Model-based Segmentation and Tracking of Multiple Humans in Complex Situations

by

Tao Zhao

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

December 2003

Copyright 2003 Tao Zhao
Dedication

To my family.
Acknowledgements

First I would like express my deepest gratitude to my advisor Prof. Ram Nevatia. Without him this thesis work is impossible in many ways. I benefit a lot from his insightful guidance, inspiring support and invaluable knowledge. I would also like to thank Prof. Gerard Medioni and Prof. Isaac Cohen for advices and encouragements on my work, and Prof. C.-C. Jay Kuo and Prof. Laurent Itti for taking their precious time to serve on committees of my qualifying exam or/and thesis defense. My 5 years' PhD study (1998-2003) has been made not only a great learning experience but also enjoyable by the interactions with other IRIS members (special thanks to Fengjun Lv, Munwai Lee, Jinman Kang who work closely) and other USC friends (Bolan Jiang, Junmei Zhu, Yunsong Huang, among many others) in this period.

My internship (with Dr. Harry Shum) in Microsoft Research Beijing (now Asia) in summer 2000 had very positive influence on my research thereafter. Many thanks to Prof. Song Chun Zhu for the introduction to DDMCMC, and to Dr. Zhuowen Tu for discussions and helps.

I am also grateful to the unconditioned support and understanding from my lovely wife Jian Wu and my family back in China. The pursuit of PhD study is partially a result of my parents’ cultivation of my interests in science and encouragements for advanced education since my childhood. I feel sorry for not being able to take care of my parents
when they are in need, and feel indebted to my sister and brother-in-law who have been taking my responsibilities.
Contents

Dedication ii
Acknowledgements iii
List Of Tables ix
List Of Figures x
Abstract xiii

1 Introduction and Motivation 1
  1.1 Background 1
  1.2 Goals 2
  1.3 General Difficulties 3
  1.4 Introduction to Our Approach 5
  1.5 Reader’s Guide 7

2 Previous Work 9
  2.1 Background Modeling and Change Detection 9
  2.2 Object Tracking 11
    2.2.1 Detection-based tracking 12
    2.2.2 Blob tracking 13
    2.2.3 Matching-based tracking 14
    2.2.4 Tracking multiple humans 16
    2.2.5 Articulated body tracking 17

3 An Overview 19
  3.1 Model-based approaches 19
    3.1.1 Blob v.s. object model 20
    3.1.2 A simple solution to the model-based segmentation and tracking 21
    3.1.3 A Bayesian approach to the model-based segmentation and tracking 21
    3.1.4 A tracking as recognition approach to human body posture estimation 23
  3.2 Modeling Image Formation 23
    3.2.1 Camera modeling 23
    3.2.2 Background modeling 24
    3.2.3 Human modeling 25
3.2.4 Shadow modeling ................................................. 25
3.3 Camera Modeling and Geometric Constraints ....................... 26
3.3.1 Sensor deployment for visual surveillance ...................... 26
3.3.2 Camera calibration ............................................. 28
3.3.3 The Transformations ........................................... 29

4 Simple Model-based Segmentation and Tracking .......................... 33
4.1 Introduction ..................................................... 33
4.1.1 Human model ............................................... 34
4.1.2 Notation ...................................................... 35
4.2 Segmenting Multiple Humans ........................................... 36
4.2.1 Head candidates by boundary/shape analysis ...................... 36
4.2.2 Geometrical shadow analysis .................................. 37
4.2.3 The algorithm ............................................... 38
4.3 Tracking Multiple Humans ........................................... 40
4.3.1 Object representation for tracking ................................. 40
4.3.2 Computing the best match ..................................... 43
4.3.3 Estimation with Kalman filter .................................. 44
4.3.4 Object updating ............................................. 46
4.3.5 A summary .................................................. 46
4.4 Implementation and Results ........................................... 47
4.4.1 Entrances and exits ........................................... 47
4.4.2 Results on two examples ....................................... 48
4.4.3 Performance evaluation ....................................... 49
4.4.4 Remarks .................................................... 56

5 Model-based Segmentation: a Bayesian Approach ......................... 59
5.1 Motivation and Introduction ......................................... 60
5.1.1 Motivation ................................................... 60
5.1.2 Introduction .................................................. 61
5.2 3D Human Shape Models ............................................. 62
5.2.1 2D models v.s. 3D models ...................................... 62
5.2.2 Multi-ellipsoid shape model ................................... 64
5.3 A Bayesian Formulation of Model-based Segmentation Problem ......... 67
5.3.1 The prior probabilities ......................................... 69
5.3.2 The image likelihood ......................................... 70
5.3.2.1 Choosing a likelihood model ................................ 71
5.3.2.2 Mapping pixel sizes into 3D ................................. 72
5.3.2.3 Foreground/background (F/B) likelihood model .......... 73
5.3.2.4 Color likelihood model ................................... 76
5.4 Computing the MAP Solution by Efficient MCMC ...................... 78
5.4.1 Introduction of MCMC and its applications in computer vision .... 80
5.4.1.1 Markov chain Monte Carlo method to sample a probability distribution ........................................... 80
5.4.1.2 The use of MCMC in computer vision ...................... 81
5.4.2 Markov chain dynamics to efficiently traverse the solution space 82
5.4.2.1 Human hypothesis addition ................................. 83
5.4.2.2 Human hypothesis removal ................................. 84
5.4.2.3 Human hypothesis split .................................... 84
5.4.2.4 Human hypotheses merge ................................. 84
5.4.2.5 Model/orientation switch ................................. 85
5.4.2.6 Stochastic diffusion of model parameters ............... 85
5.4.3 Informed proposal probabilities ............................. 86
5.4.3.1 Object addition: Head candidates from foreground bound-
aries ................................................................. 86
5.4.3.2 Object addition: Head candidates from intensity edges . 87
5.4.3.3 Object addition: Connected component analysis of fore-
ground residue ................................................... 89
5.4.3.4 Object addition: Projection analysis of foreground residue 90
5.4.3.5 Object split .............................................. 92
5.4.3.6 Object merge ........................................... 94
5.4.3.7 Object parameters diffusion .............................. 94
5.4.4 Summary of the algorithm .................................... 97
5.5 Implementation and Results .................................... 98
5.5.1 Incremental computation ..................................... 98
5.5.2 Results ..................................................... 100
5.5.3 Performance evaluation ....................................... 103
5.5.4 Computation ............................................... 105
5.5.5 Remarks ................................................... 107
5.5.5.1 Comparison with the simple segmentation in Sec.4.2 .... 107
5.5.5.2 A unifying framework integrating top-down and bottom-
up processing .................................................. 108

6 Model-based Tracking: a Bayesian Approach 115
6.1 Introduction .................................................. 115
6.2 The Posterior Probability for Multi-object Tracking .......... 117
6.2.1 The temporal prior ........................................ 118
6.2.2 Color-based likelihood .................................... 119
6.3 Computing the MAP with Uncertainty ......................... 121
6.3.1 Augmentations to Markov chain dynamics ................. 121
6.3.1.1 Establish correspondence .............................. 122
6.3.1.2 Break correspondence .................................. 122
6.3.1.3 Identities switch ...................................... 123
6.3.2 Augmentations to proposal probabilities ................. 123
6.3.2.1 Object addition: from estimate of previous frame ... 123
6.3.2.2 Proposing position update ............................ 124
6.3.3 Temporal filtering with adaptive measurement noise .... 124
6.4 Implementation and Results .................................. 125
6.4.1 Remarks ................................................... 127
6.4.1.1 Comparison with tracking approach in Sec.4.3 ....... 127
7 3D Tracking of Human Locomotion: a Tracking as Recognition Approach 132
7.1 Introduction and Motivation .................................. 132
7.2 The Locomotion Model ........................................ 134
7.3 The Observations ............................................... 136
7.4 Inferences in the Model ....................................... 138
7.4.1 Optional post-processing .................................. 141
7.5 Implementation and Results ................................... 142
7.5.1 Results on locomotion ..................................... 142
7.5.2 Human verification by walking ......................... 143
7.5.3 Remarks ..................................................... 146

8 Conclusion 148
8.1 Summary of Contributions ................................. 149
8.2 Future Directions ............................................... 150

Reference List 152

Appendix A
Camera calibration from vanishing points/line .................. 161

Appendix B
A blob tracker ..................................................... 163

Appendix C
Image projection of multi-ellipsoid human model ............. 165

Appendix D
Object detection/tracking using mean shift ...................... 167
D.1 Object attraction using mean shift .......................... 167
D.2 Background exclusion using mean shift ..................... 171
D.3 Combining object attraction and background exclusion .... 173
List Of Tables

4.1 The performance evaluation of the segmentation and tracking system . . . 57

5.1 Results of performance evaluations on “Topping” and “Commons” . . . . 104
List Of Figures

1.1 The goals of the work in this thesis. .......................... 3
1.2 Some of the difficulties to achieve the goals marked on example frames. 4
1.3 The overall diagram of our approaches. ......................... 6
3.1 The camera and the world: relationship of various geometric entities. 29
4.1 Example images from our dataset and the enlarged patches of foreground. 34
4.2 System diagram of simple model-based segmentation and tracking. .... 35
4.3 Some techniques used in human segmentation. ..................... 40
4.4 The process of human segmentation on an example. ............... 41
4.5 The algorithm of multi-human segmentation. ....................... 42
4.6 Examples of object representation for tracking and its evolution. .... 42
4.7 The algorithm of multi-object tracking. .......................... 47
4.8 Selected frames of human global motion tracking result of seq1. .... 51
4.9 Selected frames of human global motion tracking result of seq2. .... 53
4.10 Snapshots of some of the sequences that we used for performance evaluation. 53
5.1 An example of human segmentation. ............................ 59
5.2 Multi-ellipsoid models for standing and walking humans. .......... 66
5.3 The computation of a “pseudo” 3D size of a pixel. ................ 73
5.4 The relationships between the different regions. 73
5.5 The color likelihood model. 78
5.6 The algorithms to compute the likelihood values. 79
5.7 The dynamics for multi-object segmenation. 83
5.8 Head candidate detectors. 88
5.9 Generating human hypothesis using foreground residue projection analysis. 92
5.10 Splitting an object into two. 93
5.11 The block diagram of MCMC model-based segmentation. 97
5.12 The incremental computation of the F/B likelihood. 99
5.13 The incremental computation of the color likelihood. 100
5.14 Segmentation results on sequence seq1 used in Chap.4. 109
5.15 Segmentation results of sequence “Topping”. 110
5.16 Segmentation results of sequence “Commons”. 111
5.17 Segmentation results of an indoor sequence. 112
5.18 Segmentation results of a noisy infra-red airborne video sequence. 113
5.19 Experiments on convergence. 113
5.20 The histogram of the number of human objects per blob. 114
6.1 The block diagram for MCMC model-based tracking. 117
6.2 Identity-related dynamics for multi-object tracking. 121
6.3 Selected frames of the tracking results from “Commons”. 129
7.1 The human locomotion model. 135
7.2 Phases of walking and running. 135
7.3 Computing model motion template. 139
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.4 Result of estimating locomotion modes and phases.</td>
<td>144</td>
</tr>
<tr>
<td>7.5 Selected frames of human verification examples.</td>
<td>147</td>
</tr>
<tr>
<td>A.1 Camera calibration using vanishing points/line.</td>
<td>162</td>
</tr>
<tr>
<td>D.1 Comparing mean shift tracker and background exclusion tracker.</td>
<td>173</td>
</tr>
<tr>
<td>D.2 The combined tracker.</td>
<td>174</td>
</tr>
</tbody>
</table>
Abstract

Automatic detecting and tracking people from a stationary video camera is important for many applications. The problems are made difficult due to versatile human shape and appearance, persistent or temporary occlusion of multiple people and noise from various sources (e.g., shadow), which are commonplace in reality. We propose to tackle the challenges using applicable and general constraints in the form of models. In particular, we use a background appearance model and a camera model. We also use explicit human shape models as an entity for analysis both in segmentation and in global motion tracking, and use a 3D locomotion model to assist the estimation of articulated body postures.

We present two approaches towards the goal of multi-human segmentation and global motion tracking. In the first approach, a simple shape model is used. The segmentation is done using direct image features. Multiple overlapping humans tracking is factored into tracking each one according to their depth order. The second approach follows a Bayesian framework. The optimal solutions for segmentation and tracking are defined explicitly as a Bayesian posterior probability in the joint-object space. The solution is computed by a Markov chain Monte Carlo-based method. The computation also takes advantages of domain knowledge as importance proposal probabilities to direct the Markov chain intelligently to obtain significantly faster convergence. The approach is more general and applies to the scenario where a large group of people has persistent occlusion. This
approach has both the robustness and the optimality of the Bayesian formulation and the computational efficiency from the bottom-up processing. Although only applied to human segmentation and tracking, the proposed approach can be extended to general multi-object segmentation and tracking.

We propose a *tracking as recognition* approach where the estimation of body postures is accomplished by recognizing the motion in a locomotion model. It results in robust performance in low-resolution data, with temporary occlusion and without interactive initialization.
Chapter 1

Introduction and Motivation

1.1 Background

Over the last few decades, the rapid rise of information technology (IT) industry has resulted in significant and continuous increase in the capabilities of the computational hardware with low cost. The growth of the computational power is well characterized by the famous Moore’s law for almost 40 years. The increase of the capacities of storage devices: electronic storage (memory), magnetic storage and optical storage follows. The advances of sensor technology have also made CCD (charged couple device) cameras highly available. And the advances in network transmission and high bandwidth digital interface (e.g., USB, Fireware) make setting up a camera-computer system more convenient. All this has opened possibilities for automatic video understanding, which needs to manipulate huge amount of possibly real-time data, in the hardware aspect.

The ultimate goal of video understanding is to make structured decomposition of video into the scene, describe the objects and their time-varying properties, and to extract semantic meaning from them. Humans, one of the many objects in the world, are of special
significance since they are the main class of actors of daily life activities of interest. Being able to detect human objects and track their motion in video sequences is not only useful by itself but also a crucial step to higher-level analysis.

This technology can enable many applications which will provide information, convenience and security to our lives. The detection and tracking by itself can be used directly for intrusion detection, human counting, and estimation of crowd flow. Its further applications include automatic video surveillance which facilitate fighting against crime, providing higher level of security; advanced human computer interaction for computers to interact with humans more easily and naturally; understanding humans’ behaviors for better assistance in various environment (e.g., home, business, etc); content-based video indexing which makes more efficient access of the information from huge volume of video data.

1.2 Goals

The goal of our work is to detect\(^1\) and track multiple people in complex situations with a stationary video camera. More specifically:

- To segment/detect multiple possibly inter-occluded human objects in the image.

The segmentation answers the questions of how many people are in the scene and where they are.

\(^1\)Since we do detection by either segmenting the foreground from change detection into a number of human objects, or segmenting the entire image into a number of human objects and a background layer, we use the terms detection and segmentation interchangeably in this thesis.
Figure 1.1: The goals of the work in this thesis. 1. To segment multiple possibly occluding humans in the image; 2. To obtain consistent trajectories of multiple possibly occluding humans in the video sequence; 3. To estimate the 3D body posture of each human.

- To reliably track the global motion (i.e., position) of multiple possibly inter-occluded human objects in the scene and provide consistent 3D trajectories.

- To estimate the motion modes (i.e., walking, running, standing) and phases (i.e., the characteristic 3D human body postures) when the main limbs are largely visible.

1.3 General Difficulties

We summarize the main difficulties to achieve the goal below. Some of them are illustrated in the examples shown in Fig.1.2.

**Environment modeling** In the case of a stationary camera, the environment can be simplified into a 2D background appearance model, which can efficiently direct the attention to image regions different to the background, through change detection. However, the changed region may include regions which do not correspond to moving objects, for example shadow, illumination change, non-stationary objects in the environments. It may not include the complete regions corresponding to objects, if they have similar appearance as the background behind them.
Figure 1.2: Some of the difficulties to achieve the goals marked on example frames. 1. non-stationary objects in the background; 2. sun cast shadow of moving objects; 3. occlusion by scene objects; 4. objects inter-occlusion; 5. persistent inter-occlusion involving large number of objects; 6. change of object appearance (from the sun into the shadow); 7. low contrast of the foreground objects with the background.

**Object modeling** Extracting human objects from video sequences in complex situation requires the model of human. Human body is both highly articulated and deformable and capable of complex motion, which requires a large number of parameters to model. The camera projection adds perspective effect on human shape. Furthermore, the clothing of human is highly versatile with many possible colors, textures and styles.

**Occlusion** The objects may be occluded by static objects in the scene which stands between the camera and the objects. When there are multiple people in the scene and some of them may occlude others due to their spatial proximity or camera projection. When people move in a group, the occlusion is often persistent. The occlusion causes missing observations or observations produced collectively by multiple overlapping objects, which increases the difficulty of analysis.
1.4 Introduction to Our Approach

To seek robust solution in the presence of the above difficulties, we try to incorporate as many applicable and general constraints as possible. The constraints are in the form of models.

In particular, we assume a background model is available, as in most stationary camera systems, which can effectively direct the attention to the image regions which are different to the background, possibly caused by moving objects in the scene. Different from most of the previous work on change detection-based tracking, we propose to use explicit human shape model as an entity for analysis both in segmentation and in global motion tracking. The model enables making meaningful interpretation of image observations especially when low-level processing (e.g., change detection) show ambiguity. We assume that the camera model is available (and provide a way to obtain it conveniently); we assume that the human objects move on a known ground plane, which can be relaxed to any site whose geometry is known. This enables us to transform between quantities in the image and the quantities in the 3D world which provides view-independent descriptions. The camera model combined with the 3D human shape models, provide constraints on the valid human shape in the image. Although general human motion is quite flexible, human locomotion shows strong regularity. We use the locomotion model to assist the estimation of articulated body postures. A high-level diagram of our overall approach is depicted in Fig.1.3.

First we will describe a simple system for segmentation and tracking using a single ellipsoid model. The human objects are hypothesized using model-based boundary and
Figure 1.3: The overall diagram of our approaches.

shape analysis of the foreground regions when they first enter the scene. The segmented objects are tracked using their appearance in the subsequent frames. Multiple overlapping humans tracking is factored into matching them one by one according to their depth order, inferred from the camera model. This technique results in a real-time system which is not sensitive to the fragmentation of the result of change detection, and is effective for temporary occlusion and persistent occlusion of small groups of people.

The simple approach may not be effective when the number of people and the amount of occlusion between them increases. Motivated by these deficiencies, we then present a more principled approach on model-based segmentation and tracking under the Bayesian framework. The optimal solutions for segmentation and tracking are defined explicitly as the Bayesian posterior probability in the joint-object space. The optimal solution in the complex high dimensional space is computed by a Markov chain Monte Carlo (MCMC)-based method, which is otherwise difficult to obtain with other methods (e.g., standard
search or optimization techniques). The computational approach also takes advantages of domain knowledge as importance proposal probabilities to direct the Markov chain intelligently to obtain significantly faster convergence. The new formulation is more general and also applies to the case of a larger group of people moving together (e.g. Fig1.1.1(2)), for which we have not seen similar results reported before. This approach is an integration of bottom-up and top-down techniques. It has both the robustness and the optimality of the Bayesian formulation and the computational efficiency provided from the domain knowledge which is not required to be complete. Although only applied to human segmentation and tracking in this thesis, the proposed approach can be extended to general multi-object segmentation and tracking (e.g. vehicles or a mixture of human and vehicle).

We also estimate the locomotion modes (i.e., walking, running and standing) and the gross body posture. We propose a tracking as recognition approach where the estimation of body postures is accomplished by recognizing the motion in a locomotion model. It results in robust performance in low-resolution data, with temporary occlusion, without the need for interactive initialization. The method is also used to verify the tracked human hypotheses by their motion patterns. The proposed tracking as recognition approach is also a general formulation and we expect it to apply for other classes of motion.

