the self-occlusion since the models of self-occlusion configurations can be constructed offline. A lot of early works that follow this method have been reviewed by Aggarwal [2]. A model can be a set of simple equations [21] or a braided walking pattern [68] that describes the periodicity of gait activities, such as, walking, running and marching. It is quite obvious that these simple models are incapable to deal with more complex human activities. To make the pose estimation system independent of activities, a common approach is the use of kinematic models, which specify the translation, rotation and relative position for different body segments and joints. The kinematic model can be a 2D stick figure [17, 44]. The problem of using this model is that the changes of the viewing camera angle may greatly affect the extraction of the stick figure and subsequently affect the matching score. The use of a 3D volumetric human model [78, 87] won't bring such a problem. However, more parameters are required to describe the 3D model and the

computation load also increases during the matching process. Moreover, 3D models bring the kinematic singularities which could cause lost of tracking. Morris and Rehg [67] proposed a novel kinematic model, called *2-D Scaled Prismatic Model*, to minimize the occurrence of the singularities during the tracking process. Except the use of the kinematic model, some special models are proposed. In the *Ghost* system [33], silhouette models are built to specify the order of different body parts along the boundary of the silhouette and the relative distance between each body part. The philosophy behind this approach is that the order of body parts along the silhouette differs greatly among four different main postures (standing, sitting, crawling-bending and laying down) and any other posture is typically a variation of one of the main postures. As a result, a deterministic approach is applied to find the most similar posture for the observed silhouette and allocating them according to the order constraints and relative distances. However, since the body movement is continuous, the discrete models used above are unable to provide accurate estimation for the activities occurred in the transition regions between different models. To solve this problem, Agarwal and Triggs [1] first partitions the solution space into continuous regions. Each region contains a single model which gives information about the body pose parameters of a modified Scaled Prismatic Model. Finally, a linear autoregressive process is learnt to cover the whole space of each region. Similarly, Zhang *et al.* [93] built a set of shape models which are equivalent to connected piecewise linear contours. This special representation enables the deformability of the shape model and provides good matching results for the body pose. The built

shape models are learnt from certain human postures that appear during the human dynamic motion. The parameters that describe the shape model are able to be re-adjusted to model the human postures that lie in between those have been modeled. The solution space is explored by the method of Sequential Monte Carlo based on the prior distribution of shape models. However, even with the deformable models, the performance of the pose estimation is still constrained by learned priors. Due to the articulated structure of the human body, it is impossible to generalize a few fixed models to cover the whole solution space. To search all possible human configuration, Deutscher *et al.* [24] built a framework based on a kinematic chain consisting of 17 segments, among which the head, torso and limbs are represented by conical sections with elliptical cross-sections. A stochastic search algorithm, called annealed particle filter [48], was introduced to look for the global maximum in a high dimensional configuration space because of its ability to counter the distraction effect of many local extremes. However, using a kinematic model often requires the search for the best solution to be in a high dimensional space since the structure of human body has many degrees of freedom. Therefore, to simplify the search process some researchers impose additional models, such as motion models [71, 43] in order to prune the search space and reduce the computation cost. As discussed above, both the model matching and model searching strategies employed in the top-down approach have the difficulties to estimate all possible human postures due to the articulated body structure. Another common problem associated with the top-down approach is the requirement of a strong pose prior. The reason is that

the current pose state highly depends on the previous one. A bad initialization may lead to poor estimation or lost of tracking. Some work solve this problem by manual initialization [71], which is inapplicable in a real-life application. As a result, the top-down approach is only implemented for certain human activities or under certain restrictions.

2.2.2 Bottom-up approach

On the other hand, bottom-up approaches show the potential capability of significantly reducing search spaces and accurate self-initialization. A typical bottom-up approach would first detect the body parts and then assemble them into the best configuration. A grouping procedure has to be carefully designed since the detection accuracy is often not very high and the false alarms are relative big. Fleck *et al.* [26] used skin color and edge information to find cylindrical regions. A brute force grouping procedure was then employed to assemble these cylinders into human figures constrained by the geometry relationship between body parts. Obviously, such an approach is insufficient to deal with all possible configurations. Currently, a lot researchers prefer to estimate the human postures in a statistical framework. Ioffe [40] employed a pure sampling algorithm to find the appropriate body configuration, which consists of nine body parts. The nine-segment assemblies are built incrementally from single-segment ones. At each sampling stage, additional segments that correspond to one body part are sampled and combined with the existing assemblies. Resampling is applied to the new set of assemblies and those with high likelihood are

kept for the next sampling stage. Ramanan and Forsyth proposed a more robust grouping procedure in [74]. The grouping procedure based on video index is modeled in a Bayesian network. The Bayesian network models the appearance of each body part, the kinematic constraints and the motion constraints. Finally, estimating the human configuration is equivalent to find the maximum *a posteriori* (MAP) of a video sequence computed via dynamic programming [20]. In [75], a similar MAP estimation is adopted to find the human configuration in a single frame. Both the approaches mentioned above demonstrate their capabilities in identifying the correct configurations of multiple people in one frame and recovering when losing track, but since their approaches rely on the appearance models, the background should be different from the human object. A major concern associated with the bottom-up approach is the reliability of the body-part detectors. If the positions of body parts are incorrectly located, it is impossible for the grouping algorithm, no matter how robust it is, to produce correct clusters. Therefore in most cases, the capability of the bottom-up approach is hard to be realized since there may be so many candidate segments that the grouping procedure fails to cope with them.

2.2.3 Integration of Top-down and Bottom-up

From the above discussion, we can see that both top-down and bottom-up approaches have their own advantages and limitations. The state-of-the-art pose estimation technology integrates the top-down and bottom-up approaches in the sense that the estimation system could possess advantages of both approaches. A typ-

ical system that follows this approach consists of two stages, detection and pose inference. Some constraints such as kinematic and position constraints are imposed in the second stage in order to prune the search space. The detection stage builds the pose representation by detecting the candidate positions of human body parts in images. It provides a good means to achieve self-initialization, but different researches may select different methods, e.g., skin color detection [6], edge detection [41] and face detection [66]. After finding possible body parts, a solution space is often defined and a number of hypothesized postures which are normally 2D or 3D kinematic models can be generated by sampling the solution space. A search process is then performed to identify good configurations by comparing the generated hypothesized postures with the image observations. Due to the large degree of freedom associated with the articulated body structure, the solution space often has high dimensionality. As a result, the exact inference employed in the bottom-up approach may not be suitable, researchers prefer the approximation inference based on the sampling scheme. The pose inference sounds similar to the top-down approach proposed by [24]. The difference is that the detection stage provides analytic inference and hereby greatly prunes the search space. Recent works [56, 55, 38] show good results by estimating the posterior distribution in a Bayesian framework. Data-driven techniques are utilized to guide the dynamic search in the solution space and increase the search efficiency and convergence rate. The idea is inspired by the work in [95]. Particle filtering, also known as Condensation algorithm [42], is one of the popular search algorithms for visual tracking. It provides an effective way to

handle the multi-modal distribution. In [56], Lee *et al.* described a detailed particle filter framework in which the likelihood of the observed image is estimated given the human model. Body-part detection, which is based on the matching of the boundary of the foreground, provides analytic inference for updating state parameters and hence constrains the sampling process. This approach has the advantage of handling track initializations and self-occlusion. However, the tracker requires a simple background to accurately extract the foreground boundary. Lee and Cohen [55] proposed another human pose estimation system for static images in 2004. Instead of using background subtraction for the system initialization, four detection techniques including , face detection, head-shoulders contour matching, elliptical skin blob detection and ridge detection, were employed to find candidate positions of different body parts. The search algorithm they implemented is similar to the one mentioned in [56], called Data-Driven Markov Chain Monte Carlo (MCMC). The two tracking systems discussed above require the presence of the full human body without the environmental occlusion. Roberts *et al.* [77] proposed a pose estimation system, which is able to handle the occlusion caused by a cluttered environment. This is done by combining learnt likelihood ratios computed only from the parameterized, visible parts. The method for body part detection is quite different from previous work because the authors perceive that in real world scenes, edge responses are not discriminating enough so that a description taking account of colour or texture is necessary. The basic idea is to measure the dissimilarity between the appearance of the foreground and background of a transformed probabilistic region. Except the

single part likelihood, a likelihood for paired body parts with similar appearances and a model of inter-part distances are also learnt in order to encode the structure of foreground appearance and the spatial relationship of body parts. As it is impossible to generate a set of samples to cover the entire high dimensional solution space, the sampling processing in [56, 55] is taken in the sub-space sequentially. In other words, the pose parameters are inferred consecutively for each body part. In contrast, Hua *et al.* [38] propose a statistical framework called data-driven belief propagation which can carry out the inference tasks in parallel. In this approach, the human body pose is interpreted in a graph-based structure where the nodes represent the body parts and the edges encode constraints among them. Although the samples are still generated from sub-spaces, the inference of the pose parameters for one body part doesn't require to fix the pose parameter for the other body parts. In this way, the computation speed can be increased dramatically if multiple computers are available.

Chapter 3

System Structure

This thesis presents a new statistical framework for inferring 3D pose parameters of the human body from static images by integrating the top-down and bottom-up methods. In general, the proposed pose estimation system can be divided into two stages, body-part detection and configuration inference. The second stage is constrained by the kinematic structure of human. The system structure is depicted in Figure 3.1 below.

Figure 3.1: The structure of proposed pose estimation system

The overall approach relies on an assembly of detectors, which attempt to find

the candidate regions of different body joints from image cues including edges and segmented color blobs. By sampling different sets of the joint positions from the candidate regions, the corresponding 3D human models are built and are used for calculating the likelihood based on the image observation. To search for the best model, the Bayesian framework called the data driven Monte Carlo hidden Markov models (MCHMMs), employs both sample-based and tree-based techniques for density approximations. A brief introduction for each stage will be given in the following paragraphs.

3.1 Body-part Detection

One problem that has perplexed researchers for many years in the area for the vision-based human pose estimation is the ability to extract robust features to represent different human body parts. So far none of the feature representation methods are able to easily distinguish and detect human body parts from image background. One possible approach to detect different body parts is to combine several low-level features. In this project, we use three types of image cues, Haar-like feature, edges and segmented color blobs to find the positions of the head, shoulders and limbs. The reasons for using the three features are given in the following paragraphs. The detection of these body parts is achieved by two separate approaches, head-shoulder contour matching and limb detection.

For the head-shoulder contour matching approach, the Haar-like feature and edge are employed. The human head-shoulder contour is probably the most salient

feature for representing humans due to its unique structure. Intuitively, the edges extracted should contain the head-shoulder contour as long as the human head and shoulders have distinct boundary compared with the background. Hence they are employed for the detection of the head and shoulders. The basic concept is to use the gradient descent approach to match the trained contour models with the edges and search for the positions with minimum edge distance. The initial position for the search process is determined by a face detection technique based on the Haar-like feature, which is an extension of the wavelet features. Compared with the raw pixels, the Haar-like feature is more representative and can well encode the face structure. Another reason for using this feature is that it can be computed efficiently for different scale of images in constant time. Further explanations for using the Haar-like feature can be found in [58].

In contrast to the head-shoulder contour, the features that represent the limbs are less obvious. Although the skin color is a robust feature to detect the arms and legs, it is not applicable in situations where the limbs are covered with clothes. The limb-detection technique proposed here tries to lift away the dependency on the skin color. One common observation for both of the arms and legs is that they can be roughly modeled by cylinders. This suggests that in an ideal case each limb should be detected by finding two approximately parallel edges. Therefore the parallel-line detection based on the edges can be employed for the detection of limbs. The edges are first converted into straight lines through the Hough transform. A number of parallel-line pairs can then be found by grouping two lines with the similar slope.

Due to the spurious edges, a lot of the parallel-line pairs are redundant. To distinguish the true pairs from noise, the color blobs need to be included. Naturally it can be assumed that the limbs are only represented by one or a few complete blobs whereas the parallel-line pairs of the background usually consist of partial blobs. Therefore the image likelihood of the limbs measured based on the color blobs are bigger than other pairs. To further distinguish the limbs from the background, additional geometric information is employed to measure the likelihood. The implementation details for the two detection approaches will be given in the next chapter.