1.5 Reader’s Guide

In the rest of the thesis, we will first summarize pervious work related to different aspect of our work in Chap.2. The overview of our approaches as well as their rationales are
stated in Chap. 3. In Chap. 4 we present a simple attack to the multi-human segmentation and tracking problem and discuss its drawbacks. In Chap. 5 we present the Bayesian formulation for model-based segmentation and a Markov chain Monte Carlo based computational engine to pursue the optimal solution. In Chap. 6 we extend the formulation to tracking. In Chap. 7 we present an approach to estimate the locomotion modes and gross body posture when the limbs are visible in most of the time. We summarize our contributions and discuss the future research directions in Chap. 8. Due to the nature of the work, selected image frames are sometimes insufficient to show the results well due to both the space limit and the lack of motion perception. We encourage the readers to view the video files associated with the major results on the accompanying CD which are also online at http://iris.usc.edu/~taozhao/papers/thesis/.
Chapter 2

Previous Work

The topics of motion detection, human tracking and human motion analysis have been addressed in numerous research papers. We roughly group the major related work in the following categories and discuss their advantages and potential deficiencies related to the problems that we aim to solve. There are possible overlaps in different categories. We will discuss some of them in more detail or refer to other related works in the rest of this thesis when necessary.

2.1 Background Modeling and Change Detection

In the context of a stationary camera, background modeling and change detection is a classical [45] and widely-used technique. A background model is constructed (trained) at each pixel location. Each pixel in a new frame is classified into foreground or background according to its distance to the background model. Wren et al. [117] used a Gaussian model for each pixel which is simple and sufficiently accurate for many scenarios. Stauffer et al. [105] extended it to a mixture of Gaussians to handle the situations where the value at each pixel switches between multiple processes such as ripples or tree branches under
strong sunlight. A median filter was used by Haritaoglu et al. [39] which also allows some moving objects in the training phase. Elgammal et al. [28] used non-parametric technique to estimate the density of the distribution which removes the problem to determine the number of mixtures in a multi-Gaussian model. Rittscher et al. [93] proposed a probabilistic background model based on an HMM (hidden Markov model) which discriminates background, foreground and shadow in a highway environment. The models are usually continuous updated by the incoming frames to adapt the slow change of the scene.

Sudden global illumination changes (e.g., by a plane flying by) or local illumination changes (e.g., shadows by moving objects) also result in large difference from the background model. Some approaches try to differentiate the changes caused by objects and by illumination changes. A review is given in [88]. Most of them are based on the assumption that the illumination does not change the hue of the background. Analysis of hue (usually the H component in the HSV color model) is highly unreliable when the intensity is either too light or too dark. The analysis is only effective if the dynamic range of the intensity of the scene is small, which rules out environments lit by the strong sun light. The work of Jawed et al. [49] tries to solve the illumination problem with the help of gradient information. It is based on the observation that (soft) shadow regions usually do not have a high gradient boundary.

The classified foreground pixels are usually filtered with a median filter or/and a morphological filter for noise reduction and to obtain smooth boundaries. Connected components are then computed on the foreground mask. Each connected component is called a foreground blob or simply blob in this thesis.
Different in philosophy from the above technique, Sullivan et al. proposed a method to model general foreground and background in [108]. The image is divided into a number of regular grids and the statistics of some filter responses are learnt for both foreground and background. Since it is a general background model, it also applies for non-stationary cameras. As whether an object belongs to foreground or background is not fixed (e.g., a moving car v.s. a parked car), the validity of this approach is questionable. The model is also used by Isard et al. in [48].

In case of a moving camera, camera ego motion can be first compensated using a parametric motion model, which is effective for a rotating camera, a distant scene or a planer scene (e.g., [54] [86], etc). Then the change detection techniques can be applied.

Change detection and the related techniques gained popularity due to their simplicity and computational efficiency. However, the following problems should be addressed for a system based on change detection. 1) problems due to sudden change of illumination or shadows; 2) fragmentation problems due to low contrast of the moving object and the background behind it; and 3) problems due to the overlapping of multiple objects who will form a single blob, if object-level description is expected.

2.2 Object Tracking

The early works on tracking were mainly motivated by radar signal tracking, where the point like targets against a dark background could be easily segmented [1]. However, for many vision-based tracking problems, the objects are difficult to detect (indeed sometimes tracking is to ease detection in a sequence by taking advantage of the locality of
the object’s state); the measurements of states are obtained by finding the best match according to some similarity function in the neighborhood of the previous state. We call the two classes of approaches detection-based and matching-based tracking respectively.

2.2.1 Detection-based tracking

In radar-based tracking, an object (e.g., an aircraft, a ballistic missile, etc) results in a point like bright spot against a dark background. It can be easily detected after some simple operations. The tracking problem mainly concerns optimal estimation of the object states. A Kalman filter [52] suffices to provide optimal estimation in a linear system with Gaussian noise. Extended Kalman filter applies to non-linear systems by linearizing the system equation at each point. Clutter may cause more than one detections. A PDAF (probabilistic data-association filter) [1] makes probabilistic association of the object with the multiple detection. This technique gives an averaged estimate and breaks down when there is persistent distraction.

Instead of making probabilistic data association, the multiple-hypothesis tracking (MHT) (Reid [92]) enumerates all the possible assignments which will result in an exponential increase of the possibilities. An efficient implementation of the MHT is given by Cox et al. in [20], but it is still more expensive than PDAF.

In case of multiple objects, a JPDAF (joint probabilistic data-association filter) [1] can be used. It is similar to PDAF except that the objects to measurements association probabilities enforce the constraint that multiple objects should not lock on the same measurement. MHT technique also applies for multi-object tracking.
2.2.2 Blob tracking

Vision-based tracking of blobs after change detection\(^1\) resembles the radar-tracking problem. However it is different in the following aspects: 1) the objects (e.g. humans, vehicles) are larger (compared to the view) and closer to each other, move slower and are much less predictable than those in radar-tracking; 2) The detection of a single object can be fragmented; 3) When objects are close to each other, they may only result in a single detection. The last two violate the assumptions on data association in the radar-tracking techniques.

Compared to the objects’ size, its motion in adjacent frames is usually small (assuming a reasonable temporal sampling rate). Therefore, a simple blob tracker can generate satisfactory results just by using a nearest-neighbor rule for correspondence. This works in the ideal situations where objects are isolated and with no fragmentation. Hongeng et al. [44] used ground plane constraint to merge blobs which belong to the same object, however it does not address splitting. For more complex cases, different techniques based on perceptual grouping are used. Cohen et al. [16] extract tracks by representing the blobs in all the frames as a graph. However the split/merge problem is not solved. Komprobst et al. [60] use an approach based on tensor voting (Medioni et al. [71]) by grouping the blobs of all frames into tracks enforcing spatial/temporal smoothness; it can connect some broken tracks, remove noise and resolve some frequent splits/merges. Marques et al. [69] try to group the detected strokes (fragments of tracks) into consistent tracks using a Bayesian network. The problem with this class of techniques is that since no

\(^1\)Some other detection techniques, e.g., skin color detection, also fall in this category.
object shape model is involved, the split/merge due to fragmentation and due to multiple objects interactions may not be differentiated.

2.2.3 Matching-based tracking

In many vision-based tracking applications, the detection is not available as above. Instead, tracking is driven by the attraction basin of a defined similarity (likelihood) function of some object representation with image features. Comaniciu et al. [19] and Perez et al. [81] used color histogram as measurement to track an image region (non-rigid object); Koller et al. [58] used edges as measurement to track vehicles; Cham et al. [13] used a scaled image template as measurement to track human limbs; Isard et al. [47] used SNAKE [55] to track object contours with subspace constraint. The choice of feature in defining the similarity function usually depends on what is recognized as stable and discriminative against other part of the image. Multiple cues can be integrated to have more discriminative power.

If the surrounding of an object has similar features, the object is likely to be distracted and locked on the wrong place. This problem is vital to any tracking system, especially those tracking one of many similar objects. Being robust to temporary or partial occlusion is also desirable for a tracking system.

In a computational worst case, a search is done in the neighborhood of the state of the previous frame to locate the maximum, which restricts the number of parameters to be small (2 or 3 at most). If applicable, gradient information is used for locating the maximum more efficiently, e.g., in [58]. Mean-shift is a technique for mode seeking in a non-parametric distribution and has been used in [19] for efficient color tracking.
Particle filter based tracking (e.g., the Condensation algorithm by Isard et al. [47]) has become popular recently. It uses a number of samples to represent the posterior distribution of the estimate of the state instead of a single value (or a single-moded Gaussian as in the Kalman filter case), which can increase the robustness of tracking in case of short-term ambiguity. It can be regarded as a multiple hypotheses technique though the original MHT techniques are difficult to apply in matching-based tracking due to the unavailability of “detections” [13]. However it suffers from problems of high dimensionality of the state space, as non-parametric techniques do not scale well with dimension.

There is a class of tracking works which are driven only by motion (i.e., optical flow) without matching. The motion of the object is usually regularized using a parametric model. An example is Bregler et al. [9] for tracking articulated body. Because the object representation only depends on the current state, this type of technique usually suffers from drift problem. Other techniques whose object representations for matching partially depend on the current state (e.g., updated by observation in each frame) also suffer from this problem.

Tracking multiple isolated object bears little difference to single object tracking. However, if multiple objects becomes overlapped in the image, the image observations are caused by all the overlapping objects collectively. In other word, the image likelihood should be conditioned on a joint state which includes the states of all objects and their depth orders (if not already in the states) [1] [90]. The joint state has a dimension as the sum of the dimension of all the objects’ states, which makes the matching less efficient. Joint likelihood models have been used by Tao et al. [109], Isard et al. [48] Rasmussen
et al. [90]. As a way to trade optimum for efficiency, Koller et al. [59] match multiple cars and Regh et al. match multiple fingers one by one according to depth orders inferred from geometry, with the image regions claimed by the already matched objects masked out. In contour tracking, an exclusion principle was used by MacCormick et al. [68] to ensure one measurement is not caused by multiple objects.

In matching-based tracking, the initial state needs to be available either from manual interaction or from a separate object detection module. Compared with detection-based tracking, matching-based tracking is more general. Even in radar tracking, “tracking before detection” techniques are used for weak signals and close-by objects [4] which shows superior results than traditional detection-based tracking techniques.

We also want to mention the work of Jojic et al. [50] in decomposing a video sequence into a background layer and a number of translating object layers using variational method. However, it needs to know the number of objects and operates in a batch mode.

2.2.4 Tracking multiple humans

A significant amount of object tracking work is focused on human tracking in the same context as this thesis. Some explore the human specific domain knowledge which could help counter the difficulties. Haritaoglu et al. [39] used peaks of the vertical projection of foreground blobs to segment multiple humans in one blob. This only applies when humans are not vertically distributed in the image. Zhao et al. [119] and Siebel et al. [98] recognize head top on the boundary of the foreground regions to help segment overlapped humans. Some works assume that humans are isolated when they enter the scene, thus a
model (color model in [70] [29], boundary model in [39] [98]) can be initialized for further segmentation when they occlude each other.

Particle filter has also been applied to tracking multiple humans in [109] and [48] with a joint likelihood formulation. They both employ explicit human shape model similar to the work in this thesis. However, results were only shown on up to four people and dimensionality is known to be a curse to particle filters.

Mittal et al. [72] used multiple (up to 16) cameras to segment multiple humans on a ground plane using region-based stereo. Yang et al. [118] count and track people using multiple cameras by computing the visual hull.

2.2.5 Articulated body tracking

Tracking articulated object has received much attention recently for video-based motion capture. A recent survey is available by Moeslund et al. [74] which summarizes over 130 papers on this problem. There exist many possible ambiguities in tracking in the high dimensional (20-30 dimension) state space using noisy 2D measurements. The ambiguities can be reduced using multiple views. Deutscher et al [25] used particle filter with annealing and state partition technique to track relatively complex human against dark background with 3 cameras. With a single view, Sminchisescu et al [100] showed good result, though on simple motion, with all available constraints on joints, carefully designed robust likelihood models and techniques to exploit the multiple-mode structure of the solution space. Most successful systems require high image resolution and heavy computation.
Another way to resolve the ambiguities is to use motion model. The use of constrained motion models can reduce the search space; but it only works on the type of motion defined in the model. Rohr [94] describes pioneering work on motion recognition using medical motion data. In each frame, the joint angle values are searched on the motion curves of a walking cycle. Motion subspace is used in Sidenbladh et al. [99] to track human walking using a particle filter. Bregler et al. [8] use HMMs (hidden Markov models) to recognize human motion (e.g., running), but the recognition is separated from tracking. Brand [6] maps 2D shadows into 3D body poses by inference in an HMM learnt from 3D motion captured data. In Krahnstover et al. [87], human tracking is treated as an inference problem in an HMM; however, this approach is appearance-based and works well only for the viewpoints for which the system was trained.
Chapter 3

An Overview

In this chapter, we first present the architecture of the work in this thesis, justify the use of object shape model and give brief overviews of the approaches of the different components. We then briefly introduce the various models of image formation and last describe in detail camera model and geometric constraints.

3.1 Model-based approaches

The general architecture of our work is depicted in Fig. 1.3 in the first chapter. The global motion of the multiple objects is obtained by the interacting processes of segmentation and tracking with the help of background appearance model, object shape model with the constraints provided by camera model and the ground plane assumption. We provide two approaches for this part. After the global motion trajectories are obtained, human body postures are estimated with the help of an articulated motion model.
3.1.1 Blob v.s. object model

Many common approaches to object tracking with a stationary camera operate on the blob-based representations, perhaps encouraged by its success in the ideal situations where each blob corresponds to an object. However, we argue here that the “blob” is not an appropriate representation for efficient analysis for the goal of object-based tracking when the ideal situations are not met, which is common in realistic applications. Instead, we argue that the use of explicit object model as a representation for analysis is more advantageous.

The foreground blobs are inherently a result of bottom-up processing which follows the sequence of operations: change detection, filtering and connected component grouping. Its advantage is that the blob does not have any constraint on its shape. However this flexibility also causes problems. The structure (i.e., the number and the connectivity of blobs) may undergo rapid change as the objects move in the scene due to the fragmentation and multi-object split/merge as stated in Sec. 2.1. The structural change is sensitive to noise as well as the algorithmic parameters (e.g., the threshold in change detection, the amount of filtering and morphological operations, etc.). The structural change is also difficult to analyze due to its combinatorial nature because the correspondence of the blobs in two frames can be 1:1 (perfect match), 1:N (split), N:1 (merge), N:M (simultaneous split and merge). Furthermore, even if the split/merge analysis is successful, what each trajectory corresponds to is still unknown.

Reasoning with explicit object models shows advantages over blobs. First, the object description is a goal. As an entity existing in the physical world, it does not undergo the
structural changes such as split and merge. The prior knowledge (e.g., shape, motion) of the physical objects can assist in the segmentation and tracking. The object model-based approach is much less sensitive to noise or parameters of lower-level processing.

3.1.2 A simple solution to the model-based segmentation and tracking

We use an ellipsoid shape model for segmentation and tracking. First, the foreground mask is extracted by a change detection method. Human hypotheses are computed by boundary analysis and shape analysis using the knowledge provided by the human shape model and the camera model. Then, each hypothesis is tracked in 3D in the subsequent frames with a Kalman filter using the object’s appearance constrained by its shape. 2D positions are mapped onto the 3D ground plane and the trajectories are formed and filtered in 3D. Depth ordering can be inferred from the 3D information which facilitates the tracking of multiple overlapping humans and occlusion analysis. The approaches are effective in the complexity of temporary occlusion and persistent occlusion involving a small number of people in a group with shadow and runs in real time.

3.1.3 A Bayesian approach to the model-based segmentation and tracking

The simple solution has some drawbacks which become more severe as the number of people and the amount of occlusion increases. This has motivated more principled ways to solve the same problem.

The model-based segmentation problem is formulated in a Bayesian framework as to find the number of human objects and their parameters, which, together with a known background model, collectively make the best interpretation of the image observations.
The best interpretation is defined by Bayesian posterior probability which considers both the prior probabilities and the likelihood by the image observations. We propose two likelihood models, one based on foreground mask and the other based on color differences to the background. The explicit optimality definition makes the solution more robust and predictable.

However, due to the joint consideration of all unknown number of the objects, the computation of the optimal solution becomes non-trivial. This results in a large solution space with subspaces of varying dimensions, which the regular search or optimization methods are not likely to apply. We adapt a Markov chain Monte Carlo (MCMC) approach with jump and diffusion dynamics to explore the complex solution space. To improve the computational efficiency, we use heterogeneous domain knowledge (e.g., boundary analysis, contour analysis, shape analysis, efficient single object tracking techniques) as importance proposal probabilities to intelligently direct the Markov chain. The approach combines both the robustness of the Bayesian formulation and the efficiency of the data-driven methods.

The approach for segmentation is naturally extended to tracking by adding temporal constraints to enforce the connectivity and smoothness of the track. The color likelihood is augmented with the color profile of individual objects. Segmentation and tracking are obtained simultaneously. The measurement uncertainty is also estimated from the samples for optimal filtering.

We also want to mention that we extended the original mean shift color tracking [19] with a background exclusion principle to make use of a known background. It is used in both multi-human segmentation and tracking to provide approximations of gradient. It
is a relatively separate thread for color-based single object detection/tracking, therefore we include it in Appendix D.

3.1.4 A tracking as recognition approach to human body posture estimation

To estimate the human body postures, we propose a tracking as recognition approach where the estimation is accomplished by recognizing the motion in a locomotion model. Our locomotion model consists of 3 modes of walking, running and standing. The model is constructed by quantizing 3D motion capture data of human motion and is represented as a hierarchical finite state machine. The higher-level state machine represents the modes of locomotion while the lower level state machine represents phases of each mode (walking and running only) with each phase corresponding to a characteristic 3D body posture. It results in robust performance without the need for interactive initialization.

Sometimes the static shape information in one frame does not always yield correct hypotheses in the segmentation. We verify the human hypotheses by their dynamical aspect, i.e., by recognizing if they exhibit a proper walking pattern.

3.2 Modeling Image Formation

3.2.1 Camera modeling

A camera projects a 3D scene into an image. Most cameras are well approximated with a pin-hole model described by a 3 by 4 projection matrix, which is the one we use here. The camera model provides constraints on 3D objects in the image. It also enables making
estimation in 3D assuming people move on a known ground plane. More on the camera
d modeling is described in latter part of this chapter.

3.2.2 Background modeling

The stationary camera gives us the luxury to have a view-based background model without considering most of its geometry. We adapt a simple statistical background model as those used for change detection.

We use the background model in [117] where each pixel in the image is modeled by an independent Gaussian color distribution.

\[ P_b(I_j) \propto \exp\{- (I_j - \bar{T}_j)^T \Sigma_j^{-1} (I_j - \bar{T}_j)\} \tag{3.1} \]

where \( \bar{T}_j \) and \( \Sigma_j = \text{diag}\{\sigma_r^2, \sigma_g^2, \sigma_b^2\} \) are the mean and covariance matrix of the background model at pixel \( j \). The background model is first learnt from a period where no moving objects are visible and then updated for each incoming frame with the non-moving pixels. A single initial background frame is sufficient to start. The background model can be replaced with a more complex one (e.g., one with a multi-Gaussian model [105] or one which can start with moving objects in the scene [39]) if needed, however, this is not the focus of our work here.

Given the background model and an incoming frame, for each pixel \( j \), if the probability \( P_b(I_j) \) is lower than a threshold, this pixel is classified as foreground, otherwise as background. The binary result after the change detection is called a foreground mask. In
some parts of the work the foreground mask is used and the rest the background model is used directly.