3.2 Configuration Inference

The body-part detection in the first stage gives us the candidate locations for the body joints including the head, shoulders, elbows, wrists, knees and ankles. However, these candidate locations are still not enough to figure out the exact posture. Instead the candidate locations actually define a high dimensional space in which each point corresponds to one human configuration. The point is represented by a vector of pose parameters which specify the 2D coordinates of every body joint in the image and the size of every body part. Given the geometry information, the 3D positions for all joints are derived by kinematic transformation and a 3D human configuration is built subsequently. Eventually, the likelihood for the person in that configuration or pose is measured by the image likelihood obtained by projecting the 3D model to the 2D image. With the likelihood for all possible human

configurations, the high dimensional space can be regarded as a probability density function. Normally the probability density here is a multi-mode distribution. As a result, the pose estimation problem is formulated as estimating the density function over the solution space and finding the peak mode or local maximum. The Bayesian framework, data-driven MCHMMs, is employed for the density estimation. The hidden Markov model involved is efficient in inferring the posterior distribution but it is designed for model learning problem. As there are no training examples for the pose estimation, Monte Carlo methods are combined with the HMMs so that it is able to perform model search tasks in a high dimensional space. The integrated Monte Carlo method provides an approximation for the probability density. A typical Monte Carlo based Bayesian inference such as the Markov Chain Monte Carlo (MCMC), uses samples for the density approximation. The samples are generated from the proposal distribution which is usually represented by a mixture of distributions, such as Gaussian distributions. Although it is claimed that the proposal distribution will not affect the final convergence except the rate of convergence [30], in the high dimensional space a poor proposal distribution will normally lead to ill-fitted postures. To ensure convergence, the proposal distribution has to be designed to closely approximate the stationary probability distribution. This often requires a careful selection of the parameters that describe the mixture of distributions. In the detection-based pose estimation system, the parameters are inferred analytically from the body parts detected. On the other hand, the MCHMMs algorithm tries to generate samples from the uniform distribution to lift away the necessity

of constructing a dedicated proposal distribution. But sampling from the uniform distribution will result in a sparse representation in the large space. To overcome the drawback of the sample-based approximation method, the MCHMMs integrates the sample-based and tree-based representation for the density approximation. The tree-based method utilizes the generated samples to approximate the probability distribution of the solution space. The distributions of the unsampled regions are determined by the relative likelihood of the existing samples. To build a density tree for the density approximation, the whole solution space is firstly divided into subspaces. Each subspace maps to a single leaf in the tree. The distribution within each leaf is uniform while the distributions between different leaves depends on the weight of each leaf and its parent node. With the density approximation technique mentioned above, the MCHMMs is able to learn the stationary distribution over the high-dimensional space through multiple iterations of Expectation and Maximization (EM). The data-driven method is implemented to estimate the probability distributions more efficiently. An annealing process is also employed to avoid the overfitting problem when learning the HMMs. To improve the computational efficiency and convergence, the search space is pruned by kinematic constraints. By kinematic constraints, it refers to the information about impossible configurations for human beings, such as the configuration where hands penetrate into the torso. Details of the configuration inference stage and the kinematic constraints will be given later.

Chapter 4

Body-Part Detection

The purpose of integrating the body-part detection into the pose estimation system is to provide analytic inference for the model searching stage. As there are no obvious features representing all human body parts, multiple low-level image cues are fused for the detection task. Here three image cues including the Haar-like features, edges and segmented blobs are employed to detect the head, left and right shoulders, and four limbs. Based on the body parts detected, the sampling regions for the associated joints can then be constructed. Two detection techniques, namely head-shoulder contour matching and limb detection, are implemented to find the positions of the head and shoulders and the positions of the limbs respectively. To ensure the effectiveness of the two detection techniques, the minimum human size should be 150x50. Because otherwise, the face structure may not be obvious and the edges of limbs may be noisy.

4.1 Head-shoulder Contour Matching

The human head-shoulder contour is quite obvious in a binary edge image due to its unique structure. Hence a head-shoulder contour matching method is employed to locate the positions of head and shoulders. The idea is adopted from the contour matching procedure proposed in [54]. The basic concept is to align the exemplar contours to the edges extracted and select the one with minimum matching error. Each exemplar contour can be transformed into a number of different contours by specifying different translation parameters, rotation angles and scaling coefficients. As a result, each contour is modeled by five parameters, the number of the exemplar contour, translation in x and y directions, rotation angles and scaling coefficients. The gradient descent approach is chosen to search for the best model parameters. Since the gradient descent approach tends to be easily trapped in a local maximum, a face detection method based on the Haar-like features is employed in order to provide a good initialization of the transformation parameters. With the discussion above, it can be seen that the head-shoulder contour matching approach actually employs two detection methods, face detection and edge detection. A brief description for each detection method will be given followed by the gradient descent approach and the detection result.

4.1.1 Face Detection

The human face is probably one of the most unique symbols for representing human beings in 2D images. Its structural appearance offers the possibility of detecting

the face based on the feature extraction method. Generally speaking, two main problems need to be considered to build a face detection framework. One is the ability to represent the human face based on image information. Another one is the ability to distinguish them from other objects based on the known representations. To handle the two problems, we employ a rapid object detection scheme [58], which uses an extended set of Haar-like features for the face representation and a boosted multi-stage classifier for the classification.

The reason for using features instead of raw pixel values as the input to a learning algorithm is to reduce the intra-class variability while increasing the inter-class variability and thus making the classification easier. The Haar-like features are originated from the quadruple Haar wavelet features used by Papageorgiou *et al.* [70].

The basic unit of the feature pool is the sum of pixels located in an upright rectangle or 45° rotated rectangle. Assume an object can be covered by a fixed-size window of $W \times H$ pixels. A rectangle is then expressed as $r = (x, y, w, h, \alpha)$ with $0 \leq x, x+w \leq W$, $0 \leq y, y+h \leq H$, $x, y \geq 0$, $w, h > 0$, and $\alpha \in \{0^\circ, 45^\circ\}$ and its pixel sum is denoted by $RecSum(r)$. Fig 4.1 shows the sample for each rectangle. Note that the w and h indicate the number of columns and rows of pixels inside the rectangle respectively but not the width and height of the rectangle.

The raw feature set is the set of all possible features defined by the equation:

$$feature_I = \sum_{i \in I\{1, \dots, N\}} \omega_i \cdot RecSum(r_i) \quad (4.1)$$

Figure 4.1: Examples of an upright and 45° rotated rectangles

where the weights $\omega_i \in \mathfrak{R}$, the rectangles r_i , and N are arbitrarily chosen.

According to Eq 4.1, the feature set could be infinitely large and not all of the features are meaningful. Therefore, it is reduced as follows:

1. The features are calculated based on the pixel sums of two rectangles, i.e., $N=2$.
2. In case the two rectangles have different area size, we set $-\omega_1 \cdot \text{Area}(r_1) = \omega_2 \cdot \text{Area}(r_2)$.
3. The features mimic Haar-like features and early features of human visual pathway such as center-surround and directional responses.

With reference to these criteria, 14 feature prototypes are proposed as shown in Fig 4.2, including four edge features, eight line features, and two center-surround features.

There is no constraint on w and h as long as each of them equals to the multiples of the number of columns or rows shown in every feature. Thus, these prototypes are

Figure 4.2: The rectangles in the first row represent edge features. The second and third rows shows line features and the last one shows center-surround features

scaled independently in vertical and horizontal direction in order to generate a rich, over-complete set of features. For example, Fig 4.2(a) has two columns but one row meaning that w should be the multiples of 2, i.e., $w = 2 \times n$, with $0 \leq x+w \leq W$ and h can be any integer number as long as the rectangle lies within the image window. The smallest edge feature for Fig 4.2(a) has two pixels in width and one pixel in height. Note that the line features actually have three rectangles but we can still perceive the features are calculated by two rectangles only. In this case, it is assumed that the first rectangle r_1 encompasses the black and white rectangles while the second rectangle r_2 only contains the black area. For instance, line feature (a) with total rows of 2 and columns of 6 at the top left corner (5,3) can be written

as

$$feature_I = -1 \cdot RecSum(5, 3, 6, 2, 0^0) + 3 \cdot RecSum(7, 3, 2, 2, 0^0) \quad (4.2)$$

The number of features for each prototype is quite large and differs from prototype to prototype. The following paragraph shows the procedure for calculating the number of features for an upright feature. Let $X=[W/w]$ and $Y=[H/h]$ be the maximum scaling factors in x and y direction. Each sub-window with the size of $iw \times jh$ has a unique feature, with $1 \leq i \leq X$ and $1 \leq j \leq Y$. Therefore, along the x direction the number of sub-window is $\sum_{i=1}^X (W - i \cdot w + 1)$ and along the y direction it is $\sum_{j=1}^Y (H - j \cdot h + 1)$. Since these two numbers are independent, an upright feature of size $w \times h$ then generates

$$\left(\sum_{i=1}^X (W - i \cdot w + 1) \right) \cdot \left(\sum_{j=1}^Y (H - j \cdot h + 1) \right) \quad (4.3)$$

By simplifying the above equation, we obtain

$$XY \cdot \left(W + 1 - w \frac{X + 1}{2} \right) \cdot \left(H + 1 - h \frac{Y + 1}{2} \right) \quad (4.4)$$

features for an image of size $W \times H$.

The process to calculate the number of 45° rotated features is similar to the upright features. From Fig 4.1, we observe that a 45° rotated feature occupies a $(w + h) \times (w + h)$ square. Hence, by defining $z = w + h$, The expression for

calculating the number of 45° rotated features is

$$XY \cdot (W + 1 - z \frac{X + 1}{2}) \cdot (H + 1 - z \frac{Y + 1}{2}) \quad (4.5)$$

where $X = \lfloor W/z \rfloor$ and $Y = \lfloor H/z \rfloor$.

Table 4.1 lists the number of features for a window size of 24×24 .

Feature Type	w/h	X/Y	#
Ea; Eb	2/1; 1/2	12/24; 24/12	43,200
Ec; Ed	2/1; 1/2	8/8	8,464
La; Lc	3/1; 1/3	8/24; 24/8	27,600
Lb; Ld	4/1; 1/4	6/24; 24/6	20,736
Le; Lg	3/1; 1/3	6/6	4,356
Lf; Lh	4/1; 1/4	4/4	3,600
Ca	3/3	8/8	8,464
Cb	3/3	3/3	1,521
Total			117,941

Table 4.1: Number of features inside of a 24×24 window for each prototype

Since the number of features are quite large, it is inefficient to calculate the *RecSum* one by one. Thus an intermediate representation for the image, called integral image, is used to save the computation cost. The integral image for an upright rectangle at (x, y) is defined as the *Summed Area Table* $SAT(x, y)$, which is the sum of pixels of the upright rectangle ranging from the top left corner at $(0, 0)$ to the bottom right corner at (x, y) (see Fig 4.3(a)) , i.e.,

$$SAT(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \quad (4.6)$$

where $I(x, y)$ is the pixel value of original image.

Figure 4.3: (a)Integral image and (b)calculation scheme of the pixel sum for upright rectangles

It can be calculated with one pass over all pixels from left to right and top to bottom by means of

$$SAT(x, y) = SAT(x, y - 1) + SAT(x - 1, y) + I(x, y) - SAT(x - 1, y - 1) \quad (4.7)$$

with $SAT(-1, y) = SAT(x, -1) = 0$. Based on Eq 4.7, the pixel sum of any upright rectangle $r = (x, y, w, h, 0)$ can be determined by four table lookups as shown in Fig 4.3(b):

$$\begin{aligned} RecSum(r) = & SAT(x - 1, y - 1) + SAT(x + w - 1, y + h - 1) \\ & - SAT(x - 1, y + h - 1) - SAT(x + w - 1, y - 1) \end{aligned} \quad (4.8)$$

For 45° rotated rectangles the integral image is denoted as *Rotated Summed Area Table* $RSAT(x, y)$. Fig 4.4(a) describes the sum of the pixels of the rectangle rotated by 45° with the right most corner at (x, y) and extending till the boundaries of the

image. Mathematically, it can be expressed as:

$$RSAT(x, y) = \sum_{x' \leq x, x' \leq x - |y - y'|} I(x', y') \quad (4.9)$$

Figure 4.4: (a) Integral image and (b) calculation scheme of the pixel sum for 45° rotated rectangles

Similar to the procedure of calculating the pixel sum of upright rectangle, the $RSAT(x, y)$ is estimated with two passes. The first pass from left to right and top to bottom determines

$$RSAT(x, y) = RSAT(x-1, y-1) + RSAT(x-1, y) + I(x, y) - RSAT(x-2, y-1) \quad (4.10)$$

with $RSAT(-1, y) = RSAT(-2, y) = RSAT(x, -1) = 0$, whereas the second pass from the right to left and bottom to top calculates

$$RSAT(x, y) = RSAT(x, y) + RSAT(x-1, y+1) - RSAT(x-2, y) \quad (4.11)$$

From this the pixel sum of any rotated rectangle $r=(x,y,w,h,45^\circ)$ can also be determined by four table lookups as shown in Fig 3.4(b):

$$\begin{aligned} RecSum(r) = & RSAT(x+w-1, y+w-1) + RSAT(x-h-1, y+h-1) \\ & - RSAT(x-1, y-1) \\ & - RSAT(x+w-1-h, y+w-1+h) \end{aligned} \quad (4.12)$$

4.1.2 Learning Algorithm

With such a large feature pool, the difficulty appears in the way of obtaining a classifier that is robust and computationally efficient. Based on the assumption that a very small number of these features can be combined to form an effective classifier, Viola and Jones [86] proposed a boosted cascade classifier to find these features. The underlying philosophy of a cascade of classifiers is that at each stage a classifier is able to detect almost all objects of interest while rejecting a certain fraction of the non-object patterns. Suppose each stage is trained to eliminate 50% of the non-object patterns while only falsely eliminate 0.2% of the object patterns. After 13 stages, a false alarm rate of about $0.5^{13} \approx 1.2e-4$ and a detection rate of about $0.998^{13} \approx 0.97$ can be expected. AdaBoost algorithm is used for boosting the performance of the cascade of classifiers. It is a powerful machine learning algorithm, which can learn a strong classifier by weighted combination of a set of weak classifiers. The set of weak classifiers or base classifiers are all classifiers which perform a simple binary thresholding decision based on one feature from the feature

pool:

$$h_j = \begin{cases} 1 & \text{when if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \quad (4.13)$$

where $h_j(x)$ represents a weak classifier, which consists of a feature f_j , a threshold θ_j and a polarity p_j indicating the direction of the inequality sign. Details of the boosting process can be found in [86].