3.2.3 Human modeling

At the beginning of this chapter, we have discussed the benefits of using a human shape model for analysis in complex situations. We choose to use three dimensional models, together with a camera model, to account for the variance of shape due to perspective effect. This can make our system applicable for a wide variety of viewpoints. For computational reasons, we choose to use approximating models. A single ellipsoid model is used in Chap.4, and multi-ellipsoid models are use in Chap.5,6. An ellipsoid can provide reasonable approximation of a human limb and its projection is an ellipse in a simple form.

Human appearance is also useful in tracking. We use textural template in Chap.4 and color histogram in Chap.6 for tracking.

We use a constant velocity model for human global motion in Chap.4,6. It is used with a Kalman filter to provide prediction and estimation. The prediction is useful when severe occlusion occurs. We model the human body posture in locomotion with a hierarchical finite state machine. By inference in the model, we can recover the motion mode and phase (3D body posture) of the human in each frame.

3.2.4 Shadow modeling

Human objects cast shadow on the environment by causing local change of illumination. We classify the shadow into two categories: shadow of a directed light source
(hard shadow) and shadow of a diffused light source (soft shadow). Their properties are summarized in the table below.

<table>
<thead>
<tr>
<th></th>
<th>directed light shadow</th>
<th>diffused light shadow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contrast</td>
<td>usually high</td>
<td>not very high</td>
</tr>
<tr>
<td>Boundary</td>
<td>well-defined</td>
<td>not well-defined</td>
</tr>
<tr>
<td>Pixel intensity change</td>
<td>high, homogenous</td>
<td>low, inhomogeneous but slowly changing</td>
</tr>
<tr>
<td>Visibility to human</td>
<td>yes</td>
<td>usually no unless observed carefully</td>
</tr>
<tr>
<td>Light source</td>
<td>simple, usually one</td>
<td>complex, a light field</td>
</tr>
<tr>
<td>Scenarios</td>
<td>sunny outdoor</td>
<td>indoor, in shade or overcast if outdoor</td>
</tr>
</tbody>
</table>

The shadow removal techniques as reviewed in Sec.2.1 work well for diffused light shadow, but they are not effective for sun shadows due to the saturation caused by the high contrast. In this work, we propose to handle the sun shadows by a geometric approach (Sec.4.2). The two techniques for shadow are complementary.

3.3 Camera Modeling and Geometric Constraints

The camera model and the geometric constraints provided are used in all the subsequent chapters.

3.3.1 Sensor deployment for visual surveillance

Many factors need to be considered to decide sensor deployment for visual surveillance applications. Here we give some intuitive qualitative analysis.

Coverage The effective coverage of a camera should be as large as possible. By effective coverage, we mean that the coverage which gives sufficient resolution for perception.
**Sensitivity to occlusion** The inter-occlusion of the objects should be minimal. Besides, the situation where a single object completely blocks the view of the camera should be especially avoided.

**Viewpoint** The information to be extracted should be visible from the selected viewpoint. (*e.g.*, a vertical viewpoint should not be used if the faces of the people need to be seen).

**Ease of deployment** The deployment should be physically convenient. Although a dedicated mount (*e.g.*, a pole) can always be constructed, it is ideal that the sensor can make use of existing structures (*e.g.*, walls, existing poles outdoor or walls or ceilings indoor).

Some of the factors contradict with each other therefore a compromise has to be made. The best configuration, also confirmed by most commercial (though not automatic) surveillance systems, is probably to mount the camera a few meters above the ground (up to the ceiling if indoor) and look down with a tilt angle of 20-40 degrees.

From our experiments, we also found that another factor should be considered. People walking shoulder by shoulder is a common pattern. In a camera whose view direction is perpendicular to the walking direction, significant and persistent occlusion may happen. This may even cause human observers difficulty in perceiving the occluded humans. Therefore, we suggested that for sites whose traffic is strongly directional, the camera should be set up to avoid such a situation.
A different camera setting seems to be popular in some related works. In [39], the camera is mounted on a tripod similar to human height and its optical axis is approximately parallel to the ground. Under this setting, the desirable effect is that the perspective effect results almost only in change in size, which makes the shape analysis easier (discussed in more detail in Sec.5.2). However, this setup creates the most occlusion: people in the front may completely occlude other people at the back.

Multiple cameras observing the site from different viewpoints can provide better visibility and complementary information. Although the works described in this thesis are all based on single camera input, the information from multiple cameras can be easily fit in the framework to increase estimation accuracy. A network of cameras can be used to provide coverage on a large site or inside an entire building such as in [106].

3.3.2 Camera calibration

The knowledge of the camera model provides constraints of the image projection of the 3D world (objects). It also enables the reasoning in 3D with the assumption that humans move on a known ground plane. We will show that the calibration can be done easily and the benefit is well worth the overhead.

Traditional camera calibration requires information on 3D feature points ($\geq 6$ points with $\geq 2$ out of the ground plane) and their corresponding image points. A linear calibration method in [30] is used to compute the camera projection matrix. We find this method works satisfactorily if the selected points are distributed evenly in the image. Non-linear method (Tsai [112]) can also be used if the required precision is higher.
Figure 3.1: The camera and the world: relationship of various geometric entities. (h_i are humans represented as poles, h'_i are their image projections)

When enough feature points are not available or a physical survey is not possible, methods based on the projective invariances (e.g., vanishing points and vanishing lines) can be used (Liebowitz et al. [64], Cipolla et al. [15], Lv et al. [67]). Vanishing points and vanishing lines can be recognized by structures in the environments or humans walking in more than one directions. (See Appendix A for more detail.) We have used this approach for most of the datasets in experimentation and it works satisfactorily.

3.3.3 The Transformations

We show here the transformations used in the following chapters related to the camera model. The relationship of various geometric entities is shown graphically in Fig 3.1.

3D point to point in image A point \( \mathbf{x} = [x, y, z, 1]^T \) in the world is related to a point \( \mathbf{u} = [u, v, 1]^T \) in the image by a 3 by 4 projection matrix \( \mathbf{P} \):
\[
\begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix}
\sim
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\]

(3.2)

3D ellipsoid to ellipse in image The projection of an ellipsoid is the world \((x^TQx = 0\), where \(Q\) is a 4 by 4 matrix) is an ellipse in the image \((u^T Cu = 0\), where \(C\) is a 3 by 3 matrix). Then their relation is given in [41] as

\[
C^{-1} = P Q^{-1} P^T.
\]

(3.3)

Point transformation between a 3D plane and the image The points in the image and the point on a 3D plane are related by a plane perspective transformation, also called homography, represented by a 3 by 3 matrix \(H\) as

\[
\begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix}
\sim
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

(3.4)

The relation is inversible,

\[
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\sim
\begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix}
\]

(3.5)

with the exception that the image of the 3D plane is a line (i.e., the 3D plane passing the optical center, which does not happen in our case.). We call the 3D plane that objects move on the ground plane. For simplicity, we assume that the ground plane is \(z = 0\), and the matrix \(H\) has the simple form that \(H = [p_1, p_2, p_4]\), where \(p_i\) is the \(i\)-th column vector of the camera projection matrix \(P\). The transformation between 2D and 3D can be extended to any known 3D scene structures. The transformation
can be computed by the intersecting the 3D ray corresponding to a pixel with the structure.

**Velocity transformation between ground plane and image** The Jacobian of the homography transformation is

$$J = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}. \quad (3.6)$$

The velocity of \((\Delta u, \Delta v)\) at \((u_0, v_0)\) is related to the velocity of \((\Delta x, \Delta y)\) at \((x_0, y_0)\) by

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = J \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}. \quad (3.7)$$

This relationship is inversible,

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = J^{-1} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}. \quad (3.8)$$

Note that the result computed by the Jacobian only gives a first order approximation of the transformation of velocity. One can find the precise transformation by applying the homography on \([x_0 + \Delta x, y_0 + \Delta y]^T\).

**Vertical direction** The direction passing through an image point \(u_0\) corresponding to the vertical direction in the world is simply the direction from \(u_0\) to the vertical vanishing point of the camera \(V_z\), where \(V_z \sim p_3\). (Similarly, the vanishing points of \(x\) and \(y\) direction correspond to \(p_1\) and \(p_2\) respectively.)

**2D height to 3D height** Assume \(u_0\) is the image projection of a point \(x_0\) on the ground plane. Point \(x = [x, y, h]^T\) is vertically above \(x_0\). The measured 2D height is \(l\) (along
its vertical direction) and we want to compute its 3D height $h$. The projection of $x$ yields $u(h)$, $h$ can then be computed by solving the equation $\|u_0 - u(h)\|^2 = l^2$. 
Chapter 4

Simple Model-based Segmentation and Tracking

This chapter contains our work for the goal of segmentation and tracking multiple humans using human shape model. Some example images from our dataset and the enlarged foreground of interesting portions are shown in Fig.4.1. Motivated by the discussion in Sec.3.1.1, human shape model (a simple ellipsoid) is used. The segmentation is based on analyzing the shape and the boundaries of the foreground from change detection. The segmented human objects are then tracked using their appearance models. Multiple overlapping objects are tracked considering their inferred depth order. Experiments and evaluation show significant advantages over the blob-based methods in the situations where people walk in small groups, have frequent passing-bys and in the presence of shadows.

4.1 Introduction

The system diagram is shown in Fig.4.2. First, change detection is applied on the video frames with a learnt background model to extract the foreground mask (see Sec.3.2.2). We observe that human heads have a low chance to be occluded and are easy to extract.
Figure 4.1: Example images from our dataset and the enlarged patches of foreground from them. One blob may contain multiple objects due to their proximity or camera projection (a)(c). It may also contain non-object pixels such as shadow (sun shadow in (a) and soft shadow in (c)) or reflection (b). The foreground corresponding to an object may get fragmented (c).

Therefore we use them to segment human objects by boundary analysis and shape analysis using the knowledge provided by the human shape model and the camera model. Then, each segmented object is tracked in 3D in the subsequent frames using its appearance and its shape. A Kalman filter is used for filtering and prediction. 2D positions are mapped onto the 3D ground plane and the trajectories are formed and filtered in 3D. Depth ordering can be inferred from the 3D information which facilitates the tracking of multiple overlapping humans and occlusion analysis.

4.1.1 Human model

We employ a very simple human shape model in the work presented in this chapter. The model is an ellipsoid whose axes are aligned with axes of the world coordinate system. Its longest axis is in the vertical direction corresponding to the height of a human. The other two axes are of the same length with a fixed ratio $r$ ($r = 0.3$) to the height. Suppose
we know the human’s feet is at \([x,y,0]^T\) and his/her height is \(h\), the 4 by 4 matrix representing the ellipsoid is

\[
Q = 4 \begin{bmatrix}
\frac{1}{r^2 h^2} & 0 & 0 & -\frac{x}{r^2 h^2} \\
0 & \frac{1}{r^2 h^2} & 0 & -\frac{y}{r^2 h^2} \\
0 & 0 & \frac{1}{h^2} & -\frac{z}{h^2} \\
-\frac{x}{r^2 h^2} & -\frac{y}{r^2 h^2} & -\frac{z}{h^2} & \frac{x^2 + y^2}{r^2 h^2}
\end{bmatrix}
\] (4.1)

As described in Sec.3.3, the projection of ellipsoid \(Q\) in the image is an ellipse \(C\) by Equ.3.3, which defines a mask \(M\). The model is used for both segmentation and tracking.

4.1.2 Notation

In the following parts of this chapter, the following notation will be used:
\( I \) - image frame

\( F \) - foreground mask from change detection

\( Fr \) - foreground residue map

\( O_i \) - occlusion mask of object \( i \)

\( M \) - the elliptic shape mask of an object

\( Fp \) - the template of foreground probability of an object

\( T \) - the textural template of an object

Their meanings will be explained upon first appearance. We use a subscript to denote the frame number, for example \( I_n \) means the image of frame \( n \). When a specific object is involved, double subscript is used. For example, \( M_{i,n} \) means the elliptic shape mask of object \( i \) at frame \( n \).

### 4.2 Segmenting Multiple Humans

We attempt to interpret the foreground mask with such ellipsoid shape models. It involves an iterative procedure to find a head candidate on the foreground boundary, form a hypothesis, and remove the object from the foreground. This process is described below and shown step by step graphically in Fig 4.4.

#### 4.2.1 Head candidates by boundary/shape analysis

For a camera placed several meters above the ground, the head of a human has a low chance to be occluded; we found that recognizing the head top on the foreground boundary is a simple and effective way to segment multiple, possibly overlapping humans.
A point can be a head top candidate if it is a local peak (i.e., highest point in the vertical direction) along the boundary of $F$ within a range (Fig. 4.4(a)). The range is defined by the average size of a human head assuming an average height. Flat peak (i.e., the peak with more than one equally highest points) is also allowed and the head top is defined as the center of the flat peak.

Once such a head top candidate is found, an ellipsoid human model of an average height is formed (Fig. 4.3(a)). The portion of foreground pixels in the human model $r_f$ is computed. If $r_f$ is lower than a threshold $T_f$ (e.g., 0.6), this candidate is discarded. The remaining ones (Fig. 4.3(b)), denoted as a set $Hc$, will be used to generate human hypotheses as described below.

Haritaoglu et al. [39] also used foreground boundary to locate body parts (e.g., head, hands, feet) of a single human by curvature analysis. In our case, we are only interested in heads and our method is simpler and more robust. [39] also use peaks in the vertical projection of the foreground blob to segment multiple humans. It works well in the case of a ground level camera where people are all within a band in the middle of the image; it may not work well for a tilted camera in which humans also distributed vertically in the image.

4.2.2 Geometrical shadow analysis

Most of the current shadow removal approaches focus on the color aspect (summarized in Sec. 2.1) and the geometrical aspect is overlooked. The color-based approaches are not expected to work well due to the high contrast of sun cast shadow in an outdoor environment (e.g., dark pants and shadow region may be of the same intensity), though
they may be effective for lower contrast soft shadows. Here we describe the method that we use to account for sun cast shadow.

Assuming that the sun is the only light source and sun light direction is known, the shadow cast by an ellipsoid shape model on the ground after camera projection is easily determined (also an ellipse, Fig.4.3.(c)). The sun light direction can be computed by existing package (e.g., [79]) given the geographic location, time, date and year, which can be assumed known in a surveillance system.

Any foreground pixel which lies in the shadow ellipse and whose intensity is lower than that of the corresponding pixel in the background by a threshold $T_s$ is classified as a shadow pixel.

4.2.3 The algorithm

We denote the foreground mask $F$ with the existing human masks removed as foreground residue map $Fr$. At the beginning of the segmentation, $Fr$ is initialized by $F$ \footnote{If the segmentation is combined with tracking, we remove the masks of the objects already being tracking from $Fr$.}. The head top candidate set $Hc$ is computed on $Fr$. We compute a depth value for each of the head top candidate assuming that they are of an average height. The one with the minimum depth value (closest to the camera) is chosen to form a human hypothesis.

After a head top candidate is chosen, we find its potential height by finding the first point that turns from a foreground pixel to a background pixel along the 3D vertical direction (towards the vertical vanishing point) in the range determined by the minimum height and the maximum height. We do this for all points in the head area and take
the median value, which enables finding the height of different human postures. See Fig.4.3.(b) for a graphical illustration. Having head top position and the height, an ellipsoid human model $Q$ is generated.

Once a hypothesis is generated, its projection $C$ is computed by Eq. 3.3, resulting in a shape mask $M$. The mask and the shadow pixels associated with the object are removed to update $Fr$. Fig.4.4.(d) shows the first 4 segmented humans and the foreground after their masks and shadow pixels are removed. As can be seen, a large portion of the shadow pixels are removed correctly. Morphological “open” operation is performed on $Fr$ to remove the isolated small residues (Fig.4.4.(e)). This iterative procedure goes on until no new head top candidate is found. The algorithm is summarized in Fig.4.5.

This approach works well for a small number of overlapping human objects that are not severely occluded; a severely occluded object will be detected when it becomes more visible. This method is not sensitive to blob segmentation if a large portion of the object still appears in the foreground. The detected head top candidates have low false alarm rate. All the false alarms usually correspond to large foreground region not (directly) caused by a human. For example, when people move with their reflections, the reflections are also hypothesized as humans (as in Fig.4.4.(g)). We can verify the hypotheses with dynamical features to remove such false alarms during tracking, as will be described in Sec.7.5.2.
Figure 4.3: Some techniques used in human segmentation. (a) Screening a head candidate by forming a human model of average height; (b) Finding the height of a hypothesis; (c) The geometry of sun cast shadow.

4.3 Tracking Multiple Humans

Once segmented, the objects are tracked in the subsequent frames. The tracking process is a loop consisting of: predict the location from the previous frame; find the best match from the predicted location in the image; update the object representations. We will address below the object representation for tracking, how to compute the best match, state estimation with Kalman filter, object updating, and some issues on handling occlusion. The algorithm is summarized in Fig.4.7. In each frame, we first track the objects that were segmented before, then we remove the masks of these objects from the foreground and apply segmentation on the foreground residue.

4.3.1 Object representation for tracking

The elliptic shape mask $M$ projected from the ellipsoid model represents the gross human shape. The shape/scale of the mask changes automatically according to the human’s position. A texture template $T$ is used to represent the appearance of each individual
Figure 4.4: The process of human segmentation on an example. (a) peaks of the foreground boundaries; (b) screened peaks as head candidates; (c) after 3 hypotheses formed; (d) foreground residue after removing hypothesized objects and their shadows (before morphological operation); (e) finding head candidates iteratively (after morphological operation); (f) all hypotheses formed; (g) false alarm when inclusion of non-object foreground of human shape.

human. Not every pixel inside the elliptic mask corresponds to the foreground due to the ellipse approximation; we also keep a foreground probability template \((F_p)\) for each human object which stores the probability of each pixel in the elliptic mask as foreground. It enables handling some variations in body shape and posture. Fig. 4.6 shows examples of this representation.

Due to camera perspective effect, the elliptic projections of the same ellipsoid have different shape (i.e., orientations and lengths of the axes) when the human is at different locations. Therefore, a mapping is needed to align different ellipses for matching and updating. Suppose we have two ellipses \(e_1 = \{u_1, \alpha_1, \beta_1, \theta_1\}\) and \(e_2 = \{u_2, \alpha_2, \beta_2, \theta_2\}\) in their parametric forms where \(u, \alpha, \beta\) and \(\theta\) are the center, long axis, short axis
Fr ← F;
Hc ← ComputeHeadCandidates(Fr);
while Hc not empty
    select c ∈ Hc with the minimum depth;
    h ← ComputeHeight(c);
    compute Q, C, M;
    subtract M and shadow pixels from Fr;
    Fr ← Open(Fr);
    Hc ← ComputeHeadCandidates(Fr);
end.

Figure 4.5: The algorithm of multi-human segmentation.

Figure 4.6: Examples of object representation for tracking and its evolution. For each object: shape mask (1st column), texture template (2nd column), and foreground probability template (3rd column). Rows correspond to frame 1, 25, 100, 200.

and the rotation respectively. A mapping \( u' = W(u) \) transforms a point \( u \) in \( e_1 \) to its corresponding point \( u' \) in \( e_2 \) by aligning \( e_1 \) and \( e_2 \) by their centers and corresponding axes through translation, rotation and scaling by

\[
\begin{align*}
    u' &= W(u) \\
    &= \begin{bmatrix}
    \cos \theta_2 & -\sin \theta_2 \\
    \sin \theta_2 & \cos \theta_2
    \end{bmatrix}
    \begin{bmatrix}
    \frac{\alpha}{\beta} & 0 & 0 \\
    0 & \frac{\alpha}{\beta} & \beta
    \end{bmatrix}
    \begin{bmatrix}
    \cos \theta_1 & \sin \theta_1 \\
    -\sin \theta_1 & \cos \theta_1
    \end{bmatrix}
    [u - u_1] + u_2.
\end{align*}
\]  

(4.2)
4.3.2 Computing the best match

Suppose we have an estimate of the position of a human object in frame \( n - 1 \), \( \hat{\mathbf{u}}_{n-1} = (\hat{u}_{n-1}, \hat{v}_{n-1}) \) and have a prediction of its position and the covariance matrix of it in frame \( n \), \( \mathbf{u}_n = (\mathbf{u}_n, \mathbf{v}_n) \) and \( \mathbf{\Sigma}_n \) (from a filter, such as one introduced in Sec.4.3.3). Starting from \( \mathbf{u}_n \), a search is conducted for all the positions in a neighborhood (\( \mathbf{u} \in \Omega \)) to locate each human object using the representations in Sec.4.3.1.

Multiple human objects are matched one by one, starting from the one closest to the camera (their depth order to the camera are known according to their 3D positions and the camera model). An occupancy map \( O \), \( O(\mathbf{u}) = 1 \) if pixel location \( \mathbf{u} \) is occupied by another object (in front of the current object), is used to facilitate the multi-object matching. \( O \) is initialized with zeros and filled with an object mask once its position is fixed.

For each object, we determine the search range \( \Omega \) using the following constraints. Limiting \( \Omega \) as much as possible both reduces the computation of matching and minimizes the potential of false matching.

- The knowledge of maximum human motion velocity,

- The covariance of the predicted position \( \mathbf{\Sigma}_n \) (in practice, 3 times of the standard deviation),

- The physical occupancy constraint, \( i.e., \) a human object cannot occupy the space already taken by other objects.
The above constraints on 3D position are projected into the image domain. The shape of $\Omega$ is usually thin. Its size is larger when the object is closer to the camera and is smaller when the object is farther away.

The best estimate of the position of a human object in the current frame is given by

$$u^* = \arg \min_{u \in \Omega} \left( \sum_{i \in M^{n-1}} E_i(u) \right),$$  \hspace{1cm} (4.3)

where $E_i(u)$ is the penalty due to pixel location $i$ in the shape mask if the position of the human feet is at $u$, which is computed by

$$E_i(u) = \begin{cases} 
Fp_{n-1}(i)||T_{n-1}(i) - I(W_u(i) + u)||, & \text{if } O(W_u(i) + u) = 0 \\
E_o, & \text{otherwise},
\end{cases}$$

where $W_u(i)$ is the transformation from the ellipse of frame $n-1$ to the ellipse determined by feet position $u$ and $|| \cdot ||$ is the Euclidian distance of the color (RGB) of two pixels. $E_o$ is a predefined penalty for a pixel occluded by other humans and set larger than the average error of a pair of correctly matched pixels. This penalty avoids an object to “hide” behind another object unless necessary. This matching criterion is similar to template matching after ellipse warping and with consideration of occlusion.

### 4.3.3 Estimation with Kalman filter

We estimate the state of each human object using a Kalman filter [52] with a constant velocity model (Eqn.4.4). At each frame, the position is predicted by the Kalman filter, the best position is obtained as above and is used to update the filter parameters.
\[
\begin{bmatrix}
  x_n \\
  y_n \\
  \dot{x}_n \\
  \dot{y}_n
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & \Delta t & 0 \\
  0 & 1 & 0 & \Delta t \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x_{n-1} \\
  y_{n-1} \\
  \dot{x}_{n-1} \\
  \dot{y}_{n-1}
\end{bmatrix} + w_n;
\]

\[
\begin{bmatrix}
  \dot{x}_n \\
  \dot{y}_n
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
  x_n \\
  y_n \\
  \dot{x}_n \\
  \dot{y}_n
\end{bmatrix} + v_n,
\]

where \( x_n = [x_n, y_n, \dot{x}_n, \dot{y}_n]^T \) are the 3D position and velocity on the ground plane, \( \hat{x}_n = [\hat{x}_n, \hat{y}_n]^T \) are the measured 3D positions. \( w_n \sim \mathcal{N}(0, \text{diag}(\sigma_z^2, \sigma_y^2, \sigma_z^2, \sigma_y^2)) \) is the process noise, including the deviation from the constant velocity assumption. \( v_n \sim \mathcal{N}(0, \hat{\Sigma}_n) \) is the measurement noise and \( \Delta t \) is the time interval since last update. \( \Delta t \) equals to temporal sampling interval unless the object is occluded. Although the observation \( \hat{x} \) only consists of position, the estimation \( \hat{x} \) also includes velocity.

\( \hat{x}_n \) is calculated from the image position of the human object \( u_n^* = [u_n, v_n]^T \) by Equ.3.5. Assuming a fixed 2D measurement (matching) noise \( \mathcal{N}(0, \text{diag}(\sigma_u^2, \sigma_v^2)) \), the measurement noise in 3D is still different for the object at different positions due to the perspective effect of the camera projection. \( \hat{\Sigma}_n \) can be computed as in Equ.4.5 by assuming a first order approximation of the homography between the ground plane and the image plane. This gives us good filtering results (especially for the motion direction) in spite of the perspective effect.

\[
\hat{\Sigma}_n = J(u_n)^{-1} \begin{bmatrix}
  \sigma_u^2 & 0 \\
  0 & \sigma_v^2
\end{bmatrix} (J(u_n)^{-1})^T; \quad \text{where } J(u)_n \text{ is defined in Equ.3.6.} \quad (4.5)
\]
4.3.4 Object updating

The texture templates and foreground probability templates are initialized at the first frame where the human objects are hypothesized. They are updated using the new observations of each frame to take into account the change of appearance and shape over time. Simple IIR (infinite impulse response, i.e., a weighted average of warped templates in the last frame and the new observations) filters are used

\[
T_n(i) = \begin{cases} 
T_{n-1}(W(i)), & \text{if } O(i + u) = 1 \text{ or } F_n(i + u) = 0; \\
(1 - \alpha)T_{n-1}(W(i)) + \alpha I_n(i + u), & \text{otherwise.} 
\end{cases} \tag{4.6}
\]

Similarly,

\[
Fp_n(i) = \begin{cases} 
Fp_{n-1}(W(i)), & \text{if } O(i + u) = 1; \\
(1 - \alpha)Fp_{n-1}(W(i)) + \alpha F_n(i + u), & \text{otherwise.} 
\end{cases} \tag{4.7}
\]

where \(u\) is the feet position which is also the origin of the templates, and \(\alpha\) is an updating factor. Note that the texture map is only updated with foreground pixels. Both maps are updated with the non-occluded parts. Fig.4.6 shows the evolution of the templates in an example where both the objects’ size and orientation changed.

4.3.5 A summary

The algorithm for multi-human tracking is summarized in Fig.4.7. Occlusion of multiple objects are addressed in several places in the algorithm. We compute the ratio of the visible pixels \(r = N_v/N_e\), where \(N_v\) is the number of visible foreground pixels in the elliptic mask and \(N_e\) is the number of pixels in the elliptic mask of each object. By visible foreground pixels, we mean the pixels that are not in the shape masks of other human
sort objects according to depth order;
\[ O \leftarrow 0; \]
\textbf{for} each object \( i \)
\[
\begin{align*}
\{\mathbf{x}_{i,n}, \mathbf{\Sigma}_{i,n}\} & \leftarrow \text{KalmanPrediction}(); \\
\Omega & \leftarrow \text{ComputeSearchRange}(\mathbf{x}_{i,n}, \mathbf{\Sigma}_{i,n}); \\
\mathbf{u}_{i,n} & \leftarrow \text{HomographyTransI}(\mathbf{x}_{i,n}); \quad \text{//Equ.3.4} \\
\mathbf{u}^t_{i,n} & \leftarrow \text{SearchBestPosition}(\mathbf{u}_{i,n}, \Omega, M_{i,n-1}, F_{i,n-1}, T_{i,n-1}, I_n, O); \\
\{M_{i,n}, F_{i,n}, T_{i,n}\} & \leftarrow \text{UpdateRepresentation}(\mathbf{u}^t_{i,n}, F_{i,n-1}, T_{i,n-1}, I_n, F_g, O); \\
O & \leftarrow O + M_{i,n}; \\
\{\hat{\mathbf{x}}_{i,n}, \hat{\mathbf{\Sigma}}_{i,n}\} & \leftarrow \text{HomographyTransII}(\mathbf{u}^t_{i,n}); \quad \text{//Equ.3.5,4.5} \\
\{\hat{\mathbf{u}}_{i,n}, \hat{\mathbf{\Sigma}}_{i,n}\} & \leftarrow \text{KalmanUpdate}(\hat{\mathbf{x}}_{i,n}, \hat{\mathbf{\Sigma}}_{i,n});
\end{align*}
\textbf{end.}

Figure 4.7: The algorithm of multi-object tracking.

objects in front of the object. We use two thresholds \( 1 > T_{o1} > T_{o2} > 0 \) to decide the
occlusion status of an object. If \( T_{o1} > r > T_{o2} \), the object is said to be partially occluded.
If \( r < T_{o2} \), the object is said to be completely occluded. In matching, only the visible
parts are used, and the search range is adaptive to occlusion. In Kalman filtering, the
measurement noise is affected by the amount of occlusion. In case of a full occlusion,
the position is predicted by the Kalman filter. The templates are updated only with the
visible part. If the object is completely occluded for a certain number of frames, it is
removed.

4.4 Implementation and Results

4.4.1 Entrances and exits

To handle the entry and exit of human objects, the user provides the information of
entrances/exits besides the image boundaries (marked on the images in the first panes
of Fig.4.8 and Fig.4.9). The entrances/exits may be real entrances/exits (e.g. a gate)
or occluding structures (e.g. a roof). We only generate a human hypothesis if it does not touch the image boundary or the marked entrances. This effectively causes a human object to be detected only after it is fully in the scene and avoids hypothesizing it with partial evidence. A human object is removed if it touches the image boundary or a marked exit. Usually entrances and exits appear in pairs corresponding to the same physical locations. We set the entrances to be a few pixels inside the scene than the exits to avoid the case that an object is hypothesized again after it is removed. Stauffer [107] learns the entrances and exits of a scene automatically, however, user interaction is also satisfactory for many applications since it is only needed once (like the calibration of the camera).

4.4.2 Results on two examples

We have tested the above approach on various datasets from different sources and obtained good results. The system can segment and track multiple humans in small groups and with temporary severe occlusion very well. It is insensitive to image noise by blob fragmentation or camera shaking. To limit space, we only show the tracking result of two representative sequences and report the evaluation next. The system runs in real-time (30 fps or more) on videos of complexity shown in the examples with frame size 360 by 240 on a Pentium 4 2.7G Hz PC.

The first sequence (seq1, in Fig.4.8) includes three humans walking in a close group, humans passing-by each other and single/multiple humans passing an obstacle. All humans were tracked successfully and their positions were estimated accurately. We also
implemented a blob tracker which can handle split and merge (see Appendix B) for comparison. As the selected frames (as well as the movie) show, the bounding boxes of foreground blobs split and merge frequently though the humans move smoothly. The search region size changes when humans are in different positions and grows when going behind the occluding object. The last two panes of Fig.4.8 show the humans’ position and velocity mapped onto the ground plane which is the true physical metric and reveals the real relationships (e.g. distance) of different humans.

The second sequence (seq2, in Fig.4.9) is a segment containing dense traffic from a longer sequence. There are 13 people going through the scene in different directions resulting in many instances of their passing-by each other. In particular, a roller-skater (ID33) and a biker (ID34) pass through two walkers (ID35, 31) with similar clothing color at high speeds. Both their identities and their positions are tracked accurately.

4.4.3 Performance evaluation

Besides observing the results visually, we evaluate the performance of the system quantitatively on a dataset. The sequences in the dataset are selected to be heterogeneous (also summarized in Tab.4.1.(a)): they are obtained from different sources (including a sequence in PETS01 dataset [82]), captured on 9 distinctive sites, with the camera tilt angle ranging from 5° to 40° and image height of a typical human ranging from about 25 pixels to about 80 pixels. Snap shots of some of the sequences are shown in Fig.4.10. The total length of the data is 61890 frames (35 minutes) and the total number of humans that appear is 520. The number of human-frames (i.e., the summation of the frames of each human) is 243804 (135 minutes).
Figure 4.8: Selected frames of human global motion tracking result of video sequence seq1 and comparison with a blob tracker. For blob tracker, the color of bounding boxes and trails reflects the temporal correspondence. For human tracker, human objects are shown in blue ellipses, shadow direction is shown in green lines, search mask size is shown at the feet of each human. Last two panes show the human trajectories projected onto the ground plane. Circle means human, arrow means velocity, red and black trails mean measured and filtered trajectories respectively. The entrances (in yellow lines) and the exits (in blue lines) besides the image boundaries are marked in the first pane, same for Fig. 4.9.
Figure 4.9: Selected frames of human global motion tracking result of a segment containing heavy traffic seq2 from a longer sequence. Object IDs are shown on the top to help the readers differentiate people with similar clothing.

Figure 4.10: Snapshots of some of the sequences that we used for performance evaluation.
The performance of human segmentation and tracking is affected by the complexity of the data. Generally, the more occlusion, the more challenges the data pose to the algorithms. It is known that during a few people walking in a group or passing-by the errors such as track drift or switch are more likely to happen. We describe the complexity of the dataset by the number of the events challenging for the system: “human passing-by each other”, “human passing an obstacle” and “walking in a group”. A “passing-by” means two or more humans crossing each other with the people further away from the camera partially or completely occluded. A “passing an obstacle” means a human moves behind a structure (e.g., a tree or a pole) in the scene. A “walking in a group” means two or more humans walking together closely (e.g., side by side) and persistently. The counts of such events are summarized in Tab.4.1.(b).

Quantitative evaluation of multi-object tracking is more difficult than object detection, object recognition or single object tracking due to the complex behaviors created by the system. These behaviors need various statistics to capture in order to be meaningful to different applications. Some discussions have been given in [83]. We do not intend to address these general issues.

We use the occurrence of typical types of errors of the system (in relation to the size of the dataset and the occurrence of challenging events) as metrics for evaluation. The errors are based on trajectories, which are more meaningful than frame-based errors for a tracking system. The errors are first classified into detection-based errors and tracking-based errors. Detection-based errors further include missed-detection (partial or complete), false alarm and redundant detection. A missed-detection means a human’s trajectory is partially or completely not detected during its presence in the scene. The short delay
right after a human entry is not counted as partial missed-detection. If a detection does not overlap with any human objects in the scene, we call it a false alarm; if multiple detections overlap with a human object, we call the most accurately located on correct detection and the rest redundant detections. We separate redundant detections from false alarms because these two may have different influence in some applications. The tracking-based errors include track drift onto other human, track switch between two humans and other track lost. The failures categorized here are by no means a complete classification, however, they cover most of the typical errors that have been observed.

The error counts are summarized in Tab.4.1.(c), where we also derive the percentages w.r.t. the number of people to make the numbers more intuitive. These numbers should also be considered against the number of challenging events.

The main reason for missed-detection is change detection failure due to object’s lack of contrast with the background. It happens more frequently in profile views since the projected human size is smaller than the frontal/back views. Missed-detection also happens when a human is severely occluded by other humans. The number of false alarms is small since most of the image noise does not have shape and size comparable to a human. The main reason for redundant detection is the inclusion of non-human regions (e.g., soft shadow or bicycle) in the foreground. Incorporating shadow removal algorithms [88] should reduce redundant detections.

The typical scenario for the track of a human to drift on another human is when they have similar clothing and move closely together for extended period of time (in contrast to “passing-by”). Since the object closer to the camera has priority for its position, it may erroneously shift onto the human at the back. If the depth difference is small, the
human at the back may lock on the frontal human, resulting in a track switch. This error is mainly due to the greedy nature of matching the multiple objects. Joint likelihood of all the overlapping objects and the background should be considered in future research. However, the increased computation due to the joint likelihood may limit the use of the system. Other track lost includes the situations such as the object (biker or roller-skater) exhibits fast change of velocity.

4.4.4 Remarks

This work has shown the effectiveness of using human shape model for segmentation and tracking. It minimizes the difficulty of analysis of the potentially frequent split and merge behavior of blob-based representation. However, the approach is based on some assumptions that may not be valid in general setting, resulting in some possible deficiencies.

In segmentation, the initialized human hypotheses are based on the head top candidates and heuristics of finding height, which may not give the optimal localization. Initialization is important for the tracking of the object. Inaccurate segmentation can result in poor performance of tracking. In case of severe occlusion, we find that the segmentation is not always satisfactory.

In tracking, the objects are matched according to their depth order from the one in the front to the one at the back. An object in front of others has absolute priority in the matching process. The error in matching a front object may affect all the objects behind it. A typical error is that when two overlapping objects (A and B, A in front of B) have similar appearance (e.g., clothing color) and their overlap lasts long enough, object A
(a) The summary of the heterogeneous dataset used for evaluation.

<table>
<thead>
<tr>
<th>Event</th>
<th>Count</th>
<th>Event</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>passing-bys involving 2 humans</td>
<td>171</td>
<td>walking in group of 2</td>
<td>48</td>
</tr>
<tr>
<td>passing-bys involving 3 humans</td>
<td>31</td>
<td>walking in group of 3</td>
<td>13</td>
</tr>
<tr>
<td>passing-bys involving 4 humans</td>
<td>16</td>
<td>walking in group of 4</td>
<td>1</td>
</tr>
<tr>
<td>passing-bys involving 5 humans</td>
<td>5</td>
<td>passing obstacles</td>
<td>42</td>
</tr>
</tbody>
</table>

(b) The occurrence of different types of events which pose challenges to the system.

<table>
<thead>
<tr>
<th>detection-based errors</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of error</td>
<td>Count</td>
<td>Percentage</td>
<td></td>
</tr>
<tr>
<td>Missed-detection (complete,partial)</td>
<td>13,16</td>
<td>2.5%,3%</td>
<td></td>
</tr>
<tr>
<td>False alarm</td>
<td>4</td>
<td>0.8%</td>
<td></td>
</tr>
<tr>
<td>Redundant detection</td>
<td>10</td>
<td>1.9%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>tracking-based errors</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of error</td>
<td>Count</td>
<td>Percentage</td>
<td></td>
</tr>
<tr>
<td>Track drift onto other human</td>
<td>25</td>
<td>6.4% combined</td>
<td></td>
</tr>
<tr>
<td>Track switch</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Track lost</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(c) The occurrence of different types of errors.

Table 4.1: The performance evaluation of the segmentation and tracking system. (a) basic statistics of the dataset; (b) the occurrence of challenging video events, as an indication of data complexity; (c) the occurrence and percentage (w.r.t the number of people) of typical errors.
may drift on the image of object B due to the ambiguity and noise. The portion of the image of object B is then unavailable in matching the correct location of object B. This calls for a global view of all the overlapping objects where a joint likelihood is needed. This is addressed next in Chap. 5 and 6.
Chapter 5

Model-based Segmentation: a Bayesian Approach

In Chapter 4, we have described a system that segments and tracks multiple human objects when the number of people in the scene is limited and people have limited occlusion between each other. These approaches have some drawbacks as discussed in Section 4.4.4. The drawbacks may become more severe as the number of people and the amount of their occlusion increases. This has motivated the work we describe in this chapter and the next chapter, in which the same problem is reformulated and solved in a more principled way. This results in a system which is more general and capable to perform successfully in crowded situation (e.g., Fig.5.1).