In order to detect the human face at multiple scales, the scaling process is applied at the detector rather than the image. This process makes sense because the features can be evaluated at any scale with the same cost. The detector is also shifted across the image to find the human body at different locations. At each time, the base window is shifted by some number of pixels Δ . The distance of movement relates to the scale of the detector, i.e., the size of the window. If the current scale is s the window is shifted by $[s\Delta]$, where $[]$ is the rounding operation. The detection algorithm was implemented using the existing Haar cascade classifier in OpenCV library. Its detection performance has been widely tested and it is able to detect the multi-scale faces in both frontal and profile views accurately.

The human localizer gives a good initialization for obtaining the locations of different body parts in the sense that it provides the region of interest for the body-part detectors to search rather than search in the whole image. The body-part detectors employ three detection techniques including edge detection, head-shoulder contour matching and skin color detection to find candidate locations for head,

shoulder and limbs.

4.1.3 Edge Detection

Background

Edge detection is employed to calculate the edge distance of each contour so that the matching quality can be measured. It is arguably the most important operation in low-level image processing. The reason is that it significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. Generally speaking, the edges in images are defined as the areas with meaningful gray-level transitions [31]. An ideal edge is a set of connected pixels, each of which is located at an orthogonal step transition in gray levels as show in Fig 4.5(a). However, in practice, optics, sampling, and other image acquisition imperfections yield edges that are blurred. As a result, the transitions are more likely to be a ramp like as depicted in Fig 4.5(b).

Figure 4.5: (a) Grey-level transition of an ideal edge (b) Grey-level transition of a practical edge

Plenty of techniques have been published to perform edge detection [13, 53]. The

majority of them may be grouped into two categories: gradient and Laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. On the other hand, The Laplacian method searches for zero crossings in the second derivative of the image to find edges. In this project, we employ a gradient-based edge detector integrated with confidence measures [63]. The detection procedure is quite similar to a famous edge detection algorithm, called Canny edge detector [14], which consists of three steps

1. Estimation of the gradient vector. The value of the gradient magnitude $\hat{g}$ and orientation $\hat{\theta}$ is obtained by using two differentiation masks.
2. Nonmaxima suppression. This step is to trace along the edge in the edge direction and suppress any pixel value (set it equal to 0) that is not considered to be an edge.
3. Hysteresis thresholding. It is a means of eliminating streaking, which appears in an edge when the output of the mask fluctuates above and below the threshold.

The detection procedure described above only employs the gradient magnitude information. To further enhance the detection performance, a confidence measure is incorporated with all three steps to measure the similarity between the data pattern and an ideal edge template using information not employed in the gradient magnitude. The details of the gradient estimation and confidence measurement are given in the appendix. Later section will focus on the integration with the confidence

measure for the nonmaxima suppression and hysteresis thresholding.

Integration with Confidence Measures

The confidence measurement will result in a $\rho\eta$ -diagram as shown in Fig 4.6. Every pixel of an image can be converted to a point in this diagram. Let $f(\rho, \eta) = 0$ be the implicit equation of a curve in the ρ - η plane. Every point (ρ_o, η_o) is then associated with a value $f(\rho_o, \eta_o)$, which is called the *algebraic distance* of the point from the curve. As depicted in Fig 4.6, all the points “inside” have negative algebraic distances, whereas all the points “outside” have positive algebraic distances. Similar

Figure 4.6: The $\rho\eta$ -diagram

to the Canny edge detector, it is possible to define the nonmaxima suppression and hysteresis thresholding in the context of the $\rho\eta$ -diagram.

The implementation of nonmaxima suppression is based on the sign of the algebraic distance. Firstly, two virtual neighbors Q_1 and Q_2 are generated given the estimated gradient orientation (see Fig 4.7(a)). The ρ and η values of the two virtual neighbors are then determined by linear interpolation from the existing ρ and η values of P_{12} , P_{13} and P_{31} , P_{32} , respectively. The prototype curve $f^{(x)}(\rho_o, \eta_o) = 0$

is inflated to pass through $P(\rho_o, \eta_o)$, defining the decision boundary

$$f^{(x)}(\rho, \eta) - f^{(x)}(\rho_o, \eta_o) = 0. \quad (4.14)$$

The pixel is classified as an edge point only when both virtual neighbors have negative algebraic distances as depicted in Fig 4.7(c).

Figure 4.7: Nonmaxima suppression in the $\rho\eta$ -diagram (a)The two virtual neighbors (b)The pixel is not an edge point (c)The pixel is an edge point

To perform hysteresis thresholding two decision boundaries $f^{(L)}(\rho, \eta) = 0$ and $f^{(H)}(\rho, \eta) = 0$ are defined in the $\rho\eta$ -diagram. They divide the $\rho\eta$ plane into three regions. The pixel (ρ_o, η_o) is then determined by hysteresis thresholding as follows

- if $f^{(L)}(\rho_o, \eta_o) > 0$ and $f^{(H)}(\rho_o, \eta_o) \geq 0$ retain for edge map
- if $f^{(L)}(\rho_o, \eta_o) \cdot f^{(H)}(\rho_o, \eta_o) < 0$ retain if neighbor in edge map
- if $f^{(L)}(\rho_o, \eta_o) \leq 0$ and $f^{(H)}(\rho_o, \eta_o) < 0$ discard

The second condition is being applied recursively.

The decision boundaries for both nonmaxima suppression and hysteresis thresh-

olding can have arbitrary shapes. Moreover, the two decision boundaries for hysteresis thresholding can even intersect with each other if necessary. Indeed, if all the decision boundaries are vertical lines

$$f^{(\text{L})}(\rho, \eta) = \rho_l \quad f^{(\text{H})}(\rho, \eta) = \rho_h \quad f^{(\text{X})}(\rho, \eta) = \rho \quad (4.15)$$

both nonmaxima suppression and hysteresis thresholding will be exclusively based on gradient magnitude and, thus, the new method defaults into the Canny edge detection approach.

Comparison with Canny edge detection

As mentioned above, the edge detection algorithm is actually an extension of the Canny edge detector. So similar to the Canny edge detector, the detection performance still depends on the thresholds chosen initially. In this approach, the thresholds are represented by the decision boundaries used in the $\rho\eta$ -diagram. The following options are available:

- horizontal or vertical line;
- A corner of a rectangle, which consists of a horizontal and a vertical line;
- The first quadrant of an ellipse with the center in the origin;
- arbitrary polygonal curve.

Here an ellipse which passes through the points (0.5, 0) and (0, 0.5) in the $\rho\eta$ -diagram is chosen for the decision boundary of nonmaxima suppression (see Fig

4.8(a)). For hysteresis thresholding, the lower decision boundary is an ellipse passes through $(0.99, 0)$ and $(0, 0.91)$ while the higher boundary is a corner of a rectangle located at $(0.93, 0)$ and $(0, 0.96)$ as shown in Fig 4.8(b).

Figure 4.8: (a)Decision boundary for nonmaxima suppression in $\rho\eta$ -diagram
(b)Decision boundary for hysteresis thresholding

Although the performance for both edge detectors are subjective, generally speaking the edge detection algorithm we selected outperforms the Canny edge detector. Comparison between these two approaches are given in both Fig 4.9 and 4.10. The thresholds for the Canny detector have been adjusted to achieve the best performance for individual images. As we can see that in noisy images, the Canny edge detector (see Fig 4.9(b)) gives too many details when the clothes have many pleats. On the other hand, the edge detector with the confidence measure (see Fig 4.9(c)) shows more structural information and eliminate those trivial edges. For the head-shoulder contour matching, the trivial edges is useless and may even degrade the performance of localizing head and shoulders. However, in a well-controlled environment, for example a room with clean background and constant illumination, the

performance of the Canny edge detector is comparable with the detector selected. The selected detector also has better performance in dealing with high texture images as shown in Figure 4.11 and 4.12. The grass patch in Figure 4.11(a) is a high texture image portion. It introduces a lot of trivial edges for the Canny edge detector while the high texture effect is greatly eliminated by the selected detector. Moreover, it can be observed that the tree on top of the head also brings a lot of noise via the Canny edge detector. The noisy edges may probably lead to poor matching quality for the head-shoulder contours. In contrast, the selected detector gives more structural information for the tree and better matching quality can be expected. Similarly, the sandy ground in Figure 4.12(a) is also well handled by the selected detector.

Figure 4.9: Noisy image: (a)Original image (b)Binary image by Canny detector (c)Binary image by the edge detector with confidence measure

4.1.4 Gradient Descent Approach

Given the initial search position and the binary edge image, a gradient descent approach is used to match the contour models with the edges and search for the

Figure 4.10: Controlled room: (a)Original image (b)Binary image by Canny detector (c)Binary image by the edge detector with confidence measure

Figure 4.11: Image with grass patch: (a)Original image (b)Binary image by Canny detector (c)Binary image by the edge detector with confidence measure

positions with minimum edge distance. The details of this algorithm are shown below.

Totally eight contour models are used in this approach as shown in Fig 4.13. The models are the same as the ones used in [54]. They are learnt from a set of 100 training data by using the K-means clustering algorithm and have different shapes in order to sufficiently cover all possible head-shoulder contours. Each of them is represented by a set of connected points $\{u_i, v_i\}$ in 2D. The coordinate values

Figure 4.12: Image with sandy ground: (a)Original image (b)Binary image by Canny detector (c)Binary image by the edge detector with confidence measure

of u and v are normalized into floating values rather than using integer values to represent the position of each point. The image position of head and shoulders are also defined for every trained model.