Figure 5.1: A sample input frame (a), its foreground from change detection (b) and the segmentation result (c).
5.1 Motivation and Introduction

5.1.1 Motivation

When there are sparse objects in the scene, connected component analysis after some noise filtering and morphological operations (i.e., close) on the foreground obtained from change detection usually generates satisfactory segmentation results. As the density of objects increases to the level as depicted in Chapter 4, the segmentation method based on simple boundary and shape analysis suffices. However, as the number of people and the amount of occlusion continue to increase to scenarios similar to those in Fig.5.1, these methods are likely to break down. The connected components of the foreground in Fig.5.1.(b) are not sufficient to provide information at the object level. The segmentation technique in Sec.4.2 can only reliably detect the people whose heads are on the foreground boundaries. Iteratively removing hypotheses may lead to erroneous results in the big blob with heavy (e.g. over 50%) inter-occlusion since the iterative removal assumes slight occlusion.

There are a number of other techniques in the computer vision research community which seem to be related to our goal for object segmentation.

- Image (color/texture) segmentation (e.g., [18], [113]) groups image pixels into regions of similar color/texture; it is not likely to segment individual humans because human clothing may not have uniform color/texture and adjacent people may wear similar clothes (e.g., white T-shirts in summer).

- Motion segmentation (e.g., [56]) groups image pixels into regions of consistent motion; it may not give satisfactory result due to non-rigid human motion and the similarity of the motion of individuals in a group.
• Face detection technology has achieved rapid advances recently by using statistical learning approaches (e.g., Ada-boost [116], KL-boost [66]); however it may not be effective on the small image size of the faces as in the typical surveillance video resolution and arbitrary head pose w.r.t the camera (360° orientation and varying tilt angle).

• Human detection only received limited success by statistical learning approach ([75], restricted to frontal/back viewpoint) or matching shape with Chamfer distance ([34]), both without occlusion, and with a ground level camera to avoid perspective effects 1.

5.1.2 Introduction

As discussed in Chap. 4, a human shape model should be used for segmentation. Instead of a single ellipsoid, we use a number of multi-ellipsoid models to capture the gross shape of human standing or walking. The problem of segmenting multiple overlapping humans (i.e., model-based segmentation) is formulated in a Bayesian framework to find the number of human objects and their parameters which, collectively, make the best interpretation of the image observations, together with a known background model. The best interpretation is defined by the Bayesian posterior probability which considers both the prior knowledge and the likelihood of the image observations according to the image

1It is also due to the requirement of the application (smart vehicle) that a ground level camera is used in [34].
formation process. Here we design two likelihood models: a foreground/background-based likelihood model and a color-based likelihood model. The explicit definition of optimality makes the solution more robust and predictable.

However, due to the joint consideration of all the objects, the optimal solution has to be computed in a joint parameter space of all the objects. Furthermore, since the number of objects is unknown, all numbers should be considered. This results in a large solution space with subspaces of varying dimensions, for which the regular search or optimization methods are not likely to apply. We adapt a Markov chain Monte Carlo (MCMC) approach with jump and diffusion dynamics to explore the complex solution space. To improve the computational efficiency, we use domain knowledge as importance proposal probabilities to intelligently direct the moves of the Markov chain. The approach combines both the robustness of the Bayesian formulation and the efficiency of the data-driven methods. This approach can also be regarded as a stochastic version of the widely-used hypothesize and test approach (e.g. [78]) [113].

In the rest of this chapter, we first introduced the 3D shape model followed by the definition of Bayesian posterior probability. The MCMC approach to find the maximum a posteriori (MAP) estimate is described next and finally we present the experiment results.

5.2 3D Human Shape Models

5.2.1 2D models v.s. 3D models

It becomes obvious from the above analysis that some knowledge of valid human shape is needed for segmentation. In choosing the appropriate representation of human shape,
we face the choices of either 3D model-based representation or view-based representation. We will only state our choice together with its justification in our particular application.

Shapes of objects (e.g., humans) usually have some variations. A probabilistic shape model can be constructed by learning from a large number of training samples (e.g., by performing principle component analysis (PCA)). A 2D model is easier to construct because of the simplicity of representation (usually by its outer boundary with chain code or spline by a number of control points on it (e.g., [55] [47], in particular [2] [98] for modeling human shape)) and the ease of collecting training samples.

However, in 2D models, the object shape variation and the perspective effect of the camera projection are mixed and nonseparable. In a typical surveillance setting as discussed in Sec.3.3), the perspective effect includes the change of overall shape w.r.t. the relative position on the ground plane to the camera (even after size normalization). If the shape is analyzed as a 2D quantity, variations from both sources are mixed therefore the constraints are not tight. Furthermore, the orientation of the object adds one more dimension. The 3D model with camera model contributes a compact representation. In using 3D models, the perspective effect will be taken into account automatically by following the camera projection process. The orientation only involves rotating the same 3D model.

We choose to use 3D shape model in our system. The advantage of using 3D model is amplified especially in the application of surveillance where the focal length and the deployment of the sensors may differ considerably according to various physical constraints. The focal length may range from wide angle (e.g., in a small room) to tele-lens (e.g., monitoring a site from a place far away). The tilt angle of the camera may range from
nearly zero to nearly 90 degrees. The same 3D model automatically work for all these cases while the view-based representation obviously doesn’t.

5.2.2 Multi-ellipsoid shape model

To represent a 3D shape, mesh models are commonly used in computer graphics mainly for rendering. The amount of detail needed and the processing speed of our application make them not first choices. A human body is articulated. The articulation has been heavily exploited both in computer animation and in computer vision (see Sec.2.2.5). The kinematics model (bone structure) usually has over 20 DOF at the major joints. Rendering or analysis (in “analysis by synthesis” approaches) also needs a volumetric model. Mesh model is used in computer animation and all kinds of parametric models (e.g., truncated cone [99], ellipsoid [27], super quadrics [100] and generalized cylinder [61]) are used in computer vision to approximate each limb mainly for computational simplicity and manipulability. In our case, the human motion is mostly limited to standing or walking and we do not attempt to capture the detailed shape and articulation parameters of the human body. Thus we use a small number of low dimensional models to capture the gross shape of human bodies at different representative postures.

In particular, we choose to use a standing human and two walking human models (left leg forward and right leg forward, respectively). Each of the models is a composition of a number of ellipsoids in fixed relations (i.e., no relative motion is allowed between them). An ellipsoid fits a human body part well and has the property that its projection is an ellipse with a convenient form. The standing model can also represent the human walking at phases when the two legs are not separated apart (i.e., when they are crossing). The
size of the ellipsoids is controlled by the height and the thickness parameters. Besides controlling the vertical scaling of the ellipsoids, the height parameter affects all the 3 dimensions and the thickness parameter captures the extra change. We expect that these models can provide sufficient approximations to the gross shape variations of most humans in the scene at the observed level of detail.

We assume that humans move on a ground plane, therefore, besides height and fatness, the parameters of the human object also include position on the ground plane and orientation. The orientations of the models are quantized into a small number of values for computation efficiency as follows: the standing model has two orientations (0°, frontal and 90°, from side) and each of the two walking models has six orientations (0°, ±30°, ±60° and 90°). The origin of the rotation is given so that 0° corresponds to human facing the camera. Choosing the number of quantization level, same as choosing the number of models, depends on the rule of keeping the number small while maintaining sufficient approximations. Their number can be increased if the required precision or computation power increases. Since both the model type and the orientation are discrete, we combine them into one parameter model/orientation label in one of the 14 combinations (Fig.5.2).

The 3D models assume that humans are perfectly upright, but there are chances that they incline their bodies slightly for various reasons (e.g., balancing or intentional bending). The inclination may need 2 parameters (forward/background and leftward/rightward) to capture in 3D. Considering the visibility after camera projection, we use one parameter to capture the inclination in 2D.
Figure 5.2: A number of multi-ellipsoid models are used to capture the gross shape of human standing and walking. (a) one standing model and two walking models; (b) the parameters of an object. From left to right: height, thickness, inclination, position, orientation; (c) the orientation quantization: standing model (with orientation $0^\circ$ and $90^\circ$); walking models (with orientation $0^\circ$, $\pm 30^\circ$, $\pm 60^\circ$ and $90^\circ$).

Therefore, the parameters of each human object $i$ are $m_i = \{l_i, x_i, y_i, h_i, f_i, i_i\}$ which are model/orientation label, position of the head\(^2\), height, fatness and inclination respectively. We assume that the camera model and the ground plane are known as described in Sec.3.3. The computation of the projection of the multi-ellipsoid models is explained in Appendix C.

It should be noted that we use the multi-ellipsoid models to control the model complexity while maintaining a reasonable level of fidelity. Other types of 3D models can replace them if necessary, without changing the overall approach. Now the models are hand-crafted. It will be interesting to investigate how such models can be learnt from 3D shape training data.

\(^2\)Position of the head in the image and a 3D height uniquely determines the human in 3D as the position of feet and height used in Chap.4. We use the position of head here because of its observability.
5.3 A Bayesian Formulation of Model-based Segmentation Problem

The Bayesian framework has been gaining extensive popularity as a perception model. The goal of visual perception is to interpret the world $W$ (i.e., the scene, objects, lighting) given the available image data $I$. This corresponds to estimating a posteriori probability distribution $P(W|I)$. The posterior distribution is determined in part by the image formation process, including the noise model, and in part by the structure (prior knowledge) of the world. Bayes’ rule provides the mechanism for combining these two factors into a final calculation of the posterior distribution, as in

$$P(W|I) = \frac{P(I|W)P(W)}{P(I)} \propto P(I|W)P(W) \quad (5.1)$$

The first term is called likelihood which encodes the image formation and the noise model and the second term is called prior distribution which encodes the prior structures of the world.

Bayesian approaches have enjoyed a great deal of recent success in their applications to various problems of computer vision. Many influential early works on machine and human perception models are included in [57]. The Bayesian formulation has been adapted in various areas in computer vision such as color consistency [5], motion/layer segmentation [84] [50], image segmentation [113], visual tracking [47], object recognition [103] and building modeling [26].
We formulate the model-based segmentation problem as computing the maximum a posteriori (MAP) estimation $\theta^*$ as

$$\theta^* = \arg\max_{\theta \in \Theta} P(\theta|I)$$  \hspace{1cm} (5.2)

where $\theta$ is the number of human objects and their parameters and $I$ contains the image observations. Following Bayes rule, the posterior probability is decomposed into a likelihood term and a prior term:

$$P(\theta|I) \propto P(I|\theta)P(\theta)$$  \hspace{1cm} (5.3)

The prior and the likelihood model are described below.

A state in the model-based segmentation problem includes the number of objects in the scene and their associated parameters. A state containing $n$ objects can be written as $\theta = \{m_1, m_2, ..., m_n\}_n \in \Theta_n$, where $\Theta_n$ is the solution space of exactly $n$ objects. It should be noted that $\theta$ is a set instead of a vector because its elements are not ordered. The entire solution space

$$\Theta = \bigcup_{n=0}^{\infty} \Theta_n, \quad \Theta_n = (L \times R^5)^n$$  \hspace{1cm} (5.4)

where $L$ is the enumerate space of the model/orientation ($|L| = 14$) and $R^5$ is the space for position and shape parameters ($m_i - \{l_i\}$). Since we don’t know in advance how many objects exist in the scene, the solution space $\Theta$ contains subspaces of varying dimensions, which creates difficulties in pursuing the solutions.
5.3.1 The prior probabilities

The prior probability of a state includes a prior probability on the number of objects in the scene, and a prior probability on the image size ($A_i$) and prior probabilities on parameters ($m_i$) of each object.

$$P(\theta) = P(n) \prod_{i=1}^{n} P(A_i)P(m_i).$$

(5.5)

The prior on the number of objects in the scene is modeled as an exponential distribution

$$P(n) \propto \exp\{-\lambda_1 n\},$$

(5.6)

which is used to penalize over segmentation. One can consider the situation that two completely overlapping objects have exactly the same likelihood as one object, obviously the one object solution is preferred due to the principle as stated in Occam’s Razor $^3$. $\lambda_1$ is an important parameter. It basically controls the amount of likelihood gain that an object has to contribute independently to make the overall posterior increase, subject to the changes of other prior probabilities. Similar penalty is widely used in image segmentation (e.g., in [113]) to control the number of regions.

The prior on the image size of an object ($A_i$) is related to the fact that objects with small image sizes are more likely to be due to noise.

$$P(A_i) \propto 1 - \exp\{-\lambda_2 A_i\},$$

(5.7)

$^3$English philosopher William of Occam (1284-1347) stressed the Aristotelian principle that entities must not be multiplied beyond what is necessary, which is known as the Ockham’s (Ockham’s) Razor.
where \( \exp\{-\lambda_2 A_i\} \) is the distribution of the noise blob size.

The prior on the model parameters is

\[
P(m_i) = P(l_i) P(x_i, y_i) P(h_i) P(f_i) P(i_i).
\]

We set \( P(l_i) \) so that \( P(l_i = \text{walk}) = \frac{2}{3} P(l_i = \text{stand}) \) to penalize the complexity (more ellipsoids used) of the walking models. \( P(x_i, y_i) \) is a uniform distribution in the image. \( P(h_i) \) is a Gaussian distribution \( \mathcal{N}(\mu_h, \sigma_h^2) \) truncated in the range of \([h_{\text{min}}, h_{\text{max}}]\) and \( P(f_i) \) is Gaussian distribution \( \mathcal{N}(\mu_f, \sigma_f^2) \) truncated in the range of \([f_{\text{min}}, f_{\text{max}}]\). \( P(i_i) \) is a Gaussian distribution \( \mathcal{N}(\mu_i, \sigma_i^2) \). Therefore

\[
P(m_i) \propto P(l_i) \exp\{-\frac{(h_i - \mu_h)^2}{\sigma_h^2}\} \exp\{-\frac{(f_i - \mu_f)^2}{\sigma_f^2}\} \exp\{-\frac{(i_i - \mu_i)^2}{\sigma_i^2}\}. \tag{5.9}
\]

In our experiments, we use \( \mu_h = 1.7 m, \sigma_h = 0.2 m, h_{\text{min}} = 1.5 m, h_{\text{max}} = 1.9 m; \mu_f = 1, \sigma_f = 0.2, f_{\text{min}} = 0.8, f_{\text{max}} = 1.2; \mu_i = 0, \sigma_i = 3 \) degrees. The parameters of the prior distributions can be changed reflecting the statistics of specific applications, for example, the variance and truncated range of height can be enlarged if children are expected in the scene. Other kinds of prior knowledge can also be added if available.

### 5.3.2 The image likelihood

The image likelihood \( P(I|\theta) \) reflects the probability that we observe image \( I \) (or some features extracted from \( I \)) given state \( \theta \). Here we exploit the use of two likelihood models: a foreground/background (F/B) model and a color model. The F/B model uses the
foreground mask from change detection as the observation. The color model directly uses the input frame as the observation, which potentially avoids the information loss in the change detection stage.

5.3.2.1 Choosing a likelihood model

There is no general way to choose the best likelihood model. For a specific application, the choice of a good likelihood model is still a combination of art and experience. The likelihood model can be learnt from training samples, however, what to learn (i.e. feature selection) is still the designer’s choice. The general principle is to choose the cue/feature that is the most discriminative compared to the rest of the scene. If a single cue/feature is not enough, multiple cues/features need to be combined.

The first factor to consider in designing a likelihood model is its optimality, i.e., the desired solution should give the highest value. Another factor which is also important but sometimes overlooked is that the likelihood should be realistically peaked. Unrealistically peaked (usually exaggerated) likelihood may create various problems, even if optimality is satisfied. It may overrule the contribution of the prior probabilities in the posterior probability. It will also create problems in applying two important statistical tools: MCMC and particle filter. It makes different modes of the solution space difficult to explore in MCMC since the likelihood values of different modes often differ by several orders of magnitude. It makes inefficient use of samples in a particle filter because the samples are usually concentrated around the dominant mode, impairing its strength of
keeping multiple hypotheses. The exaggeration sometimes comes from the violation of independence assumption in combining multiple cues or elements.

5.3.2.2 Mapping pixel sizes into 3D

The same object will have different image size in the projected images if its distance to the camera is different (see $h_1$ and $h_3$ in Fig. 3.1 for an example). Human perception does not assign likelihood values proportional to their image sizes. Instead, the perceived likelihood is more or less the same unless the size is too small to see clearly. Therefore, we want to obtain a size which is relatively “invariant” to its image size for the likelihood computation in the subsequent two subsections. It will provide a balanced likelihood values of objects of different image sizes in the joint likelihood. We do it by associating a “pseudo” 3D size with each pixel.

If we know the depth of the point in 3D (on a plane perpendicular to the viewing ray) which a pixel corresponds to, we will easily know the size of the 3D region corresponding to this pixel. The depth of a pixel is, of course, unknown. If we assume that the foreground pixels are created by human objects, we know that their 3D heights are in the range of human height. The knowledge of 3D height $h$ can be transformed into the knowledge of depth since the 3D point is fixed by the intersection of the ray given by the image point and the plane of $z = h$. Since the pixel can be of any 3D height in the range of human heights, we compute the expectation (average) of the 3D size considering all possible height values. Fig. 5.3 shows the idea graphically. We denote the 3D size that a pixel $j$
Figure 5.3: A graphical illustration of the computation of a “pseudo” 3D size of an image pixel.

Figure 5.4: The relationships between the different regions. (a) The object regions $S_i$ and the visible object regions $\tilde{S}_i$; (b) The entire solution region $S = S_1 \cap S_2 = \tilde{S}_1 + \tilde{S}_2$; (c) The foreground region $F$; (d) The relationship of $S$ and $F$, and the resulting four kinds of regions. NOTE: elliptic models are used for illustration.

corresponds to as $a_j$. They are computed once and stored at the beginning of the system initialization and used later as a look-up table.

5.3.2.3 Foreground/background (F/B) likelihood model

In this subsection, we describe the likelihood model based on the foreground mask computed from the change detection process (e.g. Fig.5.4.(c)). Denote $S_i$ as the region (a mask) occupied by object $i$ defined by $m_i$, and $\tilde{S}_i$ as the visible part of $S_i$ (Fig.5.4.(a)).
The entire solution region $S = \bigcup_{i=1}^{n} S_i = \sum_{i=1}^{n} \tilde{S}_i$, since $\tilde{S}_j$ are disjoint regions. We use $\mathcal{S}$ to denote the supplementary region of $S$ (Fig.5.4.(b)). In $\tilde{S}_i$, the ratio of background pixels (denoted as $p_{01,i}$) should be low. We weight $p_{01,i}$ with the 3D size of region $\tilde{S}_i$ (denoted as $\hat{A}_i$) to balance the contributions of objects with different visible sizes. We can defined the likelihood of region $\tilde{S}_i$ as

$$P_{fb}(I^{\tilde{S}_i}|m_i, \theta) \propto \exp\{-\lambda_{10}(p_{10,i}\hat{A}_i)\} = \exp\{-\lambda_{10}(\sum_{j \in \tilde{S}_i \cap F} a_j)\} \quad (5.10)$$

where $\tilde{S}_j \cap F$ is the region of background pixels in $\tilde{S}_j$. $\theta$ is used to determine the visible part of object $j$.