Based on the transformation equations below, the normalized coordinates $\{u_i, v_i\}$ can be converted into the pixel location (x_i, y_i) inside an image,

$$x_i = s(u_i \cos \alpha + v_i \sin \alpha) + T_x \quad (4.16)$$

$$y_i = s(-u_i \sin \alpha + v_i \cos \alpha) + T_y, \quad (4.17)$$

where T_x and T_y are the translation parameters, s is the scaling coefficient and α is the rotation angle. Edge-distance transformation is implemented to generate a *floating* image with each pixel value, denoted as $D(x_i, y_i)$, equal to the nearest distance to the image edges. Hence our target is to find a set of $\{T_x, T_y, s, \alpha\}$ such that it minimize the error function, i.e., the summation of the edge distance along

Figure 4.13: Eight head-shoulder contour models

the positions of the transformed points

$$E = \sum_i D(x_i, y_i) \quad (4.18)$$

The updating of these parameters by the gradient descent algorithm is as follows,

$$\Delta T_x = -\gamma \frac{\partial E}{\partial T_x} = -\gamma \sum_i \left(\frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial T_x} + \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial T_x} \right) \quad (4.19)$$

$$\Delta T_y = -\gamma \frac{\partial E}{\partial T_y} = -\gamma \sum_i \left(\frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial T_y} + \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial T_y} \right) \quad (4.20)$$

$$\Delta s = -\gamma \frac{\partial E}{\partial s} = -\gamma \sum_i \left(\frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial s} + \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial s} \right) \quad (4.21)$$

$$\Delta \alpha = -\gamma \frac{\partial E}{\partial \alpha} = -\gamma \sum_i \left(\frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial \alpha} + \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial \alpha} \right) \quad (4.22)$$

where γ is the step size and can be different for different parameters. The partial derivatives with respect to parameters involved in the updating process are computed as follows:

$$\begin{aligned} \frac{\partial x_i}{\partial T_x} &= 1 & \frac{\partial y_i}{\partial T_x} &= 0 \\ \frac{\partial x_i}{\partial T_y} &= 0 & \frac{\partial y_i}{\partial T_y} &= 1 \\ \frac{\partial x_i}{\partial \alpha} &= s(-u_i \sin \alpha + v_i \cos \alpha) & \frac{\partial y_i}{\partial \alpha} &= s(-u_i \cos \alpha - v_i \sin \alpha) \\ \frac{\partial x_i}{\partial s} &= u_i \cos \alpha + v_i \sin \alpha & \frac{\partial y_i}{\partial s} &= -u_i \sin \alpha + v_i \cos \alpha \end{aligned} \quad (4.23)$$

The partial derivatives of the error function can be converted to other formats

$$\frac{\partial E}{\partial x_i} = \frac{\partial D(x_i, y_i)}{\partial x} = D_x(x_i, y_i) \quad (4.24)$$

$$\frac{\partial E}{\partial y_i} = \frac{\partial D(x_i, y_i)}{\partial y} = D_y(x_i, y_i) \quad (4.25)$$

where $D_x(x_i, y_i)$ and $D_y(x_i, y_i)$ are computed using *finite difference* based on the edge-distance transformed map.

4.1.5 Results for Locating Head and Shoulders

The face detection technique involved normally results in an accurate position for the center of the face and a rectangle that closely bounds the face region. Thus it is able to provide a good initialization for the transformation parameters, T_x , T_y , s . The rotation angle α , is initially set at zero with the assumption that the upper body is in the up-standing pose. Fig 4.14 shows the head-shoulder contour matching process for different images with different head scales and poses. The matching process is illustrated by a sequence of images. The faces detected in the second column of Fig 4.14 show a good starting point for the matching process. The results of the edge-distance transform are shown in the forth column of Fig 4.14. The edge-distance values are quantized into gray levels. The pixels that are near to the edges looks much darker than those pixels that are far away from the edges. From the four distance transformation map, it can be observed that the areas where the true contours lie are all fall into the dark regions. That means the starting points are already closed to the local modes or the positions with minimum matching cost. Thus it ensures the gradient descent approach searches in the nearby locations of the starting points and results in the good matching quality. In the fifth column, it can be seen that the contours are well fitted to the head-shoulder boundaries. The positions of the head and shoulders have been predefined for each contour model. Through the translation, rotation and scaling process, the new positions can be obtained given the best model parameters selected and are highlighted by green spots. Sometimes, the green spots deviates a bit from the true positions as depicted

in Fig 4.14e. This is probably due to the inflexibility for the curvatures of the contour models. The performance of the head-shoulder contour matching based on the Canny edge detection is also investigated. Fig 4.15(d) shows one of the best matchings between the contours and head-shoulder boundary. It shows that the alignment of the contour is indeed affected by the trivial edges.

Since the procedure to detect the head and shoulders is through the contour matching process, the detected positions will usually align with the boundary of the head and shoulders. A single position is not accurate enough to determine the true position of the neck or shoulder joints. Thus the first 50 candidate positions with the smallest edge distance for each image are used to calculate the mean positions and the sampling regions for the neck and shoulders are then defined using three rectangles centered at the corresponding mean positions as shown in Figure 4.16. The values for the width, height and rotation angle for the rectangles are determined empirically in order to cover the true positions with smallest possible regions. Here both the width and height are set as 8 pixels since the height of the persons inside the images ranges from 150 pixels to 200 pixels. The width and height of the rectangles can adapt to the size of the face rectangles detected for large human size but large sampling regions may increase the sampling complexity. The rotation angle is again set to zero since it is assumed the upper body is in the upstanding pose.

Figure 4.14: (a), (f), (k)&(p)Original image; (b), (g), (l)&(q)Face detection; (c), (h), (m)&(r)Binary edge image; (d), (i), (n)&(s)Edge-distance transformation map; (e), (j), (o)&(t)Head-shoulder contour matched

4.2 Limb Detection

Compared to the detection of the head and shoulders, the limb detection is much more difficult due to the lack of the salient features for representing them. The

Figure 4.15: (a)Original image; (b)Face detection; (c)Binary edge image; (d)Edge-distance transformation map; (e)Head-shoulder contour matched

Figure 4.16: Head-shoulder sampling regions

approach proposed here is an extension of the body-part detection method presented in [41].

4.2.1 Parallel Line Detection

Since the shape of the limbs in 2D images can be approximated by rectangles, the parallel-line detection is employed to find the positions of limbs. The binary edge used in the head-shoulder detection method doesn't provide the information of the straight lines but they can be found by simply finding the subset of the edge points that lie on them. To detect the straight lines in the binary edge image efficiently, the Hough transform is used. The basic idea is to represent a single line by the slope

a and the intercept b instead of two different points in the x-y plane. By quantizing the a and b values into discrete values, the a-b plane is subdividing into square cells. Then, every edge point falls into a number of cells, which means every edge point is passed through by a number of straight lines. So for the edge points lie in the same cell, they can be represented by a single line with the discrete slope and intercept value for that cell. To handle the infinity slope problem, the normal representation is select to replace the slope-intercept representation. Thus every line is equivalent to a point in the $\rho - \theta$ plane. The general approach for the Hough transform is as follows:

- subdividing the $\rho - \theta$ plane into cells;
- check the number of edge points on every cell and choose the cells with high point concentrations for line representation;
- check the connectivity of the points in a chosen cell and link the points with small gaps into one line.

Finally, two lines are grouped into a parallel-line set if the slopes for both lines are more or less the same.

4.2.2 Candidate Selection by Maximum Likelihood

The parallel-line detection results in a large number of redundant parallel-line sets due to the complex image content and noisy edges. To find the corresponding parallel-line set for each limb, the likelihood of each parallel-line set being the limb

is measured. Denote PL_n as the n -th parallel-line set of N parallel-line sets and L_m as the m -th limb of M limbs. Since human beings only have two types of limbs, i.e., arms and legs, M should equal to 2. The framework for computing the likelihood $p(PL_n|L_m)$ is formulated as follows.

$$p(PL_n|L_m) = p(d_n|L_m) \cdot p(b_n) \cdot p(R_n), \quad 1 \leq n \leq N \quad 1 \leq m \leq M \quad (4.26)$$

The first component $p(d_n|L_m)$ in Equation 4.26 represents the probability distribution of the distance d_n between the two lines in the n -th parallel-line set given the m -th limb. Intuitively, the distribution is approximated as Guassians:

$$p(d_n|L_m) \propto N(\mu_m, \sigma_m^2) \quad (4.27)$$

where μ_m and σ_m^2 are the mean and variance of the width of the m -th limb respectively.

The second component $p(b_n)$ represents the distribution for the parameter b_n , which gives the overlap ratio between the two lines in the n -th parallel-line set by projecting one line to another. The distribution is defined by:

$$p(b_n) = \exp^{-c(1-b_n)}, \quad 0 \leq b_n \leq 1 \quad (4.28)$$

where c is a constant determined empirically.

The last component $p(R_n)$ represents the region likelihood based on the image

segmentation proposed in [19]. The parameter R_n denotes the set of segmented regions that overlap with the n -th parallel-line set. The region likelihood is calculated using the approach similar to the one in [55]. Assume the area covered by the parallel-line set belongs to the foreground and the rest belongs to the background. For a region r inside R_n , denote N_{rf} as the number of pixels classified as the foreground within the region r and N_{rb} as the number of pixels classified as the background. The region r is then labeled as foreground if $N_{rf} > N_{rb}$ and background if $N_{rf} \leq N_{rb}$. As a result, the cost of the region r for being the foreground or background is defined as:

$$f_r = \begin{cases} N_{rb}/N_r & \text{if } r \text{ is foreground} \\ N_{rf}/N_r & \text{if } r \text{ is background} \end{cases} \quad (4.29)$$

where N_r is the total number of pixels in the region r . The total cost of R_n is then calculated as:

$$f_{R_n} = \sum_{r \in R_n} f_r \quad (4.30)$$

Finally, the region likelihood is computed as:

$$p(R_n) = \exp^{-h \cdot f_{R_n}} \quad (4.31)$$

where h is a constant.

Given the $p(PL_n|L_m)$ calculated for each parallel-line set, the first five parallel-line sets with the maximum likelihood are selected as the candidates for the m -th

limb. If the number of the parallel-line sets is less than five, then the current available parallel-line sets are used as candidate. In the worst case, if the number of the parallel-line sets is less than two, then the limb detection fails and the image will be discarded. Due to the similar color of the arm pair and leg pair, the histogram of each candidate is compared with another candidate and two candidates which have the highest histogram similarity are selected to represent the arm pair or leg pair. The left and right arms and legs are distinguished with the assumption that the centroid of the parallel-line set representing the left arm or leg is on the left hand of that of the right arm or leg. The sampling regions for the the elbow, wrist, knee and ankle are then determined from the detected limbs.

4.2.3 Results for Locating Arms and Legs

Figure 4.17a to Figure 4.17e show a sequence of images that result from the limb detection. Each limb is represented by an ellipse constructed given the length and width of the corresponding parallel-line set. The sampling regions for the joints are bounded by rectangles as depicted in Figure 4.18 and 4.19.

Figure 4.17: (a)Straight lines by Hough transform; (b)Parallel-line sets; (c)Segmented image; (d)Candidate parallel-line sets for limbs; (e)Candidate sets represented by ellipse; (f)Limbs detected

Figure 4.18: Sampling region for (a)left elbow; (b)left wrist; (c)right elbow; (d)right wrist

Figure 4.19: Sampling region for (a)left knee; (b)left ankle; (c)right knee; (d)right ankle

Chapter 5

Human Pose Inference

5.1 Data-Driven MCHMM in Human Pose Estimation

The motivation of integrating Monte Carlo methods with hidden Markov models is that HMM is an efficient Bayesian inference for estimating the posterior distribution $P(x|Y)$, where x denotes the vector of the pose parameters and Y denotes the image observations. By employing Monte Carlo methods, MCHMM is able to approximate all probability density functions required by HMM. The approximation is achieved using samples, along with density trees generated from these samples. This algorithm was first proposed by Thrun and Langford [83] with the objective of learning non-parametric hidden Markov models with continuous states and observation space and running HMMs in an any-time fashion. Here we apply the MCHMM algorithm in a searching problem and extend it by integrating the data driven technique in

order to explore the solution space more efficiently.

5.1.1 Hidden Markov Models

This section gives a brief introduction to the conventional hidden Markov models. HMMs are widely used in the area of temporal signal processing due to its outstanding performance in learning the statistical characteristics of partially observed stochastic time sequences. Rabiner presents a detailed description of HMMs in [73].

In most applications, HMMs are represented by a first order Markov chain, which assumes that the current state only depends on the previous state and the state transitions are time-invariant. Though the states in HMMs are unobservable, another set of stochastic processes associated with the state sequences can be observed by measuring the likelihood of the observation at time t conditioned on the state of the HMM. Thus, training an HMM is equivalent to estimating three probability densities denoted by:

$$\lambda = \{A, B, \Pi\}. \quad (5.1)$$

$A = \{a_{ij}\}$ defines the set of the state transition probability distributions.

$B = \{b_j(k)\}$ represents the set of observation densities given the states of the HMM.

$\Pi = \{\pi_i\}$ is the set of the initial state distribution.

Denote x_t as the state at time t , $S = \{s_1, s_2, \dots, s_N\}$ as the set of N individual states in the HMM and $V = \{v_1, v_2, \dots, v_M\}$ as the set of M observation symbols.

The three probability distributions can be expressed mathematically as follows:

$$a_{ij} = P(x_{t+1} = s_j | x_t = s_i), \quad 1 \leq i, j \leq N. \quad (5.2)$$

$$b_j(k) = P(v_k \text{ at } t | x_t = s_j), \quad 1 \leq j \leq N, \quad 1 \leq k \leq M. \quad (5.3)$$

$$\pi_i = P(x_1 = s_i), \quad 1 \leq i \leq N. \quad (5.4)$$

These distributions are derived from a sequence of observations denoted as $O = \{o_1, o_2, \dots, o_T\}$, where o_t is the observation at time t and T is the total number of observations. As a result, the problem for estimating the model parameters $\lambda = \{A, B, \Pi\}$ is to find the model parameters that maximize the likelihood $P(O|\lambda)$. The most well-known algorithm for solving this problem is the Baum-Welch method, which provides a computationally efficient approach based on the Expectation-Maximization (EM) method. A brief description of this algorithm is presented below.