The likelihood of the entire solution region $S$ is

$$P_{fb}(I^{\tilde{S}}|\theta) = \prod_{i=1}^{n} P(I^{\tilde{S}_i}|m_i, \theta)$$

$$\quad \propto \exp\{-\lambda_{10}(\sum_{i=1}^{n} \sum_{j \in \tilde{S}_i \cap F} a_j)\}$$

$$\quad = \exp\{-\lambda_{10}(\sum_{j \in S \cap F} a_j)\}$$

$$\quad = \exp\{-\lambda_{10}A_{10}\} \quad (5.11)$$

where $A_{10}$ is the sum of the 3D sizes of all background pixels in region $S$. It is obvious that $P(I^{\tilde{S}}|\theta)$ only depends on the entire solution region $S$, not the individual visible regions $\tilde{S}_i$ (i.e., the likelihood is independent of the depth ordering of the objects). This depth-order invariance simplifies computation.
The likelihood of the image portion outside the solution region is

\[ P_{\theta}(I^S|\theta) \propto \exp\{-\lambda_{01}(\sum_{j \in S \cap F} a_j)\} = \exp\{-\lambda_{01}A_{01}\} \tag{5.12} \]

where \(A_{01}\) is the sum of the 3D sizes of all foreground pixels in region \(S\). Combining Equ.5.11 and Equ.5.12, we have the likelihood of the entire image

\[ P_{\theta}(I|\theta) = P(I^S|\theta)P(I^\overline{S}|\theta) \propto \exp\{-\lambda_{10}A_{10} - \lambda_{01}A_{01}\} \tag{5.13} \]

This is the joint likelihood which looks at the global interpretation of the entire image.

There should be as little background as possible in \(S\) and as little foreground as possible in \(\overline{S}\). We used likelihood \(P(I|\theta) \propto \exp\{-\lambda_{01}N_{01} - \lambda_{10}N_{10}\}\) as in our earlier work [123], where \(N_{01} = \sum_{i \in \overline{S} \cap F}\) and \(N_{10} = \sum_{i \in S \cap \overline{F}}\). Equ.5.13 can be thought of as its modification by weighting each pixel with its 3D size. This modification has resulted in better balance of objects of different image sizes in the joint likelihood.

By combining Equ.5.13 and Equ.5.5, we will have the posterior probability with the F/B likelihood. Parameters \(\lambda_{01}\) and \(\lambda_{10}\) can be estimated from training samples by counting. The relative value of \(\lambda_1\) to \(\lambda_{01}\), \(\lambda_{10}\) controls the amount of occlusion allowed. The optimality is only dependent on their relative values (if other parts of the prior probabilities are neglected), their absolute values should be given to ensure the overall probability is properly peaked to avoid the problems addressed in Sec.5.3.2.1. The algorithm of computing \(P_{\theta}(I|\theta)\) is summarized in Fig.5.6(a).
5.3.2.4 Color likelihood model

The foreground/background likelihood described above takes the binary mask from change detection as observation. The hard decision on foreground/background in change detection makes the quality of the mask sensitive to its threshold setting. Here we propose to use directly the color image input, which will be augmented and used in tracking in the next chapter (Fig. 5.5).

**Likelihood of an isolated object**

An object \(i\) is represented by a distribution of its color (color histogram) weighted by its distance to the object center. The likelihood of its being a foreground object is related to the difference of the color distribution within \(\tilde{S}_i\) (\(\tilde{S}_i = S_i\) for an isolated object) in the input frame \(\mathbf{p}\) and the color distribution within \(S_i\) in the background image \(\mathbf{d}\). The similarity of the two color histograms is defined based on the Bhattachayya coefficient as in [19]. Bhattachayya coefficient \(B_{\mathbf{p}, \mathbf{d}}\) of two distributions \(\mathbf{p}\) and \(\mathbf{d}\) with \(m\) bins is computed as

\[
B_{\mathbf{p}, \mathbf{d}} = \sum_{j=1}^{m} \sqrt{p_j d_j}. \tag{5.14}
\]

If \(\mathbf{p}\) and \(\mathbf{d}\) are identical, \(B_{\mathbf{p}, \mathbf{d}} = 1\); if \(\mathbf{p}\) and \(\mathbf{d}\) has no overlap at all, \(B_{\mathbf{p}, \mathbf{d}} = 0\). Such a metric has been used for color-based tracking in [19].

The likelihood of an object \(i\) to be a foreground object is to penalize the similarity to \(\mathbf{p}_i\) and \(\mathbf{d}_i\), weighted by its visible 3D area:

\[
P_c(I_\tilde{S}, \mathbf{m}_i) \propto \exp\{-\lambda_S B_{\mathbf{p}_i, \mathbf{d}_i}^2 \tilde{A}_i\} = \exp\{-\lambda_S \sum_{j \in \tilde{S}_i} B_{\mathbf{p}_i, \mathbf{d}_i} a_j\} \tag{5.15}
\]
Joint likelihood of multiple occluding objects and background

In case of multiple possibly overlapping objects, the likelihood of the object region

$$P_c(I^S|\theta) = \prod_{i=1}^{n} P_c(I^{S_i}|m_i, \theta) \propto \exp\{\lambda_S \sum_{i \in S_j} \sum_{i \in S_j} B^2_{p_i, d_i, a_j}\}$$  \hspace{1cm} (5.16)

The likelihood of the non-object region

$$P_c(I^{\overline{S}}|\theta) = \prod_{j \in \overline{S}} (P_b(I_j))^{\lambda_{\overline{S}_j}} \propto \exp\{-\lambda_{\overline{S}} \sum_{j \in \overline{S}} c_j a_j\}$$  \hspace{1cm} (5.17)

where $P_b(I_j)$ is the likelihood of an image pixel $I_j$ in its corresponding background model and $c_j = \log(P_b(I_j))$. The background model is a modified version of the Gaussian distribution learnt in Sec.3.2.2 with a wider tail to be more robust. Denote the $(\overline{\mu}_j, \sigma_j, \overline{b}_j)$ and $\sum_j = diag\{\sigma_{r_j}^2, \sigma_{g_j}^2, \sigma_{b_j}^2\}$ as the mean and the covariance of the background model at $j$-th pixel,

$$P_b(I_j) \propto \begin{cases} 
\exp\left\{-\frac{(r_j - \overline{\mu}_j)^2 + (g_j - \overline{\mu}_j)^2 + (b_j - \overline{b}_j)^2}{\sigma_j^2}\right\} & \text{if } (r_j - \overline{\mu}_j)^2 + (g_j - \overline{\mu}_j)^2 + (b_j - \overline{b}_j)^2 < k^2, \\ 
\exp\{-k^2\} & \text{otherwise.} 
\end{cases}$$  \hspace{1cm} (5.18)

The likelihood of the entire image

$$P_c(I|\theta) = P_c(I^S|\theta) P_c(I^{\overline{S}}|\theta) \propto \exp\{-\lambda_S E_S - \lambda_{\overline{S}} E_{\overline{S}}\}.$$  \hspace{1cm} (5.19)

\[\text{footnote: Optimizing this single object likelihood is equivalent to minimizing } B_{b_i, p_i}, \text{ which can be done efficiently using the mean-shift technique shown in Appendix D.2.}\]
Figure 5.5: The color likelihood model favors the differences of the input color histograms and the background color histogram of each visible object region (indicated by dark arrows) and penalizes the difference of the non-object region (indicated by an yellow arrow). Note that the elliptic models are used for illustration.

Different from the F/B likelihood, the visible part of each object has to be determined to compute its color histogram. In case of multiple overlapping objects, which part of an object is visible depends on the relative depth ordering of the objects. We infer the relative depth order by the feet positions of the human objects which are on the ground plane, as we did in Chap.4. The algorithm to compute $P_c(I|\theta)$ is summarized in Fig.5.6(b). It should be noted that the two algorithms to compute likelihood values in Fig.5.6 need to go through the entire image $I$ and all the objects in $\theta$. In Sec.5.5.1 we will introduce techniques for more efficient computation.

5.4 Computing the MAP Solution by Efficient MCMC

We want to find the state that maximizes the posterior probability defined in the previous section. However, as stated in Sec.5.3, the solution space contains subspaces of varying dimensions and may contain many local minimums. This makes the standard optimization or search techniques difficult to apply. The standard optimization techniques operates in a fixed dimension and solution space is too large for exhaustive search. Markov chain
Figure 5.6: The algorithms to compute the likelihood values. (a) $P_{bf}(I|\theta)$ and (b) $P_c(I|\theta)$.

Monte Carlo (MCMC) methods combined with jump-diffusion dynamics provide a way to sample the posterior probability in such a complex solution space to search for the maximum.

First we introduce the basic idea of MCMC. Then we describe the reversible dynamics that we designed for the multi-object segmentation problem. The domain knowledge that are used to compute the proposal probabilities follows next. Finally we summarize the algorithm.
5.4.1 Introduction of MCMC and its applications in computer vision

5.4.1.1 Markov chain Monte Carlo method to sample a probability distribution

A Markov chain can be designed to sample a probability distribution $Q(\theta)$ (in our case $Q(\theta) = P(\theta | I)$). At each iteration $g$, we sample a candidate state $\theta'$ according to $\theta_{g-1}$ from a proposal distribution $q(\theta_g | \theta_{g-1})$ (in simple words, what new state should the Markov chain go to from the previous state). The candidate state $\theta'$ is accepted with the following probability

$$p = \min \left\{ 1, \frac{Q(\theta')} {Q(\theta_{g-1})} \cdot \frac{q(\theta_{g-1} | \theta')} {q(\theta' | \theta_{g-1})} \right\}$$

The term $\frac{Q(\theta')} {Q(\theta_{g-1})}$ is the relative probability of the states $\theta'$ and $\theta_{g-1}$; the term $\frac{q(\theta_{g-1} | \theta')} {q(\theta' | \theta_{g-1})}$ reflects the relative difficulty of $\theta'$ going to $\theta_{g-1}$ and $\theta_{g-1}$ going to $\theta'$. If the probability of the proposed state, combined with the probability of going back to $\theta_{g-1}$, is greater than the reverse, the move is accepted; otherwise, it is still accepted probabilistically, which differs from gradient ascent. This is the well-known Metropolis-Hasting algorithm [111]. If the candidate state $\theta'$ is accepted, $\theta_g = \theta'$; otherwise, $\theta_g = \theta_{g-1}$. It can be proven that the Markov chain constructed this way has its stationary distribution equal to $Q()$, independent of the choice of the proposal probability $q()$ and the initial state $\theta_0$ [111]. This generalization and flexibility leave full freedom in designing algorithms for various applications. The proposal probability relates to the dynamics of the Markov chain. There are usually two kinds of dynamics: jump and diffusion. Jump refers to the motion of Markov chain between subspaces of different dimensions and diffusion refers to its motion within a subspace.
5.4.1.2 The use of MCMC in computer vision

There are a few reasons why MCMC method with jump/diffusion dynamics is suitable for many computer vision problems. Let’s assume that the problem is to compute the maximum of a posterior distribution, as in our case.

- Firstly, in many problems, the goal is to interpret the scene with some known stochastic processes (e.g., objects, regions, etc) of unknown number, unknown types and unknown parameters. This makes the structure (i.e., dimension) of the solution also a variable. The existing optimization techniques usually assume that the structure of the solution is fixed and only the parameters need to be optimized. However, MCMC with jump diffusion can traverse the different subspaces of solution space, exploring the different structures. An exhaustive search can also used to solve the problem, in theory, but the computation is usually prohibitive in practice. The stochastic search by MCMC may still achieve reasonable results given limited computation time.

- Secondly, the probability distributions of the solution to vision problems usually contain multiple modes, corresponding to ambiguities, distractions, etc. Common optimization technique can only pursue one of the modes, thus possibly getting trapped in a local minimum. The jumps of the Markov chain enable multiple modes to be explored.

- Thirdly, lots of bottom-up methods have been developed in many years of computer vision research. Data-driven MCMC (DDMCMC) has been proposed by Zhu and his collaborators to incorporate such domain knowledge as importance proposal
probabilities (mainly for jumps). Good proposals can lead the Markov chain to the places of the distribution with high densities (probabilities), therefore increase the efficiency significantly compared to traditional random proposals.

- Finally, with the continuous increase of computational power, the sampling-based approaches which were prohibitive can be made feasible with careful design now.

Zhu and his collaborators have shown the effectiveness of MCMC on a number of problems including a toy problem for object recognition (Zhu et al. [124]), image segmentation (Tu et al. [113]), range image segmentation (Han et al. [37]), combined object recognition and image segmentation (Tu et al. [114]), etc. Srivastava et al. [103] used MCMC with jump and diffusion to recognize 3-dimensional object in images. Forsyth et al. [31] used MCMC for structure from motion and color constancy. The MCMC approaches out-perform traditional approaches and also provide the uncertainties of the estimates. Dick et al. [26] used MCMC for building modeling. The Markov chain involves jumps of adding and removing building primitives (e.g., wall, window) and arranging them with domain knowledge. Sminchisescu et al. [100] used MCMC to solve the reflective ambiguity problem in tracking highly articulated human body which was not possible with an exhaustive search.

5.4.2 Markov chain dynamics to efficiently traverse the solution space

We design the following reversible dynamics for the Markov chain to explore the solution space. They are also shown graphically in Fig.5.7. Denote the state at iteration $g - 1$ as $\theta_{g-1} = \{m_1, \ldots, m_n\}_n$. We will sample a candidate state $\theta$ from the proposal
<table>
<thead>
<tr>
<th>addition</th>
<th>removal</th>
<th>split</th>
<th>merge</th>
<th>switch</th>
<th>diffusion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Diagram](figure_5.7.png)

Figure 5.7: The dynamics for multi-object segmentation.

probability $q(\theta'|\theta_{g-1})$. $p(\text{add})$, $p(\text{remove})$, $p(\text{split})$, $p(\text{merge})$, $p(\text{switch})$ and $p(\text{diff})$ are the probabilities are choose the respective dynamics. Their values sum up to one and are fixed in our implementation.

### 5.4.2.1 Human hypothesis addition

We sample the parameters of a new human hypothesis $m_{n+1}$ from $q_{a}(m)$ and add the object to $\theta_{g-1}$. $\theta' = \{m_1, \ldots, m_n, m_{n+1}\}_{n+1}$.

$$q(\theta'|\theta_{g-1}) = p(\text{add})q_{a}(m_{n+1}) = p(\text{add})q_{a}(x, y)q_{a}(h)q_{a}(f)q_{a}(l)q_{a}(i)$$

(5.21)
5.4.2.2 Human hypothesis removal

Select an existing human hypothesis \( r \in [1,n] \) with probability \( q_r(r) \) to remove. We currently use a uniform distribution on \( q_r() \). \( \theta' = \{ \mathbf{m}_1, ..., \mathbf{m}_n \}_n - \{ \mathbf{m}_r \}_1. \\

\[
q(\theta'|\theta_{g-1}) = p(\text{remove})q_r(r) = p(\text{remove})(1/n)
\] (5.22)

5.4.2.3 Human hypothesis split

Split and merge (described next) are needed in the situation where heavy occlusion is present. The same effect can be achieved by combinations of addition/removal/diffusion, but using split/merge can make the exploration of the solution space more efficient. If only slight occlusion is expected (or allowed), the gain of using split/merge may be small.

Select a human hypothesis \( r \in [1,n] \) with probability \( q(r) \). Split it into two hypotheses \( r_1 \) and \( r_2 \) with parameters \( \mathbf{m}_{r_1} \) and \( \mathbf{m}_{r_2} \) respectively. \( \theta' = \{ \mathbf{m}_1, ..., \mathbf{m}_n \}_n - \{ \mathbf{m}_r \}_1 + \{ \mathbf{m}_{r_1}, \mathbf{m}_{r_2} \}_2 \)

\[
q(\theta'|\theta_{g-1}) = p(\text{split})q(r)q(\mathbf{m}_{r_1}, \mathbf{m}_{r_2}|\mathbf{m}_r)
\] (5.23)

5.4.2.4 Human hypotheses merge

Select two overlapped human hypotheses \( r_1, r_2 \in [1,n] \) and merge them into one object \( r \) with parameters \( \mathbf{m}_r \). \( \theta' = \{ \mathbf{m}_1, ..., \mathbf{m}_n \}_n - \{ \mathbf{m}_{r_1}, \mathbf{m}_{r_2} \}_2 + \{ \mathbf{m}_r \}_1 \)

\[
q(\theta'|\theta_{g-1}) = p(\text{merge})q_m(r_1, r_2)q_m(\mathbf{m}_r|\mathbf{m}_{r_1}, \mathbf{m}_{r_2})
\] (5.24)
We chose to sample \((r_1, r_2)\) according to its 3D overlapping area

\[
q_m(r_1, r_2) \propto \sum_{j \in S_{r_1} \cap S_{r_2}} a_j
\]  

(5.25)

5.4.2.5 Model/orientation switch

Switch the model/orientation label (discrete variable) of a human hypothesis. Randomly select an existing human hypothesis \(r \in [1, n]\) with \(q_w(r)\) and randomly switch the model/orientation label \(l_r\) to another one with \(q_w(l_r|r)\). All other parameters are inherited. Both \(q_w()\) and \(q_w(|r)\) are uniform distributions. It is only applied to those objects whose lower part is visible because most of the difference of the models is in the lower part.

\[
q(\theta'|\theta_{g-1}) = p(\text{switch})q_w(r)q_w(l_r|r)
\]  

(5.26)

5.4.2.6 Stochastic diffusion of model parameters

Update the continuous parameters of an object. Randomly select an existing human hypothesis \(r \in [1, n]\) with \(q_d(r)\), and update its continuous parameters (i.e. \(\mathbf{m}_r = \mathbf{m}_r - l_r\)).

\[
q(\theta'|\theta_{g-1}) = p(\text{diff})q_d(r)q_d(\mathbf{m}_r' | \mathbf{m}_r)
\]  

(5.27)

where \(q_d(r)\) is a uniform distribution.

Different to previous dynamics, the acceptance of \(\theta'\) in stochastic diffusion may or may not rely on the Metropolis-Hasting rule, as will be discussed in Sec. 5.4.3.7.
The first four dynamics (i.e. addition, removal, split and merge) are referred to as jump dynamics and the rest are referred to as diffusion dynamics. It is guaranteed that the Markov chain designed this way is ergodic (i.e., any state is reachable from any other state within a finite number of iterations) and aperiodic (i.e., the Markov chain does not oscillate in a fixed pattern) since all of moves are stochastic [113]. In the four jumps, addition and removal are sufficient, split and merge are added for the ease of traversal in the solution space.

5.4.3 Informed proposal probabilities

In theory, the proposal probability $q()$ does not affect convergence. However, different $q()$ lead to different performances. The speed of the Markov chain strongly depends on the proposal probabilities, for example, where to add a new object ($q_a(x, y)$). We use the following domain knowledge to compute these proposal probabilities to make the Markov chain more intelligent and have a higher acceptance rate.

5.4.3.1 Object addition: Head candidates from foreground boundaries

This method detects the heads which are on the boundary of the foreground as used in Sec.4.2. The basic idea is to find the local vertical peaks along the boundary. The peaks are further filtered by checking if there are enough foreground pixels below them according to the human height range and the camera model. It also outputs a candidate height for each head candidate. This detector has a high detection rate and is also effective when the human is small and image edges are not reliable; however, it cannot detect the heads
in the interior of the foreground blobs. Fig.5.8.(a) shows the boundary head candidates on the example frame.

We denote the set of boundary head candidates and their associated heights as \( \mathcal{HC}_b = \{(x_1, y_1, h_1), \ldots, (x_{N_{HC}_b}, y_{N_{HC}_b}, h_{N_{HC}_b})\} \). The proposal probability of \( x, y \) and \( h \) is

\[
q_{a1}(x, y, h) \propto \sum_{i=1}^{N_{HC}_b} \mathcal{N}((x_i, y_i), \text{diag} \{\sigma_x^2, \sigma_y^2\})(P(h) + \mathcal{N}(h, \sigma_h^2))
\]

(5.28)

where \( P(h) \) is the prior distribution of human height as defined in Equ.5.9. Rest of the parameters are sampled from their respective prior distributions.

5.4.3.2 Object addition: Head candidates from intensity edges

Here we describe a head detector based on image intensity edges, which is also effective for the heads in the interior of the blobs. First a Canny edge detector [12] is applied to the dilated foreground region of the input image, which is assumed to include all the “real” foreground regions. A distance transformation [11] is computed on the edge map. Fig.5.8.(b) shows the exponential edge map where \( E(x, y) = \exp(-\lambda D(x, y)) \) \((D(x, y)\) is the distance to the closest edge point and \( \lambda \) is a factor to control the response field and set to 0.25.). Besides, the coordinates of the closest pixel point are also recorded as \( \vec{C}(x, y) \). The unit image gradient vector \( \vec{O}(x, y) \) is only computed at edge pixels.

The “\( \Omega \)” shape of head and shoulder contour (Fig.5.8.(c)) is easily derived from our human model. The head-shoulder contour is generated from the projected ellipses by taking the whole head and the upper quarter torso as the shoulder. The normals of the contour points are also computed. The size of the human model is determined by
the camera calibration assuming a known height (1.6, 1.7, 1.8 meters are used and the maximum response is recorded).

Denote by \( \{\vec{m}_1, ..., \vec{m}_k\} \) and \( \{\vec{v}_1, ..., \vec{v}_k\} \) the positions and the unit normals of the model points respectively when head top is at \((x, y)\). The model is matched with the image in the following way.

\[
S(x, y) = \frac{1}{k} \sum_{i=1}^{k} e^{-\lambda D(\vec{m}_i)} (\vec{v}_i \cdot \vec{O}(\vec{C}(\vec{m}_i)))
\]

Figure 5.8: Head candidate detectors. (a) Head candidates (shown by crosses) from foreground boundaries \((HC_b)\); (b) Distance transformation on Canny edge detection result; (c) The head-shoulder \(\Omega\) model: dark contour-head and shoulder, light line-normals; (d) Head candidates (shown by crosses) from intensity edges \((HC_r)\).
A head candidate map is constructed by evaluating $S(x,y)$ on every pixel in the dilated foreground region. After smoothing it, we find all the peaks above a threshold. The threshold is selected to give a very high detection rate but may also result in a high false alarm rate. An example is shown in Fig.5.8.(d). The false alarms tend to happen in the areas of rich texture where there are abundant edges of various orientations.

We denote the set of edge head candidates and their associated heights of maximum responses as $HC_e = \{(x_1, y_1, h_1), \ldots, (x_{N_{HC_e}}, y_{N_{HC_e}}, h_{N_{HC_e}})\}$. Similar to $q_{a1}$, The proposal probability is

$$q_{a2}(x, y, h) \propto \sum_{i=1}^{N_{HC_e}} (N((x_i, y_i), \text{diag}\{\sigma_x^2, \sigma_y^2\}))(P(h) + N(h_i, \sigma_h^2))$$  \hspace{1cm} (5.30)

Rest of the parameters are sampled from their respective prior distributions.

### 5.4.3.3 Object addition: Connected component analysis of foreground residue

We denote the foreground map with the already formed hypotheses removed as the foreground residue map ($Fr$ as in Chap.4). The foreground may contain some blobs of isolated human objects. Furthermore, after some human objects are hypothesized and removed from the foreground, the foreground residue map may become more isolated. Morphological “open” operation can help isolate the objects that are only connected by thin bridges and remove small/thin residues. We generate human candidates from the foreground residue map as the following.

---

6For computational reasons, $S(x,y)$ may be evaluated on a coarser grid.
Given the foreground $F$ and the region $S$ corresponding to the current solution $\theta$, compute the foreground residue map $Fr = F \cap \overline{S}$. Perform morphological “open” operation with a vertically elongated structural element (e.g., height=5, width=3) and compute connected components (CCs). The proposal probability for the position is

$$q_{03}(x, y) \propto A_i \sum_{i=1}^{3} \mathcal{N}(x_{i,j}, y_{i,j}, \text{diag}\{\sigma_x^2, \sigma_y^2\}), \quad (5.31)$$

where $A_i$ is the 3D size of the connected component $i$, $(x_{i1}, y_{i1})$, $(x_{i2}, y_{i2})$ and $(x_{i3}, y_{i3})$ are positions obtained respectively by: the centroid of $CC_i$ is aligned with the center of human body; the top center point of $CC_i$ is aligned with the human head; and the bottom center point of $CC_i$ is aligned with the human feet. Rest of the parameters are sampled from their respective prior distributions.

Although the foreground residue map can be computed incrementally (as will be shown in Sec.5.5.1), the morphological operation and the connected component analysis is still expensive. Therefore, we consider the following method based on projection analysis.

### 5.4.3.4 Object addition: Projection analysis of foreground residue

Suppose we have the vertical projection $Vp$ of the current foreground residue map $Fr$ smoothed with a kernel whose size equals to average human width in image:

$$Vp(x) = \left(\sum_{y=1}^{h} Fr(x, y) a_{x,y}\right) \oplus w \quad (5.32)$$
We sample \( x_r \) from \( Vp \) as the \( x \) coordinate of the human object with probability \( q_{o4}(x) \propto Vp(x) \). After \( x_r \) is fixed, we compute the horizontal projection \( Hp \) within a band whose \( x \) coordinate is around \( x_r \):

\[
Hp(y) = \sum_{x=x_r-w}^{x_r+w} Fr(x,y)a_{x,y}
\]  

(5.33)

We sample the vertical location of the object in one of the following three ways (shown in Fig. 5.9)

\[
q_{o4}(y|x) = \frac{1}{3}q_{o4,1}(y|x) + \frac{1}{3}q_{o4,2}(y|x) + \frac{1}{3}q_{o4,3}(y|x)
\]  

(5.34)

The rest of the parameters are sampled from their respective prior distributions.

1. Sample the head. We compute the positive gradient of \( Hp \):

\[
G_{H_p}^+(y) = \begin{cases} 
Hp'(y) & \text{if } Hp'(y) > 0 \\
0 & \text{otherwise.}
\end{cases}
\]  

(5.35)

And sample \( y_r \) from \( q_{o4,1}(y) \propto G_{H_p}^+(y) \) as the \( y \) coordinate of the head.

2. Sample the feet. We choose to sample the \( y \) coordinate of the feet with probability \( q_{o4,2} \). We compute the negative gradient of \( Hp \):

\[
G_{H_p}^-(y) = \begin{cases} 
-Hp'(y) & \text{if } Hp'(y) < 0 \\
0 & \text{otherwise.}
\end{cases}
\]  

(5.36)

And sample \( y_r \) from \( q_{o4,2}(y) \propto G_{H_p}^-(y) \) as the \( y \) coordinate of the feet. The coordinate of the feet is transformed to the corresponding coordinate of the head.
Figure 5.9: Generating human hypothesis using foreground residue projection analysis. First \(x_r\) is sampled from vertical projection \(Vp(x)\), then \(y_r\) is sampled from the band horizontal projection around \(x_r\) - \(Hp(y)\), positive gradient of \(Hp(y) - G_H^+(y)\), or negative gradient of \(Hp(y) - G_H^-(y)\).

3. Sample the center. We smooth \(Hp\) with a kernel whose size is determined by the image height of an average 3D height at \(y\) (assuming a small yaw angle), and sample \(y_r\) from \(q_{a_4,3}(y) \propto Hp(y)\) as the \(y\) coordinate of the center of the body.

Projection analysis is a common tool in document analysis. Vertical projection analysis was used in [39] for segmenting connected human objects. However, it assumed that each peak in the vertical projection corresponds to the head of one human object, which is obviously not valid for our data.

The four ways of adding a human object (Sec.5.4.3.1-5.4.3.4) may seem redundant. However, the redundancy increases robustness in case some fail in certain conditions.

5.4.3.5 Object split

To split an object \(r\) into two, first we sample the direction of the split by the foreground residue around the object \(r\). As shown in Fig.5.10, the direction is quantized into a
Figure 5.10: Sample the direction of a split according to the histogram of the foreground residue around the object. (a) quantized split directions, (b) foreground residue histogram of the directions. Note that an elliptic model is used for illustration.

A histogram of direction is constructed by accumulating the foreground residue within twice the size of the object. The split direction (angle $\alpha$) is sampled from the histogram. The parameters of the two new objects are given as the following.

Having the split direction fixed, the object $r$ is slid in the direction and the likelihood within the object is recorded in $R(d)$ where $d$ is the offset to the original position in the split direction. We sample $d_1$ from $R(d), d > 0$ and $d_2$ from $R(d), d < 0$. Then the positions of the two new objects are:

\[
\begin{align*}
  x_{ri} &= x_r + d_i \cos \alpha \\
  y_{ri} &= y_r + d_i \sin \alpha 
\end{align*}
\]

The sizes of the new objects are slightly reduced according to the split direction. The heights of the two new objects $h_{r1}$ and $h_{r2}$ are sampled from $\mathcal{N}((1 - 0.1 \cos \alpha) h_r, \sigma_h^2)$; the thickness of the two new objects $f_{r1}$ and $f_{r2}$ are sampled from $\mathcal{N}((1 - 0.1 \sin \alpha) f_r, \sigma_f^2)$. The rest of the parameters are inherited.
5.4.3.6 Object merge

Merging two objects are easier compared to splitting. We sample its position around the average position of \( r_1 \) and \( r_2 \). The height and the thickness are slightly enlarged according to the relative position of the two objects being merged.

\[
q_m(x_r, y_r | x_{r_1}, y_{r_1}, x_{r_2}, y_{r_2}) = \mathcal{N}\left(\frac{A_1 x_{r_1} + A_2 x_{r_2}}{A_{r_1} + A_{r_2}}, \frac{A_1 y_{r_1} + A_2 y_{r_2}}{A_{r_1} + A_{r_2}}, \Sigma\right)
\]  
(5.38)

\[
q_m(h_r) = \mathcal{N}\left(\frac{A_{r_1} h_{r_1} + A_{r_2} h_{r_2}}{A_{r_1} + A_{r_2}}(1 + 0.1 \cos \alpha), \sigma_h^2\right)
\]  
(5.39)

\[
q_m(f_r) = \mathcal{N}\left(\frac{A_{r_1} f_{r_1} + A_{r_2} f_{r_2}}{A_{r_1} + A_{r_2}}(1 + 0.1 \sin \alpha), \sigma_f^2\right)
\]  
(5.40)

where \( \alpha \) is the angle decided by the centroids of the two objects being merged (similar to the split direction above).

5.4.3.7 Object parameters diffusion

We could choose one of the following strategies to compute \( q_d(\bar{m}_r | \bar{m}_r) \):

**Random sampling** Pure random sampling can be used (as in [122]).

\[
\bar{m}' \leftarrow \bar{m} + w
\]  
(5.41)

where \( w \) is zero mean Gaussian noise. This is inefficient due to its blind nature.

The random moves are symmetrical and therefore balanced.
Stochastic gradient ascent The parameters can be updated in the direction of their gradients in the posterior plus random noise (as in [123]), as

\[ \hat{\mathbf{m}} \leftarrow \hat{\mathbf{m}} - k \frac{dE}{d\hat{\mathbf{m}}} + \mathbf{w} \]  \hspace{1cm} (5.42)

where \( E = -\log P(\theta|I) \) is called an energy function, \( k \) is a coefficient to control the step size. The noise helps avoid local maxima. The parameters are also bound to their minimum and maximum allowed values. It is shown in Geman et al. [35] that randomly selecting an object and following Eqn. 5.42 will lead to a Markov chain with the stationary distribution of \( P(\theta|I) \) (i.e., the Gibbs sampler). If this is used, the probabilistic acceptance of Metropolis-Hasting does not need to be applied.

In practice, the gradient of the energy function can not be computed analytically in our definition of the likelihood model. Computing it numerically involves \( 2 \) times of the number of parameters evaluations of the posterior probability. Besides, value of \( k \) needs to be chosen carefully to keep the gradient descent procedure well-behaved.

Stochastic mean shift for position Mean shift is a technique for mode seeking in a non-parametric distribution [14] [18], which has been used for object tracking represented with a color histogram ([19], Appendix D.1). It can be used here to compute the gradient of the likelihood of visible region of an object \( P(I^S|m_i, \theta) \) w.r.t. its position, which is an approximation of the gradient of the joint-likelihood. The mean shift is advantageous due to its computational efficiency and the stability.

A. Mean shift for F/B likelihood.

For F/B likelihood, the rule to update the position for an isolated object can be
regarded as a special case of the original mean shift tracker (given in Appendix D.1 in Eq(D.8)). We modify it for overlapping objects by giving less weight to the overlapped pixels (\( \frac{1}{L(j)} \)), the more overlapping layers, the less weight).

\[
u' \leftarrow \frac{\sum_{j \in S_i \cap F} \frac{1}{L(j)} u_j}{\sum_{j \in S_i \cap F} \frac{1}{L(j)}} + w.
\]  

(5.43)

B. Mean shift for color likelihood.

For color likelihood, since we know the visible part of each object, we perform the mean shift only on its visible part. The position update rule to optimize Eq(5.15) is

\[
u' = \frac{\sum_{j \in \hat{S}_i} u_j w_j}{\sum_{j \in \hat{S}_i} \left| w_j \right|} + w, 
\]  

(5.44)

where

\[
w_j = \sum_{u=1}^{m} \left( -\sqrt{\frac{d_u(y_0)}{p_u(y_0)}} \delta[b_f(u_j) - u] - \sqrt{\frac{p_u(y_0)}{d_u(y_0)}} \delta[b_b(u_j) - u] \right).
\]  

(5.45)

The derivation is given in Appendix D.2.

The mean shift algorithm only provides the gradient of the likelihood of individual objects without a global view of all the other objects and the background process.

Therefore we add some Gaussian noise and still use the Metropolis-Hasting rule to accept the proposed position change to simulate a Gibbs sampler. The mean shift only updates the position, the rest of the parameters (i.e., height, fatness, inclination) have to be updated with either Eq(5.41) or Eq(5.42).
5.4.4 Summary of the algorithm

In this subsection, we summarize the algorithm of the MCMC model-based segmentation.

The block diagram is shown in Fig. 5.11. At each iteration, a candidate state $\theta'$ is generated according to $\theta_{g-1}$ by the methods described in Sec.5.4.2. The values of $q(\theta'|\theta_{g-1})$ and $q(\theta_{g-1}|\theta')$ are also computed for jumps. The posterior probability of the proposed candidate state $P(\theta'|I)$ is computed. The candidate state is accepted by the Metropolis Hasting rule as in Equ.5.20.

This iterative process starts from an initial state (Sec.5.5 shows that the approach is independent to the selection of initial state.). After a number ($M$) of iterations called burn-in period, the samples $\theta_g$ become independent of the initial state and can be regarded as unbiased samples from the posterior probability. The state corresponding to
the maximum posterior value up to the current iteration is recorded and it becomes the solution when the given number \((N)\) of iterations is reached. The iterations needed to obtain satisfactory results depends on the complexity of the scene. More iterations are needed for a scene containing more humans and more occlusion.

Besides the MAP estimation, we can use the samples to compute the expectations of some parameters. However, given that the objects in the state \(\theta\) have no identities, we only can compute the expectations of the quantities independent of the object identities\(^7\), which is limited compared to the state with identities as in the next chapter.

5.5 Implementation and Results

We have tested the approaches described above on a number of data sets and the results are consistently good. In this section, we describe some computational issues and then show the results as well as performance evaluation.

5.5.1 Incremental computation

In one iteration of the algorithm, only one or two objects are changed. Thus the new likelihood can be computed more efficiently by incrementally computing it only within the neighborhood of the area associated with the changed objects and those overlapping with it. Although a joint state and joint likelihood are used, the computation involved in one iteration is reduced to a small local region. This is in contrast to particle filters

\(^7\)One such expectation is the average number of objects in the scene: \(E[n] = \frac{1}{N} \sum_{y=1}^{N} n_y \).
where the likelihood evaluation of each particle (joint state) needs the computation of
the full joint likelihood because the samples are not correlated.

In case of the F/B likelihood, the incremental likelihood computation is only needed
within the region of the object being changed. This requires to buffer the object layer
map $L$ ($L(i)$ means the number of object layers at pixel $i$, as in the algorithm to compute
the full likelihood). The algorithms for the incremental updates for object addition and
removal are summarized in Fig.5.12. In case of split, merge or diffusion, $A_{01}$ and $A_{10}$ can
be simply computed by a composition of additions and removals. The foreground residue
map $Fr$, and the vertical projection $Vp$ also get updated incrementally at the same time
(not shown in the algorithm).

\[
\begin{array}{|l|l|}
\hline
\text{object } i \text{ is just added} & \text{object } i \text{ is to be removed} \\
\text{for all } j \in S_i & \text{for all } j \in S_i \\
\quad \text{if } L(j) = 0 \text{ AND } F(j) = 0 & \quad \text{if } L(j) = 0 \text{ AND } F(j) = 0 \\
\quad A_{10} \leftarrow A_{10} + a_j; & \quad A_{10} \leftarrow A_{10} - a_j; \\
\quad \text{if } L(j) = 0 \text{ AND } F(j) = 1 & \quad \text{if } L(j) = 0 \text{ AND } F(j) = 1 \\
\quad A_{01} \leftarrow A_{01} - a_j; & \quad A_{01} \leftarrow A_{01} + a_j; \\
\quad L(j) \leftarrow L(j) + 1; & \\
\quad P_{fB} \leftarrow \exp\{-\lambda_{01}A_{01} - \lambda_{10}A_{10}\}; & \text{(a)} \\
\hline
\end{array}
\]

Figure 5.12: The incremental computation of the F/B likelihood for (a) object addition
and (b) object removal.

Incremental computation of color likelihood is slightly more complex since depth order
has to be considered. Visibility index map $V$ needs to be buffered ($V(j) = i$ means
that object $i$ is visible at pixel $j$; $-1$ represents background). The algorithms of the
updates of adding and removing an object are summarized in Fig.5.13. Adding an object
only involves computation within the new object region. Removing an object $i$ involves

99
Figure 5.13: The incremental computation of the color likelihood for (a) object addition and (b) object removal.

computation within the region of the object \( i \) and the objects it occluded. This is because the visibility indices of the pixels covered by the object \( i \) need to be fixed.

5.5.2 Results

We perform change detection as described in Sec. 3.2.2. The foreground mask is filtered with morphology “close” operation for computing proposal probabilities. The unfiltered foreground mask is used to compute the F/B likelihood. The Markov chain starts from a null state \( \theta_0 = \emptyset \). The results are shown in input and output pairs. The outputs are the human models overlaid on the original images. Due to the small image size, we have explicitly marked the errors on the images when they occur: false alarms in black arrows and missed detections in white arrows.
**Example 1: results on a sequence in Chap.4.** We applied our method to the data used in Chap.4 and it works unsurprisingly well. We show the results of a few frames in Fig.5.14. Compared to the segmentation results in Chap.4, the results obtained here are usually more accurate with all the parameters being optimized. Besides, the shadow is also involved in the optimization therefore improves the accuracy of localization. Compared to Fig.4.4(d), a larger portion of the shadow pixels are recognized correctly (marked in red in Fig.5.14(c)). 300 iterations per frame were used to generate the results.

**Example 2: sequence “Topping”.** This is a 605-frame sequence captured from a camera on second floor with the camera tilt angle around 20°. A group of 22 humans walked through the scene. The image sizes of human objects in the scene have a large variation (more than 10 times in area). The dense edges of the tree branch shadows at the far side result in high false positives for head candidates. There are 50 to 100 head candidates per frame. Results on some frames of “Topping” are shown in Fig.5.15. The results were obtained by 2000-iteration runs.

**Example 3: sequence “Commons”.** This is a 900-frame sequence captured from a camera above a building gate with the camera tilt angle = 40°. A large tilt angle results in significant perspective effect on human shape in images. 33 humans passed by the scene with 23 going out of and 10 going in the building. Both Example 2 and 3 contains persistent overlapping involving large number of people, which is very challenging to a detection/tracking system. Results on a few frames are shown in Fig.5.16. They were obtained by 1000-iteration runs.