In E-step, to ease the computational load, a set of additional distributions needs to be defined as follows. In the n -th E-step,

$$\alpha_t^{(n)}(i) = P(x_t = s_i | o_1, \dots, o_t, \lambda^{(n)}) \quad (5.5)$$

$$\beta_t^{(n)}(i) = P(o_{t+1}, \dots, o_T | x_t = s_i, \lambda^{(n)}) \quad (5.6)$$

$$\gamma_t^n(i) = P(x_t = s_i | O, \lambda^{(n)}) \quad (5.7)$$

$$\xi_t^n(i, j) = P(x_t = s_i, x_{t+1} = s_j | O, \lambda^{(n)}) \quad (5.8)$$

The discrete version of the Baum-Welch algorithm is described in [73]. The continuous case is analogous. The procedure for the estimation of the parameters of an HMM is shown below. At each iteration of the E-step, firstly the forward variable α at time 1 and the backward variable β at time T are initialized as follows:

$$\alpha_1^{(n)}(s) = \pi^{(n)}(s) \quad (5.9)$$

$$\beta_T^{(n)}(s) = 1 \quad (5.10)$$

while all other α_t and β_t are calculated as

$$\alpha_t^{(n)}(s) = \int \alpha_{t-1}^{(n)}(s') a^{(n)}(s|s') b^{(n)}(o_t|s) ds' \quad (5.11)$$

$$\beta_t^{(n)}(s) = \int \beta_{t+1}^{(n)}(s') a^{(n)}(s'|s) b^{(n)}(o_{t+1}|s') ds' \quad (5.12)$$

where s and s' are the points in the state space. Hence, the conditional density γ at time t can be derived from α_t and β_t based on the Bayesian rule.

$$\gamma_t^{(n)}(s) = \frac{\alpha_t^{(n)}(s) \beta_t^{(n)}(s)}{\int \alpha_t^{(n)}(s') \beta_t^{(n)}(s') ds'} \quad (5.13)$$

Similarly, the state transition density ξ at time t is computed as

$$\xi_t^{(n)}(s, s') = \frac{\alpha_t^{(n)}(s) a^{(n)}(s'|s) b^{(n)}(o_{t+1}|s') \beta_{t+1}^{(n)}(s)}{\int \int \alpha_t^{(n)}(\bar{s}) a^{(n)}(\bar{s}'|\bar{s}) b^{(n)}(o_{t+1}|\bar{s}') \beta_{t+1}^{(n)}(\bar{s}) d\bar{s} d\bar{s}'} \quad (5.14)$$

where $\bar{s}$ and $\bar{s}'$ also represent the points in the state space.

Based on the $\gamma_s^{(n)}$ and $\xi_t^{(n)}(s, s')$ derived above, the M-step computes a new HMM λ at iteration $n + 1$ as follows:

$$\pi^{(n+1)}(s) = \gamma_1^{(n)}(s) \quad (5.15)$$

$$A^{(n+1)}(s'|s) = \frac{\sum_{t=1}^{T-1} \xi_t^{(n)}(s, s')}{\sum_{t=1}^{T-1} \gamma_t^{(n)}(s)} \quad (5.16)$$

$$B^{(n+1)}(v|s) = \frac{\sum_{t=1}^T I(o_t = v) \gamma_t^{(n)}(s)}{\sum_{t=1}^T \gamma_t^{(n)}(s)} \quad (5.17)$$

where $I(x)$ is an indication function, which equals to 1 if x is true and 0 otherwise.

5.1.2 Density Approximation

In contrast to the conventional HMMs, the MCHMM estimates the HMM parameters via the density approximation. This section describes the sample-based and tree-based methods for density approximation.

Two sampling techniques which are the special cases of *importance sampling* are employed by the MCHMM, namely, likelihood-weighted sampling and uniform sampling. The importance sampling method approximates a density f by generating samples s from a non-zero support density distribution g and setting the sample weight ω_s proportional to $\frac{f(s)}{g(s)}$.

In **likelihood-weighted sampling**, the support distribution g is set to $g \equiv f$. Thus, a set of N samples $S = \{\langle s_1, \omega_{s_1} \rangle, \dots, \langle s_N, \omega_{s_N} \rangle\}$ can be obtained as follows:

$$s_n \sim f(s), \quad 1 \leq n \leq N \quad (5.18)$$

$$\omega_{s_n} = N^{-1} \quad (5.19)$$

In **uniform sampling**, g is a uniform distribution U over the domain of the density f . As a result, the sampling process is shown below.

$$s_n \sim U, \quad s_n \in \text{dom}(f) \quad (5.20)$$

$$\omega_{s_n} = f(s_n) \left[\sum_{i=1}^N f(s_i) \right]^{-1} \quad (5.21)$$

where $\text{dom}(f)$ represents the domain of the density f .

It has been approved that both likelihood-weighted sampling and uniform sampling are asymptotically consistent since for $N \rightarrow \infty$, S converges to the density f with probability 1 when integrated over the system of half-open Borel sets, i.e.,

$$\lim_{N \rightarrow \infty} \sum_{\langle s, \omega_s \rangle \in S} I(s \in \text{dom}(f)) = \int_{\text{dom}(f)} f(s) ds \quad \text{w.p. 1} \quad (5.22)$$

However, in a high dimensional space, only using the sampling method is not enough for representing a complex distribution. Hence a tree-based approximation method, called **density trees**, is employed to transform the sparse sample set into the density functions. Another advantage associated with density trees is that they can approximate the combination of different densities. For example, $f \cdot g$ is the multiplication of two density functions f and g defined over the same domain. Using the sampling method, two sets of samples S_f and S_g are drawn from the two densities respectively. Intuitively, it can be proved that none of the samples in S_f and S_g are

identical with probability 1 since the samples are discrete.

Not all density trees are asymptotically consistent in approximating distributions. A simple tree growing method is described here. The asymptotic consistency of this algorithm is proved in [84]. First of all, the root node of the tree is initialized to cover all samples. The space of the root node is set to the entire domain of f denoted by D . At each level of the tree, a node i need to be divided into two sub-nodes if the following two criterions are satisfied.

- At least $\sqrt{N}$ samples fall into the space of the i -th node denoted by D_i .
- Its distance from the root node is less than $\lceil \frac{1}{4} \lg_2 N \rceil$.

The division results in two equal-space sub-nodes by cutting the longest dimension of the space of the i -th node equally. A node becomes a leaf node of the density tree if the splitting process can't be continued. Consequently, it is observed that the entire solution space is partitioned into mutually exclusive sub-space represented by the leave nodes in the tree. A typical structure of the density tree and its growing process is shown in Figure 5.1.

The weight for an arbitrary node i denoted by σ_i is calculated by summing the weight of all the samples which fall into the region of this node D_i , i.e.,

$$\sigma_i = \sum_{\langle s, \omega_s \rangle \in S} I(s \in D_i) \omega_s \quad (5.23)$$

Since the probabilities of all the K nodes in the same level sum to 1, the weight of

Figure 5.1: The structure and grow process of the density tree. The domain of each node is marked by grey color.

each node is then normalized.

$$\bar{\sigma}_i = \frac{\sigma_i}{\sum_1^K \sigma_j} \quad (5.24)$$

Thus, the stationary distribution of the solution space can be approximated by the weight of the leaf nodes. Such an approximation method may cause incorrect convergence of the MCHMM. As shown in Figure 5.2, the distribution contains an

Figure 5.2: A distribution with an isolated mode.

isolated mode q_0 . With the random initialization, the number of the sparse samples may not be sufficient to cover the isolated mode. In such a scenario, the MCHMM probably converges to the mode q_1 , which has a wide support and a relatively high probability distribution. Another reason for the incorrect convergence is the splitting rule of the density tree. As mentioned above, a tree node is equally divided into two sub nodes by cutting the longest dimension of the space of the parent node. Hence it is possible that the isolated mode q_0 in Figure 5.2 will be divided into two sub nodes with the probability distribution spread out and degraded. On the other hand, the splitting process has little effect on the the distribution of the mode q_1 . Eventually more samples are generated from the mode q_1 rather than from q_0 and incorrect convergence occurs. To handle the two problem mentioned above, an annealing technique is implemented. The weight of each leaf node is recalculated by mixing the weight of each node along the path from the leaf node to the root node. As shown in Figure 5.1, node i, j, k, l and m are the nodes along the path from the leaf node m to the root node i . Denoting the normalized weight as $\bar{\sigma}_i, \bar{\sigma}_j, \bar{\sigma}_k, \bar{\sigma}_l$ and $\bar{\sigma}_m$ for node i, j, k, l and m respectively, the mixing rule is defined as follows:

$$\hat{\sigma}_m = \rho^4 \bar{\sigma}_i + (1 - \rho) \rho^3 \bar{\sigma}_j + (1 - \rho) \rho^2 \bar{\sigma}_k + (1 - \rho) \rho \bar{\sigma}_l + (1 - \rho) \bar{\sigma}_m \quad (5.25)$$

where ρ is a constant annealing factor with $0 \leq \rho \leq 1$. The re-estimate weight $\hat{\sigma}$ is then normalized in the leaf level. According to Equation 5.25, it can be seen that when the value of ρ is close to 1, the root node contributes more to the weight estimation at the leaf node. Hence a relatively flat distribution is obtained at first.

In the Baum-Welch algorithm, at each iteration, the value of ρ is decreased through Equation 5.26:

$$\rho = \rho \bar{\rho} \quad (5.26)$$

where $\bar{\rho}$ is a constant less than 1. Thus, as the value of ρ is reduced from 1 exponentially, the influence of the nodes above the leaf node also decays. The further the node is away from the leaf node, the less the influence it has. The initial relatively flat distribution is also evolved to a more representative distribution.

5.1.3 Data-Driven MCHMM Algorithm

With the integration of the HMM and two density approximation methods, a general framework of the data-driven MCHMM for the pose estimation can be formulated as shown below.

At the **initialization** stage, denote the HMM as $\lambda^{(0)} = \{A^{(0)}, B^{(0)}, \Pi^{(0)}\}$

- a. Randomly draw three sets of N samples:

$$s_i^A \sim U, \quad s_i^A \in \text{dom}(A^{(0)}) \quad 1 \leq i \leq N \quad (5.27)$$

$$s_i^B \sim U, \quad s_i^B \in \text{dom}(B^{(0)}) \quad 1 \leq i \leq N \quad (5.28)$$

$$s_i^\Pi \sim U, \quad s_i^\Pi \in \text{dom}(\Pi^{(0)}) \quad 1 \leq i \leq N \quad (5.29)$$

- b. Set the respective sample weight with $\omega_{s_i^A} = 1/N$, $\omega_{s_i^B} = p(O|s_i^B)$ and $\omega_{s_i^\Pi} = 1/N$, where $p(O|s_i^B)$ is the image likelihood given the current state of the

human pose s_i^B . Note that in the original MCHMM, the sample weight $\omega_{s_i^B}$ is set to $1/N$. The reason is that the algorithm is initially designed for model learning problem. Since the training samples used for model training are all positive, it is reasonable to set their sample weights identical. However, for a model searching problem, there are no positive or negative training samples. Equal sample weights are informationless and will lead to poor convergence in most times. Hence to guide the search process, the image likelihood has to be measured and used as the sample weight.

- c. Generate three density trees $tr(A^{(0)})$, $tr(B^{(0)})$ and $tr(\Pi^{(0)})$ from the three sample sets respectively. The symbol $tr(f)$ denotes the density tree for the density f .

In the k -th **E-step**, three distributions $\alpha^{(k)}$, $\beta^{(k)}$ and $\gamma^{(k)}$ are calculated for T time sequences.

Initialize $\alpha^{(k)}$ by $tr(\alpha_1^{(k)}) \equiv tr(\Pi^{(n)})$, $\alpha_t^{(k)}$ for $1 < t \leq T$ can be computed through the following procedure.

- a. Generate N samples at time $t-1$: $\{s_{t-1,i}\}_{i=1}^N \sim tr(\alpha_{t-1}^{(k)})$.
- b. For each sample $s_{t-1,i}$, construct a child-tree for the conditional density $A(s_{t,i}|s_{t-1,i})$ from $tr(A^{(k)})$. Draw the i -th sample at time t : $s_{t,i} \sim tr(A(s_{t,i}|s_{t-1,i}))$.
- c. Set $\omega_{s_{t,i}} \propto B(O|s_{t,i})$, where $B(O|s_{t,i})$ is obtained from $tr(B^{(k)})$.
- d. Build $tr(\alpha_t^{(k)})$ based on the new sample set $\{s_{t,i}, \omega_{s_{t,i}}\}_{i=1}^N$.

The computation of $\beta^{(k)}$ is similar to that of $\alpha^{(k)}$. Let $\beta_T^{(k)} = 1$, i.e., represent $tr(\beta_T^{(k)})$ only by the root node, $tr(\beta_t^{(k)})$ for $1 \leq t < T$ are computed as follows.

- a. Generate N samples at $t+1$: $\{s_{t+1, i}\}_{i=1}^N \sim tr(\beta_{t+1}^{(k)})$.
- b. Draw the i -th sample at time t : $s_{t, i} \sim tr(A(s_{t+1, i}|s_{t, i}))$.
- c. Set $\omega_{s_{t, i}} \propto B(O|s_{t+1, i})$.
- d. Build $tr(\beta_t^{(k)})$ based on $\{\langle s_{t, i}, \omega_{s_{t, i}} \rangle\}_{i=1}^N$.

Based on the $tr(\alpha_t^{(k)})$ and $tr(\beta_t^{(k)})$ estimated above, the computation of $\gamma_t^{(k)}$ for $1 \leq t \leq T$ is described below.

- a. Generate $N/2$ samples at t : $\{s_{t, i}\}_{i=1}^{N/2} \sim tr(\alpha_t^{(n)})$ and set $\omega_{s_{t, i}} \propto \beta_t^{(k)}(s_{t, i})$.
- b. Generate $N/2$ samples at t : $\{s_{t, i}\}_{i=N/2+1}^N \sim tr(\beta_t^{(n)})$ and set $\omega_{s_{t, i}} \propto \alpha_t^{(k)}(s_{t, i})$.
- c. Build $tr(\gamma_t^{(k)})$ based on $\{\langle s_{t, i}, \omega_{s_{t, i}} \rangle\}_{i=1}^N$.

In the k -th **M-step**, The new HMM $\lambda^{(k+1)}$ is calculated based on the $\{\alpha_t^{(k)}\}_{t=1}^T$, $\{\beta_t^{(k)}\}_{t=1}^T$ and $\{\gamma_t^{(k)}\}_{t=1}^T$ estimated in the k -th E-step.

For the state transition density $A^{(k+1)}$,

- a. Sample N time indexes: $t \sim U, t \in [1, T-1]$.
- b. Given each time t , $s_{t, i} \sim tr(\gamma_t^{(k)})$ and $s_{t+1, i} \sim tr(\gamma_{t+1}^{(k)})$.
- c. Set $\omega_{s_{t, i}, s_{t+1, i}} = 1/N$.
- d. Group the new N samples with the existing sample set that forms $tr(A^{(k)})$ and build $tr(A^{(k+1)})$ from the new sample set.

For the observation density $B^{(k+1)}$,

- a. Sample N time indexes: $t \sim U, t \in [1, T]$.
- b. Given each time t , $s_{t,i} \sim tr(\gamma_t^{(k)})$
- c. Set $\omega_{s_{t,i}} \propto p(O|s_{t,i})$.
- d. Group the new N samples $\{\langle s_{t,i}, \omega_{s_{t,i}} \rangle\}_{i=1}^N$ with the existing sample set that forms $tr(B^{(k)})$ and build $tr(B^{(k+1)})$ from the new sample set.

For the initial state distribution Π^{k+1} , group the sample set representing $\gamma_1^{(k)}$ with the existing sample set that forms $tr(\Pi^{(n)})$ and build $tr(\Pi^{(n+1)})$ from the new sample set.

The annealing factor for the $k+1$ iteration is updated as defined in Equation 5.26. To ensure the convergence of the MCHMM [84], the sample set size N is increased to ηN , where η is a constant bigger than 1.

5.1.4 Computational Complexity Analysis

Based on the data-driven MCHMM algorithm described above, the algorithm mainly requires to compute six distributions. For each of them, the computation basically requires the generation of a new sample set from existing distributions and the construction of a new density tree. To generate N samples, the computation cost mainly concentrates on the generation of N random numbers. Hence the computation time for sampling increases linearly with the number of samples. In other words, the sampling process has a time complexity of $\Omega(N)$. For the construction of the density

tree, since the annealing technique is employed here, The weight of each node in each level of the tree needs to be computed. For each level, the time complexity is $\Omega(N)$. As the distance from the root node is less than $\lceil \frac{1}{4} \lg_2 N \rceil$, the time complexity for building the entire density tree is $\Omega(N \lg_2 N)$, i.e., the computation time increases loglinearly with the number of samples. Therefore the overall time complexity for computing each distribution is $\Omega(N \lg_2 N)$.

5.1.5 Implementation Issues

As mentioned before, the human pose estimation problem is actually a search problem in a high dimensional space. Given the sampling regions constructed in the body-part detection stage, the human model can be represented by a 22D vector of pose parameters which specify the image positions of the main body joints including neck, shoulders, elbows, wrists, knees and ankles. In addition, a 3D vector is employed to describe the global positions and size of the human model. To determine the image positions of the hips, an additional parameter which defines the inclination of the torso with respect to the image plane is also employed. Thus the high dimensional space is described by 26 dimensions and one human model is actually equivalent to a point in the 26D space. So when applying the data-driven MCHMM framework into the human pose estimation, a single MCHMM is not enough to infer the pose. Instead, a series of MCHMMs is employed to sequentially estimate the pose parameters of different body parts. The reason is that using a single MCHMM requires a very large number of samples to sufficiently cover the high dimensional so-

lution space even for a sparse representation as the number of samples is increased exponentially with the increase of the space dimension. The sequential approach can dramatically reduce the total number of samples and achieve computationally efficient. As depicted in Figure 5.3, totally five data-driven MCHMMs are used to

Figure 5.3: A data-driven MCHMM framework for human pose estimation.

iteratively estimate the pose parameters of different body parts in the order of head, left arm, right arm, left leg and right leg. The pose parameters for the head are just the image positions for the neck joint. While the pose parameters for the arm include the image positions for the shoulder, elbow, and wrist and for the leg consist of the torso inclination and the image positions for the knee and ankle. As the head is only represented by a 2D vector, the pose parameters for the global positions and size are combined with the head and are inferred together by the same MCHMM. The sequential approach is as follows. At first, an initial human pose, which fit the kinematic constraints are sampled from the entire space. The five MCHMMs are then initialized by randomly generating samples from the sub-spaces. In the iterative process, each MCHMM only conducts one EM run and updates the corresponding partial parameters for the next MCHMM. The partial pose parameters for update is obtained via the following procedure: find the leaf node with the maximum weight

in $\text{tr}(B^{(n)})$ and then select the sample with the maximum observation likelihood among the set of samples lying in the leaf node. With this framework, the initial annealing factor ρ should not be set to 1 otherwise the weight of each leaf node is the same and the sample selected for updating may not be kinematically possible. Hence, ρ is set to 0.95. To determine the sample size for each EM run, both the computational efficiency and coverage of the solution space need to be taken into consideration. As discussed above, the computation time increases loglinearly with the increase of the sample size. The number of the samples should remain as low as possible. On the other hand, the number of the samples should be as big as possible to give sufficient coverage of the sample space and maximize the convergence rate. For the original MCHMM, there are no constraints on the image position of each body joint so the sampling space for each joint turns out to be the entire image and it is impossible to find the appropriate number of samples which can ensure sufficient coverage while maintain computationally efficient. By incorporating the data-driven method, the candidate positions for each joint are narrowed down to a small rectangle detected. Eventually a small number of samples are enough to cover the sampling space. Thus compared to the MCHMM, the data-driven MCHMM can find the correct human pose within a short period of time. Here, the number of samples N for the first iteration is set to 500 as size of the sub-space is up to six dimensions. At each round, N is increased by 50.

5.2 3D Human Model & Image Likelihood

In order to measure the image likelihood for the human model, the 26D vector that represents the pose parameters is converted geometrically to obtain the orientation and translation of each body part in the 3D space. A 3D human model which defines the articulated structure of the human body is then built. It consists of 10 body parts including head, torso, upper arms, lower arms, upper legs and lower legs. Each of them is represented by a truncated 3D cone as shown in Figure 5.4. The orientation of each 3D cone is achieved by means of a single rotation around

Figure 5.4: 3D human model (a)front view; (b)profile view

an appropriate axis. So the orientation parameters needed is the rotation angle ϕ and the direction of the rotation axis v as depicted in Figure 5.5. Details of this orientation method can be found in [45]. Sometimes the 3D model generated may not be possible in reality. A kinematic constraint is imposed to eliminate the unfeasible human models. This constraint also prunes the solution space and help to improve the searching efficiency. Intuitively, the angles between the 3D axes and the symmetric axis of the 3D cone are used to check whether the position of the 3D cone violate the kinematic constraint. Figure 5.6 shows the three angles defined

Figure 5.5: Orientation in 3D space

here.

Figure 5.6: Three angles defined for checking kinematic constraint

By projecting the 3D model into a 2D plane, a binary image can be obtained.

Mapping the binary image to the observed image, the hypothesized human region

and non-human region are obtained. The image likelihood can then be calculated by measuring the region likelihood and color likelihood. The measurement of the region likelihood is the same as the approach mentioned in the limb detection while the measurement of the color likelihood adopts the approach in [55].

Chapter 6

Experimental Results

6.1 Image Data

The images used in this experiment show various of human poses. In the estimation process, the images are initially scaled to a fixed height in order to ease the computation of parallel line detection. There are no constraints on the image background or the position, size and clothing of the human inside the image. However the whole human body is required to be present and the human face should be visible. Self-occlusion is allowed as long as the color of the occluded body parts is distinctive. The above requirement is due to the pose estimation framework employed. As shown in Figure 5.3, the iterative process for the data-driven MCHMM involves 5 EM runs for different body parts. So if one body part is severe occluded or completely invisible, then the candidate positions for different body joints may be missing or incorrect. Eventually the iterative process cannot work properly and

it leads to the wrong estimation of the human pose.

6.2 3D Pose Result & Analysis

Figure 6.1 presents the 3D poses estimated from the 2D images. The original images are listed in the first row. The 3D models are projected orthogonally to the 2D images and shown in the second row. The last row gives the profile views of the 3D models in order to show the depth information. The poses varies from sitting to standing in different orientations. It can be seen that the projections of the 3D models are well matched with the humans inside the images. Also the depth estimated is similar to the subjective observations by the human eyes. For the image in Figure 6.1b, the arms and the torso are self-occluded but the poses for both arms are still interpreted closely with the correct poses since the sample regions for both the wrists and elbows are well detected. Due to the lack of the ground truth data of the positions of the joints, the accuracy can not be estimated quantitatively. However some obvious errors can still be observed. For example, the legs in Figure 6.1c and Figure 6.1e should be vertical to the ground but they incline slightly backward. Moreover the tips of the lower arms and legs, i.e., the wrists and ankles may not match exactly with the images positions such as the left wrists in Figure 6.1d and Figure 6.1e and the ankles in Figure 6.1b. In Figure 6.1a and Figure 6.1b, the poses of the right arms are not correct either.

The main reason for these errors is that the human pose estimation is an inference problem over a multi-mode distribution. Due to the limitation of the image

Figure 6.1: 3D pose models estimated from 2D images. First row: original images; Second row: projection on 2D images; Third row: profile views

segmentation, the human object inside the image is hardly to be segmented as a whole blob but rather it is segmented into several or even a number of small regions. As a result, different 3D models may have similar likelihood when projecting back to the 2D image and the mismatch of the joint positions will occur. Another reason could be the incapability of the 3D human model designed. In this project, the 3D model only defines its global size. Therefore if the relative size of the body parts doesn't follow the 3D model, the pose may not be estimated correctly such as the errors occurred in Figure 6.1a and Figure 6.1b. In addition, the shot angle of the camera may also cause the relative size of the body parts changes and lead to wrong

estimation as shown in Figure 6.1e.

The pose estimation results shown above are comparable to the work in [55]. The major difference is that our work estimates the postures from random initialization instead of proposed initial distributions. To illustrate the process of evolving from arbitrary postures to the correct ones, three sets of distributions are employed here. As mentioned above, the sampling region for each joint is represented by a rectangle. If the rectangle is divided into smaller rectangles, it should be expected that initially the number of samples falling into each sub-rectangle is more or less the same and gradually the sampling will concentrate more and more on a few or even one sub-rectangle. Thus a series of distributions for the x dimension of the left knee in Figure 6.1a can be constructed as follows: first divide the width of the sampling region into several equally spaced portions; second for each iteration of the data-driven MCHMM, find the number of samples falling into each portion given the total number of samples generated so far; finally normalized the number of samples for each portion by the total number of samples. Based on this procedure, the convergence process for the sampling can be observed as shown in Figure 6.2. From Figure 6.2a to Figure 6.2c, it can be seen that at first few iterations, the distributions are relative flat and none of the portions contains samples that exceeds 10% of the total samples. This is because of the annealing technique employed. Such an initialization ensures that the overall sampling region is covered. However, from Figure 6.2d, the distribution starts to converge to a local maximum mode. The underlying concept is that as the annealing factor decreases, the data-driven method

forces the sampling concentrating more on the portion which can generate postures with high likelihood. Similar convergence process can be observed in Figure 6.3 and Figure 6.4 which show the distributions for the y dimension and 2D space of the sampling region respectively. Based on Figure 6.2, Figure 6.3 and Figure 6.4, it can be seen that the MCHMM converges to the local maximum after ten iterations. The computation time required for each image is six minutes on average on a Pentium IV 1.8GHz computer with 1Gb memory.

Figure 6.2: Sample distribution over the width of the right-knee sampling region for the image in Figure 6.1a for (a)first MCHMM iteration; (b)first three MCHMM iterations; (c)first five MCHMM iterations; (d)first six MCHMM iterations; (e)first eight MCHMM iterations; (f) first ten MCHMM iterations. The horizontal axis indicates the x coordinates of the samples while the vertical axis indicates the normalized sample distribution.

Figure 6.3: Sample distribution over the height of the right-knee sampling region for the image in Figure 6.1a for (a) first MCHMM iteration; (b) first three MCHMM iterations; (c) first five MCHMM iterations; (d) first six MCHMM iterations; (e) first eight MCHMM iterations; (f) first ten MCHMM iterations. The horizontal axis indicates the y coordinates of the samples while the vertical axis indicates the normalized sample distribution.

Figure 6.4: Sample distribution over the 2D space of the right-knee sampling region for the image in Figure 6.1a for (a)first MCHMM iteration; (b)first three MCHMM iterations; (c)first five MCHMM iterations; (d)first six MCHMM iterations; (e)first eight MCHMM iterations; (f) first ten MCHMM iterations. The two horizontal axes indicates the x and y coordinates of the samples while the vertical axis indicates the normalized sample distribution.

Chapter 7

Conclusions and Recommendation

7.1 Conclusion

In this thesis, we present a new pose estimation system that is able to estimate arbitrary human postures in 2D images without the constraints on the background and clothing. Two main problems have been considered here. One is the initialization of the system and another is the search of the human pose in the high dimensional solution space. A top-down and bottom-up approach is adopted to handle the two problems. The underlying concept is to first find the sampling regions of different body joints based on several image cues and then search the sampling regions for the best 3D model using the data-driven MCHMM algorithm. Hence the pose estimation system can be divided into two stages, the body-part detection stage and the pose inference stage.

The first stage of this system focuses the work on the detection of the human

body parts including the head, shoulders, arms and legs. The candidate regions for the associated body joints can then be derived. The objective of this stage is to initialize the system automatically and provide analytic inference for the next stage. Two detection techniques are implemented based on an integration of three low-level image cues, the Haar-like features, edges and segmented blobs. The head and shoulders are found by the head-shoulder contour matching process proposed in [54]. The basic idea is to first detect the face with the Haar-like features. With the initial position and size of the face detected, a gradient descent approach is able to start to search for the contours transformed from eight exemplar ones and select the one with the minimum edge distance by aligning it to the edge extracted. The positions of the head and shoulders can be determined by the predefined positions associated with the exemplar contours. Through this approach, the detection of the head and shoulders is accurate in most cases except for some contours that are not well modeled by the exemplar contours. The sampling regions for the neck and shoulder joints can then be easily constructed. Unlike the detection of the head and shoulders, the limb detection is relatively hard as there are no salient features to represent human limbs. The parallel-line detection is employed since the shape of the limbs in 2D images can be approximated by rectangles. However, due to the imperfect edge detection, the parallel-line sets are redundant for determining the limb positions. To select the best parallel-line set for each limb, a framework based on the maximum likelihood estimation is proposed. The maximum likelihood is measured by three criteria, the distance between the two lines of the parallel-

line set, the overlap between the two lines and the portion of the segmented blobs covered by the parallel-line set. The first few parallel-line sets with the maximum likelihood are then selected as the candidates for the arms or legs. The similarity of the histogram for every two candidates is measured and the pair of the candidates with the highest histogram similarity is chosen as the pair of the arms or legs. A deterministic approach based on the geometry information is then used to distinguish between the left and right arms or legs. Finally, with the left and right limbs detected the sampling regions for the joints associated with each limb can be constructed. The detection accuracy highly depends on the complexity of the background and quality of the edge detection.

The second stage tries to find the best 3D human configuration with the sampling regions for each joint constructed in the first stage. Due to the articulated body structure, the configuration inference has to be designed to handle the problem of searching in the high dimensional space. The problem is solved under a Bayesian framework here. This is equivalent to compute the *posterior* distribution via the integration of a high-dimensional probability distribution. In such a circumstance, the analytic evaluation is computationally intractable. Instead a Monte Carlo based Bayesian inference method called data-driven Monte Carlo Hidden Markov Model (MCHMM) is proposed to compute the *posterior* distribution via density approximations. In contrast to other Monte Carlo approaches, the MCHMM employs both sample-based and tree-based techniques for density approximations. Compared with the large sampling space, the samples generated can only provide a sparse represen-

tation. To compensate the drawback of the sample based method, the tree-based technique is employed to approximate the density distribution by the density tree which is built given the relative likelihood for each sample generated. At every iteration of the Expectation and Maximization, a new set of samples is generated from the current approximated density distribution and used to construct the density tree for next iteration. Through this process, the approximated density distribution can gradually converge to the stationary distribution. Hence the MCHMM is capable to infer human poses with random initialization and lift away the necessity of the proposal distribution, which is widely used in other Monte Carlo methods. The annealing and data-driven methods are integrated to further improve the efficiency and the convergence rate of the probabilistic inference. The experimental results show that the proposed algorithm works well for images with different human postures and image contents.

7.2 Future Work

The future research directions based on the work presented in this thesis includes:

Integrating the human detection method

The current pose estimation system is designed for the images with the human body as the main object. To estimate the human poses for arbitrary images, the human detection method needs to be integrated in future to first locate the position of the human body in the image. The purpose is to eliminate most of the background of the image and minimize its effect on the body-part detection, which is normally

worked on the basis of a few low-level image cues. For the moving human object or videos with the static background, the detection is straightforward. The motion detection or background subtraction is usually sufficient to find the human object. However, the human detection in static images still remains a big issue due to the lack of robust features for the human representation. A possible solution to this problem is to use the Haar-like features and the head-shoulder contours to represent the human upper body. Similar to the head-shoulder contour matching process used for the body-part detection, the head-shoulder contours are also modeled by the type of the exemplar contour, the translation parameters, the rotation angle and the scale. A Monte Carlo based method like MCHMM or the Markov Chain Monte Carlo(MCMC) can be used to infer the model parameters. The dimensions of the solution space are delineated by the parameter. For every sample sampled from the solution space, the likelihood can be measured by the edge distance and the similarity of the Haar-like features extracted.

Improving the computation efficiency

Another issue related to the current system is that constructing density trees is quite time-consuming. The current approach selects the binary sorting algorithm for constructing the density tree. Hence the computation time increases exponentially with the number of the samples increased. To improve the computational speed, a better sorting algorithm may need to be investigated in future. Moreover, we are also trying to integrate the density tree approximation with the MCMC algorithm. Thus instead of building the density trees for the six distributions involved in the

Hidden Markov Model, the MCMC only requires a single density tree for updating the proposal distribution at each iteration.

Tracking the human body-part movement

The 3D postures estimated by the current system may not carry enough information for the human activity recognition. The reason is that the human activity often happens in a sequence of images rather than a single image. Therefore the pose estimation system needs to be extended to a body-part tracking system to retrieve the motion trajectories of different body joints. The pose estimation system may be only required to be applied in the key frames instead of being applied in every frame. The key frames can be selected with short intervals apart. Assuming the human movement is smooth in a short period, the motion trajectory can then be recovered from the discrete 3D positions in a video index basis. In the tracking system, the body-part detection can be further enhanced by the appearance model built upon the successful pose estimation. The appearance models are able to simplify the detection work and increase its accuracy but live updating of the appearance models is required.

Bibliography

- [1] A. Agarwal and B. Triggs, “Tracking articulated motion using a mixture of autoregressive models,” in *Proc. of European Conference on Computer Vision*, Prague, Czech Republic, May 2004, pp. 54–65.
- [2] J. K. Aggarwal and Q. Cai, “Human motion analysis: a review,” *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, Mar. 1998.
- [3] J. K. Aggarwal, Q. Cai, W. Liao, and B. Sabata, “Nonrigid motion analysis: articulated and elastic motion,” *Computer Vision and Image Understanding*, vol. 70, no. 2, pp. 142–156, May 1998.
- [4] K. Akita, “Image sequence analysis of real world human motion,” *Pattern Recognition*, vol. 17, no. 1, pp. 73–83, 1984.
- [5] C. Andrieu, N. D. Freitas, A. Doucet, and M. I. Jordan, “An introduction to mcmc for machine learning,” *Machine Learning*, vol. 50, no. 1, pp. 5–43, Jan. 2003.

-
- [6] A. A. Argyros and M. I. A. Lourakis, “Real-time tracking of multiple skin-colored objects with a possibly moving camera,” in *Proc. of European Conference on Computer Vision*, Prague, Czech Republic, May 2004, pp. 368–379.
 - [7] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
 - [8] A. Azarbayejani, C. Wren, and A. Pentland, “Real-time 3-d tracking of the human body,” Media Lab Perceptual Computing, MIT, Cambridge, MA, TR 374, 1996.
 - [9] A. M. Baumberg and D. C. Hogg, “An efficient method for contour tracking using active shape models,” in *Proc. of the Workshop on Motion of Non-Rigid and Articulated Objects*, Nov. 1994, pp. 194–199.
 - [10] A. G. Bharatkumar, K. E. Daigle, M. G. Pandey, Q. Cai, and J. K. Aggarwal, “Lower limb kinematics of human walking with the medial axis transformation,” in *Proc. of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, TX, USA, 1994, pp. 70–76.
 - [11] A. F. Bobik and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

-
- [12] C. Bregler and J. Malik, “Estimating and tracking kinematic chains,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, USA, 1998, pp. 8–15.
- [13] C. H. Cai and S. K. Mitra, “A normalized color difference edge detector based on quaternion representation,” in *Proc. of International Conference on Image Processing*, Vancouver, BC, Canada, Sept. 2000, pp. 816–819.
- [14] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, Nov. 1986.
- [15] B. P. Carlin, N. G. Polson, and D. S. Stoffer, “A monte carlo approach to nonnormal and nonlinear state-space modeling,” *Journal of the American Statistical Association*, vol. 87, no. 418, pp. 493–500, June 1992.
- [16] M. H. Chen, Q. M. Shao, and J. G. Ibrahim, *Monte Carlo Methods in Bayesian Computation*. NY, USA: Springer-Verlag, 2000.
- [17] Z. Chen and H. J. Lee, “Knowledge-guided visual perception of 3d human gait from a single image sequence,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 336–342, 1992.
- [18] R. T. Collins, “Mean-shift blob tracking through scale space,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Madison, WI, USA, June 2003, pp. 234–240.

-
- [19] D. Comaniciu and P. Meer, “Meanshift: A robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.
- [20] J. Coughlan and S. J. Ferreira, “Finding deformable shapes using loopy belief propogation,” in *Proc. of European Conference on Computer Vision*, Copenhagen, Denmark, May 2002, pp. 453–468.
- [21] D. Cunado, M. S. Nixon, and J. N. Carter, “Automatic extraction and description of human gait models for recognition purposes,” *Computer Vision and Image Understanding*, vol. 90, pp. 1–41, 2003.
- [22] R. Cutler and L. Davis, “Real-time periodic motion detection, analysis, and applications,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Fort Collins, USA, June 1999, pp. 326–331.
- [23] D. Demirdjian, “Combining geometric- and view-based approaches for articulated pose estimation,” in *Proc. of European Conference on Computer Vision*, Prague, Czech Republic, May 2004, pp. 183–194.
- [24] J. Deutscher, A. Blake, and I. Reid, “Articulated body motion capture by annealed particle filtering,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, June 2000, pp. 126–133.

-
- [25] P. F. Felzenszwalb, “Representation and detection of deformable shapes,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Madison, WI, USA, June 2003, pp. 102–108.
- [26] M. M. Fleck, D. A. Forsyth, and C. Bregler, “Finding naked people,” in *Proc. of European Conference on Computer Vision*, 1996, pp. 593–602.
- [27] H. Fujiyoshi and A. J. Lipton, “Real-time human motion analysis by image skeletonization,” in *Proc. of IEEE Workshop on Applications of Computer Vision*, Princeton, New Jersey, Oct. 1998, pp. 15–21.
- [28] A. Galata, N. Johnson, and D. Hogg, “Learning variable-length markov models of behavior,” *Computer Vision and Image Understanding*, vol. 3, no. 81, pp. 398–413, Mar. 2001.
- [29] D. M. Gavrilu, “The visual analysis of human movement: a survey,” *Computer Vision and Image Understanding*, no. 1, pp. 82–98, Jan. 1999.
- [30] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall, 1996.
- [31] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
- [32] I. Haritaoglu, R. Cutler, D. Harwood, and L. S. Davis, “Backpack: Detection of people carrying objects using silhouettes,” in *Proc. of International Conference on Computer Vision*, Kerkyra, Greece, Sept. 1999, pp. 102–107.

-
- [33] I. Haritaoglu, D. Harwood, and L. S. Davis, “Ghost: A human body part labeling system using silhouettes,” in *Proc. of International Conference on Pattern Recognition*, Brisbane, Australia, Aug. 1998, pp. 77–82.
 - [34] —, “ w^4 : Who? when? where? what? a real time system for detecting and tracking people,” in *Proc. of International Conference on Face and Gesture Recognition*, 1998, pp. 222–227.
 - [35] B. Heisele and C. Wohler, “Motion-based recognition of pedestrians,” in *Proc. of International Conference on Pattern Recognition*, Brisbane, Australia, Aug. 1998, pp. 1325–1330.
 - [36] S. Hongeng and R. Nevatia, “Multi-agent event recognition,” in *Proc. of International Conference on Computer Vision*, Vancouver, BC, Canada, July 2001, pp. 84–91.
 - [37] G. Hua and Y. Wu, “Multi-scale visual tracking by sequential belief propagation,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, June 2004, pp. 826–833.
 - [38] G. Hua, M. Yang, and Y. Wu, “Learning to estimate human pose with data driven belief propagation,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, San Diego, June 2005, pp. 747–754.
 - [39] P. S. Huang, “Automatic gait recognition via statistical approaches for extended template features,” *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 31, no. 5, pp. 818–824, Oct. 2001.

-
- [40] S. Ioffe and D. Forsyth, “Finding people by sampling,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, 1999, pp. 1092–1097.
- [41] S. Ioffe and D. A. Forsyth, “Probabilistic methods for finding people,” *International Journal of Computer Vision*, vol. 43, no. 1, pp. 45–68, 2001.
- [42] M. Isard and A. Blake, “Condensation - conditional density propagation for visual tracking,” *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [43] Y. Iwai, K. Ogaki, and M. Yachida, “Posture estimation using structure and motion models,” in *Proc. of International Conference on Computer Vision*, Corfu, Greece, Sept. 1999, pp. 206–213.
- [44] S. Iwasawa, K. Ebihara, J. Ohya, and S. Morishima, “Real-time estimation of human body posture from monocular thermal images,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Puerto Rico, 1997, pp. 15–20.
- [45] A. Jochheim, M. Gerke, and A. Bischoff, “Modeling and simulation of kinematic systems,” Control Systems Engineering Group, University of Hagen, Hagen, Germany, TR, 1999.
- [46] G. Johansson, “Visual perception of biological motion and a model for its analysis,” *Perception and Psychophysics*, vol. 14, no. 2, pp. 210–211, 1973.

-
- [47] I. Kakadiaris and D. Metaxas, “Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, San Fran., CA, USA, June 1996, pp. 18–2.
- [48] S. Kirkpatrick, C. Gellatt, and M. Vecchi, “Optimisation by simulated annealing,” IBM Thomas J. Watson Research Centre, Yorktown Heights, NY, USA, TR, 1982.
- [49] D. Koller and R. Fratkina, “Using learning for approximation in stochastic processes,” in *Proc. of International Conference on Machine Learning*, 1998, pp. 287–295.
- [50] N. Krahnstoever and R. Sharma, “Articulated models from video,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, June 2004, pp. 894–901.
- [51] N. Krahnstoever, M. Yeasin, and R. Sharma, “Automatic acquisition and initialization of articulated models,” in *Machine Vision and Applications*, 2003, pp. 218–228.
- [52] B. Kwolek, “Stereovision-based head tracking using color and ellipse fitting in a particle filter,” in *Proc. of European Conference on Computer Vision*, Prague, Czech Republic, May 2004, pp. 192–204.

-
- [53] M. K. Lee, S. W. Leung, T. L. Pun, H. L. Cheung, and A. M. K. Lee, “Edge detection by genetic algorithm,” in *Proc. of International Conference on Image Processing*, Vancouver, BC, Canada, Sept. 2000, pp. 478–480.
- [54] M. W. Lee and I. Cohen, “Human upper body pose estimation in static images,” in *Proc. of European Conference on Computer Vision*, Prague, Czech Republic, May 2004, pp. 126–138.
- [55] —, “Proposal maps driven MCMC for estimating human body pose in static images,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, June 2004, pp. 334–341.
- [56] M. W. Lee, I. Cohen, and S. K. Jung, “Particle filter with analytical inference for human body tracking,” in *Proc. of IEEE Workshop on Motion and Video Computing*, Dec. 2002, pp. 159–165.
- [57] M. W. Lee and R. Nevatia, “Dynamic human pose estimation using markov chain monte carlo approach,” in *Proc. of the IEEE Workshop on Motion and Video Computing*, Washington, DC, USA, 2005, pp. 168–175.
- [58] R. Lienhart and J. Maydt, “An extended set of haar-like features for rapid object detection,” in *Proc. of International Conference on Image Processing*, Rochester, NY, USA, Sept. 2002, pp. 900–903.
- [59] A. J. Lipton, H. Fujiyoshi, and R. Patil, “Moving target classification and tracking from real-time video,” in *Proc. of the Workshop on Application of Computer Vision*, Oct. 1998, pp. 8–14.

-
- [60] F. Liu and R. W. Picard, “Detecting and segmenting periodic motion,” Media Lab Vision and Modelling, MIT, Cambridge, MA, TR 400, 1996.
- [61] W. Lu and Y. P. Tan, “A color histogram based people tracking system,” in *Proc. of International Symposium on Circuits and Systems*, Sydney, Australia, May 2001, pp. 137–140.
- [62] A. McIvor, Q. Zang, and R. Klette, “The background subtraction problem for video surveillance systems,” in *Proc. of International Workshop on Robot Vision*, 2001, pp. 176 – 183.
- [63] P. Meer and B. Georgescu, “Edge detection with embedded confidence,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 12, pp. 1351–1365, Dec. 2001.
- [64] D. Meyer, “Human gait classification based on hidden markov models,” in *3D Image Analysis and Synthesis’97*, 1997, pp. 139–146.
- [65] D. Meyer, J. Posl, and H. Niemann, “Gait classification with hmms for trajectories of body parts extracted by mixture densities,” in *Proc. of British Machine Vision Conference*, Southampton, U.K., Sept. 1998, pp. 459–468.
- [66] K. Mikolajczyk, C. Schmid, and A. Zisserman, “Human detection based on a probabilistic assembly of robust part detectors,” in *Proc. of European Conference on Computer Vision*, Prague, Czech Republic, May 2004, pp. 69–82.

-
- [67] D. D. Morris and J. M. Rehg, "Singularity analysis for articulated object tracking," in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, USA, June 1998, pp. 289–296.
- [68] S. A. Niyogi and E. H. Adelson, "Analyzing and recognizing walking figures in XYT," Media Lab Vision and Modelling, MIT, Cambridge, MA, TR 223, 1995.
- [69] M. Oren, C. Pagageorgiou, and P. Sinha, "Pedestrian detection using wavelet templates," in *Proc. of International Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997, pp. 193–199.
- [70] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.
- [71] V. Pavlović, J. M. Rehg, T. Cham, and K. P. Murphy, "A dynamic bayesian network approach to figure tracking using learned dynamic models," in *Proc. of International Conference on Computer Vision*, Corfu, Greece, Sept. 1999, pp. 94–101.
- [72] J. Posl and H. Niemann, "Statistical 3-d object localization without segmentation using wavelet analysis," in *Proc. of International Conference on Computer Analysis of Images and Patterns*, Kiel, Germany, Sept. 1997, pp. 440–447.
- [73] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

-
- [74] D. Ramanan and D. A. Forsyth, “Finding and tracking people from the bottom up,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Madison, WI, USA, June 2003, pp. 467–474.
- [75] D. Ramanan, D. A. Forsyth, and A. Zisserman, “Strike a pose: tracking people by finding stylized poses,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, June 2005, pp. 271–278.
- [76] R. F. Rashid, “Towards a system for the interpretation of moving light displays,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, no. 6, pp. 574–581, Nov. 1980.
- [77] T. J. Roberts, S. J. McKenna, and I. W. Ricketts, “Human pose estimation using learnt probabilistic region similarities and partial configurations,” in *Proc. of European Conference on Computer Vision*, Prague, Czech Republic, May 2004, pp. 291–303.
- [78] K. Rohr, “Towards model-based recognition of human movements in image sequences,” *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 59, no. 1, pp. 94–115, Jan. 1994.
- [79] R. J. Schalkoff, *Digital Image Processing and Computer Vision*. NY, USA: John Wiley & Sons, 1989.
- [80] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, “Tracking loose-limbed people,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, June 2004, pp. 421–428.

-
- [81] C. Sminchisescu and B. Triggs, “Covariance scaled sampling for monocular 3d body tracking,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, USA, Dec. 2001, pp. 447–454.
- [82] C. Stauffer and W. Grimson, “Learning patterns of activity using real-time tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 747–757, Aug. 2000.
- [83] S. Thrun and J. Langford, “Monte carlo hidden markov models,” School of Computer Science, CMU, Pittsburgh, USA, TR 179, 1998.
- [84] —, “Monte carlo hidden markov models: Learning non-parametric models of partially observable stochastic processes,” in *Proc. of International Conference on Machine Learning*, San Francisco, CA, USA, 1999, pp. 415–424.
- [85] P. S. Tsai, M. Shah, K. Keiter, and T. Kasparis, “Cyclic motion detection for motion based recognition,” *Pattern Recognition*, vol. 27, no. 12, pp. 1591–1603, 1994.
- [86] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Kauai, HI, USA, Dec. 2001, pp. 511–518.
- [87] S. Wachter and H. H. Nagel, “Tracking of persons in monocular image sequences,” in *Proc. of the Workshop on Motion of Non-Rigid and Articulated Objects*, 1997, pp. 2–9.

-
- [88] T. Wada and T. Matsuyama, “Multiobject behavior recognition by event driven selective attention method,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 873–887, Aug. 2000.
- [89] L. Wang, H. Z. Ning, T. N. Tan, and W. M. Hu, “Fusion of static and dynamic body biometrics for gait recognition,” in *Proc. of International Conference on Computer Vision*, Nice, France, Oct. 2003, pp. 1449–1454.
- [90] L. Wang and T. Tan, “Automatic gait recognition based on statistical shape analysis,” *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1120–1131, Sept. 2003.
- [91] C. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-time tracking of the human body,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, July 1997.
- [92] M. Yamada, K. Ebihara, and J. Ohya, “A new robust real-time method for extracting human silhouettes from color images,” in *Proc. of International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, Apr. 1998, pp. 528–533.
- [93] J. Y. Zhang, R. Collins, and Y. X. Liu, “Representation and matching of articulated shapes,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, June 2004, pp. 342–349.

-
- [94] T. Zhao and R. Nevatia, “Bayesian human segmentation in crowded situations,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Madison, WI, USA, June 2003, pp. 459–466.
- [95] S. Zhu, R. Zhang, and Z. Tu, “Integrating bottom-up/top-down for object recognition by data driven markov chain monte carlo,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, June 2000, pp. 738–745.
- [96] Z. Zivkovic and B. Krose, “An em-like algorithm for color-histogram-based object tracking,” in *Proc. of International Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, June 2004, pp. 798–803.