**Example 4: an indoor sequence.** Fig.5.17 shows the results of an indoor sequence. This sequence has a low viewpoint (camera mounted on a tripod on the ground) and
the inter-occlusion is more severe. The average number of head candidates is around 100. 5000 iterations are used for each frame⁸. The people in the scene cause significant illumination changes both on the carpet and on the pillars which causes the change detection method to give erroneous foreground regions. We manually removed those false foreground detections to isolate the difficulty of human segmentation. The missed detections either have too few pixels or are almost completely overlapped with other humans.

**Example 5: an airborne video sequence with very high noise level.** Fig.5.18 shows the result of our segmentation algorithm on an infra-red airborne video after ego-motion compensation (by Kang [86]). The main difficulty here is the high noise (false alarm of the change detection) level of the input. There are both random noise and large persistent structural noises. Some blobs due to structural noise are larger than the human blobs. Even though the number of people is small in this example, simple methods such as noise filtering and connected components analysis are not sufficient to successfully detect the people. Our method detects the two people in large part of the frames and produces very few false alarms. 100 iterations are used for each frame and the system runs in real-time.

**Experiment 1: informed proposals v.s. random proposal.** To verify the benefit of the use of informed proposals, we did the following experiment. We compare the convergence of the system with the proposal probabilities described in Sec.5.4.3 and a random proposal probability (uniform distribution over the image) for human position

---

⁸The results were generated using random diffusion only. The use of gradient information will reduced the number of iterations needed.
when adding new human hypotheses. The log posterior probability histories of the two cases in 5000-iteration runs on the third pane in Fig. 5.15 are shown in Fig. 5.19.(a). The log posterior probability of our approach climbs quickly and then stays close to maximum value for fine adjustments while the log posterior probability of the random proposal probability improves significantly more slowly.

**Experiment 2: sensitivity to initial state.** Our proposed approach is not sensitive to the initial state. We show the log posterior probability histories of 1000-iteration runs from a null initial state and an initial state containing 20 random humans on the foreground in Fig. 5.19.(b). Although showing some difference at the beginning, they show little difference after 400 iterations.

From our experiments, we did not observe significantly better results of the color likelihood over the F/B likelihood with carefully tuned change detection result. However, in a more general setting where a fixed threshold is applied in the change detection, the use of color likelihood gives better results. The color likelihood is interesting in that it will be extended for tracking in the next chapter.

### 5.5.3 Performance evaluation

To evaluate the accuracy of the method quantitatively, we compare our results on sequence "Topping" and "Commons" with ground truth derived from hand annotation. If a human object in the solution has an over 50% overlap with a human object in the ground truth, a match is declared. One-to-one mapping of the objects is enforced. The unmatched objects in the solution are declared as false alarms and the unmatched objects in the
Table 5.1: Results of performance evaluations on “Topping” and “Commons”.

<table>
<thead>
<tr>
<th></th>
<th>“Topping”</th>
<th>“Commons”</th>
</tr>
</thead>
<tbody>
<tr>
<td>valid humans</td>
<td>8466</td>
<td>6726</td>
</tr>
<tr>
<td>correct detections</td>
<td>7881</td>
<td>6243</td>
</tr>
<tr>
<td>missed-detections</td>
<td>585</td>
<td>483</td>
</tr>
<tr>
<td>false alarms</td>
<td>291</td>
<td>12</td>
</tr>
<tr>
<td>detection rate</td>
<td>93.09%</td>
<td>92.82%</td>
</tr>
<tr>
<td>false alarm rate</td>
<td>3.43%</td>
<td>0.18%</td>
</tr>
</tbody>
</table>

ground truth are declared as miss-detections. The humans whose bodies are partially outside the scene are not counted as matches or errors (i.e., they are “don’t cares”).

To make the evaluation results more meaningful, we characterize the complexity of the dataset for human segmentation by the number of human objects per foreground blob. Usually more humans per blob imply greater challenge to the segmentation algorithm. Fig.5.20 shows the histogram of the occurrence that a human object is in a blob of $n$ human objects. In sequence “Topping” 50% of the humans appear in a blob containing 5 or more humans and 30% of the humans appear in a blob containing 9 or more humans. In sequence “Commons”, about 30% of the humans appear in a blob containing 5 or more humans. We have not seen any reported results on data of similar complexity.

The results of the evaluation are summarized in Tab.5.1. In both sequences, $\sim 93\%$ detection rate is achieved with relatively low false alarm rate. Please note that better results are achieved for less complex data (e.g. in example 1). We observed from the experiments that most of the missed detections are due to the failure of the change detection algorithm when the human’s clothing has color similar to the background or the overlapping humans form ambiguous shape or a combination of the two. The false alarms are mainly due to the inclusion of large (relative to a human’s size) non-human regions.
in the foreground. These errors can be reduced by utilizing other image cues or temporal information (e.g. tracking as shown in Chap.6). Some of the selected model/orientations are not accurate. They are mainly due to the error of the foreground around the feet. From the experiments we performed, we found that our proposed method is insensitive to blob fragmentation if a significant amount of foreground indicates the presence of a human/humans. It also works robustly when there are significant amount of noises in the foreground.

5.5.4 Computation

The total computation time for processing a frame is a product of the number of iterations needed and the computation time of each iteration. The examples of Seq.1 and Seq.2 require 1000 to 2000 iterations to get stable results. However, scenes containing less people require substantially less number of iterations. The computation time per iteration is affected by the likelihood we choose and the average image size of the people. The F/B likelihood takes less time for evaluation than the color likelihood. This results in twice the running speed of using the F/B likelihood than the color likelihood for the same number of iterations. A 1000-iteration run on the above reported dataset using F/B likelihood requires about 0.5 seconds of CPU time on a Pentium IV 2.7G Hz PC, with un-optimized C++ code. If there is a small number of people in the scene, the system runs in real-time.

Although showing significant speedup compared to uninformed MCMC, the computation required is still heavy for complex scenes. The possible optimization in computation can be made through the following aspects.
**Efficient mask operation** Current we are using integer operation, less computation is required if we use bit-wise operation.

**Efficient object mask generation** Generating object masks is essential in the system performance. Current we evaluate the quadratic equation to check if a point is in an ellipse or not, which can be replaced with method similar to the polygon filling algorithm [43]).

**Parallel processing** A Markov “chain” seems sequential, however, this framework can be made faster if parallel processing is available. First, each of the different ways of making proposals can be made as one process. Ideally, all of them are active all the time to compete for the proposal according to the possible change to the posterior probability (“proposal of the proposal”). The different ways of making proposals can be thought as *attentions* and they compete for the most salient one. Second, multiple Markov chains can be used instead of a single chain. There can be interactions between the multiple chains similar to the crossovers in Genetic Algorithms. For example, instead of having a single chain running 2000 iterations, we can have 10 chains each running 200 iterations. The exchange of partial good states among multiple chains will make the Markov chains burn-in faster than a single chain.

Besides, techniques such as multi-resolution processing can also be used.
5.5.5 Remarks

5.5.5.1 Comparison with the simple segmentation in Sec.4.2

The segmentation technique described in Sec.4.2 was used successfully on the data where a small number of people walked in a group. In Sec.4.2, a simple model is used and the parameters are not optimized: the position is fixed directly by the head candidate, the height is computed by heuristics and others are fixed by average values. In this chapter, more complex models are used and the choice of model and their parameters are optimized for all the objects according to a single object function.

The techniques in Sec.4.2 also see their contributions in this chapter (Sec.5.4.3.1 and Sec.5.4.3.3). However, instead of directly forming solutions, they are used to make hypotheses (suggestions), the hypotheses are accepted according to the change of a globally-defined posterior probability. This is very important because the proposals are usually computed locally and based on partially knowledge (cues), therefore do not necessarily lead to a globally optimal solution on their own.

In simple situations, the domain knowledge itself could be sufficient to give satisfactory results, and the gain of a formal Bayesian formulation and joint likelihood is small and may not worth its demand for computation power. In more complex situations, the domain knowledge alone is more fragile, and the approach by formal Bayesian analysis based on the image formation process is more robust.
5.5.5.2  A unifying framework integrating top-down and bottom-up processing

In summary, the Bayesian-MCMC approach described in this chapter is a general framework which integrates both top-down and bottom-up processing. First, the Bayesian posterior probability is defined according to the solution to be computed and the associated image formation process. In theory, the solution can be computed from the posterior probability. In practice, the computation can be infeasible due to the large and complex solution space. Fortunately, there are lots of existing bottom-up techniques developed over the many years of computer vision research. Using them alone does not guarantee optimality because most of them are computed locally and using partial cues. However, they are useful in pointing out the possible peaks of the posterior distributions or provide approximate direction of improvement. Therefore, using the results of various heterogeneous bottom-up processing as the proposal probabilities for the Markov chain in an MCMC computational engine both takes advantages of the computational efficiency of the bottom-up techniques and retains the optimality and robustness of the Bayesian formulation from the global view (top-down).
Figure 5.14: Segmentation results on sequence seq1 used in Chap.4. (a) Segmentation on selected frames; (b) Foreground mask of the frame in 3rd pane; (c) Pixel classification associated with the solution in 3rd pane: blue-$S \cap F$, yellow-$S \cap \overline{F}$, black-$\overline{S} \cap F$ (excluding shadow), red-shadow.
Figure 5.15: Segmentation results of sequence “Topping”. Left column: input; right column: output. False alarms are marked with dark arrows and miss detections are marked with white arrows (same for the following figures).
Figure 5.16: Segmentation results of sequence “Commons”. Left column: input; right column: output.
Figure 5.17: Segmentation results of an indoor sequence. (a) The foreground mask and the head candidates by intensity edges, (b) The results shown by semi-transparent color masks.
Figure 5.18: Segmentation results of a noisy infra-red airborne video sequence after ego-motion compensation. (a) original frame with foreground overlaid in red; (b) detection results overlaid on enlarged portion of (a).

Figure 5.19: Experiments on comparing informed proposals with random proposals and comparing different initial states. (a) $\log P(\theta|I)$ of using random proposals climbs significantly slower compared to informed proposals; (b) The difference of $\log P(\theta|I)$ from different initial states are indistinguishable after a number (400) of iterations.
Figure 5.20: The histogram of the number of human objects per blob in sequence “Topping” (a) and “Commons” (b).
Chapter 6

Model-based Tracking: a Bayesian Approach

The previous chapter addressed the problems of “how many people are in the scene and where are they” in each frame. The segmentation result can be used for counting people. However, recovering the trajectories of multiple people as they move in the scene, i.e. tracking, is even more useful. With their trajectories, we can analyze the dynamic behavior of each individual, the interactions of multiple people, as well as group behaviors.

Given the Bayesian framework for segmentation, tracking is a natural extension. The joint state is augmented with object identities. Temporal prior probabilities are added for object correspondences and consistency of their parameters. Consistency of object appearance is also enforced through temporal likelihood.

6.1 Introduction

Tracking can be either viewed as establishing the temporal correspondence of the objects in different frames or as estimating the trajectory of each object in the scene. The first view applies more often in detection-based tracking and the second one applies more often in matching-based tracking.
A naïve way to extend segmentation to tracking is to follow the theme of detection-based tracking: perform segmentation in each frame independently and then make correspondences of the segmented human objects across frames, in the same way as a multi-blob tracker [16]. This approach has the drawback that the segmentation process does not take advantage of the temporal relationship of the same object in adjacent frames. Given the speed of normal human locomotion and temporal sampling rate of 15-30 frames per second, the change of the state (e.g., position, height) and the appearance of a human object in two adjacent frames is usually small. Inefficient use of information requires more computation.

Matching-based tracking can make better use of information when the detection problem is difficult and is the approach that we take. The multi-object tracking approach we describe here is a natural extension of the segmentation approach we presented in the previous chapter. The framework remains the same, however some components are augmented or modified due to the temporal information involved in tracking. The joint state includes the number of objects, their associated parameters and augmented with their identities linking to the previous frame. The posterior probability is defined considering the prior probabilities, the temporal prior and the likelihood. The temporal prior enforces object correspondences and smoothness of its track. Besides the difference of the object color histogram with the background, the new color likelihood also includes its similarity to its corresponding object in the previous frame, which gives more discrimination to different objects. The MAP estimate is also computed by MCMC. Besides the Markov chain dynamics used before, we added identity related dynamics. New human hypotheses can also be created by the estimates from the previous frame. The block diagram is shown in
Figure 6.1: The block diagram for MCMC model-based tracking. The differences (complete or partial) from the approach for MCMC model-based segmentation (Fig. 5.11) is highlighted in gray. (X in the diagram is the same as θ in the text.)

Fig. 6.1 with the differences highlighted. Uncertainty of the estimates are also computed during the sampling process and is used in the Kalman filters for optimal filtering. We show a better results compared to the those obtained only by segmentation.

6.2 The Posterior Probability for Multi-object Tracking

The state θ is augmented from the state used for segmentation in Chap. 5 with the identity (ID) of each object as θ = {m₁, ..., mₙ} = {(k₁, m₁), ..., (kₙ, mₙ)}ₙ, where kᵢ is an ID of an object i. The two states θ₁ = {0, m₀}, (1, m₁)₂ and θ₁ = {0, m₀}, (1, m₀)₂ are different. We use braced superscript to denote the frame number and subscript to denote
the iteration. For example, $\hat{\theta}^{(t)}_g$ means the $g$-th sample (iteration) in the $t$-th frame. In case of no ambiguity, they may be omitted for simplicity.

The posterior probability of $\theta^{(t)}$ is given by

$$P(\theta^{(t)}|I^{(t)}, \theta^{(t-1)}) \propto P(I^{(t)}|\theta^{(t)})P(\theta^{(t)})P(\theta^{(t)}|\theta^{(t-1)})$$

(6.1)

The prior probability $P(\theta^{(t)})$ follows the same definition as Eqn.5.5 in Chap.5. The temporal prior $P(\theta^{(t)}|\theta^{(t-1)})$ reflects the smoothness and the connectivity of the trajectories. $P(I^{(t)}|\theta^{(t)})$ reflects the likelihood of observing image $I^{(t)}$ given the objects and their appearances.

6.2.1 The temporal prior

For the convenience of expression of the temporal prior, we re-arrange $\theta^{(t)}$ and $\theta^{(t-1)}$ as $\hat{\theta}^{(t)} = \{m_1^{(t)}, \ldots, m_N^{(t)}\}_N$ and $\tilde{\theta}^{(t-1)} = \{\tilde{m}_1^{(t)}, \ldots, \tilde{m}_N^{(t-1)}\}_N$ so that one of $k_i^t = k_i^{t-1}, \tilde{m}_i^{(t)} = \phi, \tilde{m}_i^{(t-1)} = \phi$ is true, where $N = |\theta^{(t)} \cup \theta^{(t-1)}|$.

The temporal prior

$$P(\theta^{(t)}|\theta^{(t-1)}) = \prod_{i=1}^{N} P(\tilde{m}_i^{(t)}|\tilde{m}_i^{(t-1)}).$$

(6.2)

The temporal prior of each object follows the definition

$$P(m_i^{(t)}|\tilde{m}_i^{(t-1)}) = \begin{cases} P((k_i^{(t)}, m_i^{(t)}))(k_i^{(t-1)}, m_i^{(t-1)}) = P(m_i^{(t)}|\tilde{m}_i^{(t-1)}), \\ P((k_i^{(t)}, m_i^{(t)})|\phi) = P_{\text{appear}}(m_i^{(t)}), \\ P(\phi|(k_i^{(t-1)}, m_i^{(t-1)})) = P_{\text{disappear}}(m_i^{(t-1)}). \end{cases}$$

(6.3)
We assume that the position and the inclination of an object follow a constant velocity models with Gaussian noise and the height and thickness follow Gaussian distributions. Therefore we use Kalman filters for temporal estimation (same as in Chap.4).

\[
P(m_i^{(t)}|m_i^{(t-1)}) = P(u_i^{(t)}|u_i^{(t-1)})P(h_i^{(t)}|h_i^{(t-1)})P(w_i^{(t)}|w_i^{(t-1)})P(\theta_i^{(t)}|\theta_i^{(t-1)})
\]

\[
\propto \exp\left\{ (u_i^{(t)} - \mu)^T \Sigma^{-1} (u_i^{(t)} - \mu) \right\} \exp\left\{ (h_i^{(t)} - \overline{h})^2 \right\} \exp\left\{ (w_i^{(t)} - \overline{w})^2 \right\} \exp\left\{ (\theta_i^{(t)} - \overline{\theta})^2 \right\}
\]

(6.4)

where \((\mu, \Sigma), (\overline{h}, \sigma_{\overline{h}}^2), (\overline{w}, \sigma_{\overline{w}}^2), (\overline{\theta}, \sigma_{\overline{\theta}}^2)\) are the predicted mean and variance (covariance matrix) of the position, height, width and inclination of the object \(i\) from their respective Kalman filters.

\(P_{\text{appear}}(m_i^{(t)}) = P_{\text{appear}}(u_i^{(t)})\) and \(P_{\text{disappear}}(m_i^{(t-1)}) = P_{\text{disappear}}(u_i^{(t-1)})\) are the penalties of the initialization of a new track and the termination of an existing track respectively. They are set according to the distance of object to the entrances/exits (see Sec.4.4.1) empirically. The probabilities are high when the object is close to the entrances/exits and vice versa.

6.2.2 Color-based likelihood

The likelihood (Sec.5.3.2) for segmentation defined in the previous chapter does not involve object specific representation of appearance because the analysis is only for one frame. When the objects are tracked continuously, a representation can be constructed
for each object so that the matching can be done with less ambiguity. Here we use a color histogram as the representation.

For an object whose parameter is \( m_i \) with a correspondence in the previous frame, we evaluate its likelihood as

\[
P(\tilde{I}^i | m_i) \propto \exp\{ -\lambda_0 B_{\tilde{p}_i, d_i}^2 A_i + \lambda_1 B_{\tilde{p}_i, \hat{A}_i}^2 \}
\]

where \( \tilde{p}_i \) is the color histogram of the object with the same ID in the previous frame and \( B \) is the Bhattacharyya coefficient of two histograms. This likelihood both penalizes the similarity with the background and favors the similarity with it correspondence in the previous frame. We call the two terms background exclusion and object attraction respectively. Compared to the color likelihood used in segmentation (Equation 5.15), the object attraction term is new, which enforces consistency of the object appearance and reduces the ambiguity. For an object without a correspondence (i.e. a new object), only the background exclusion part is used.

The likelihood of the non-solution region \( P(\tilde{I}^c | \theta) \) is the same as Equation 5.17 in Chapter 5. The posterior probability for the entire image is assembled accordingly. The posterior probability includes components for both detection (i.e. as in the previous chapter) and tracking (i.e. the temporal prior and temporal likelihood). This leads to simultaneous detection and tracking where new objects are detected and the previous detected objects are tracked.
6.3 Computing the MAP with Uncertainty

The maximum of the posterior probability is computed in the same fashion as in Chap. 5 using Markov chain Monte Carlo. However, there are a few augmentations to reflect the temporal prior and likelihood.

6.3.1 Augmentations to Markov chain dynamics

We design the following identity-related dynamics (shown graphically in Fig. 6.2) to augment those defined in Sec. 5.4.2. Again we assume that we have the sample in \((g-1)\)-th iteration \(\theta_{g-1}^{(t)}\) and now propose a candidate \(\theta'\) for the \(g\)-th iteration.
6.3.1.1 Establish correspondence

Establish the correspondence of a new object in $\theta^{(l)}_{g-1}$ and a dead object (an object which does not have a correspondence in the current state) in $\theta^{(l-1)}$. 

Randomly select $r \in [1, n]$ where $k_r$ not in $\theta^{(l-1)}$ and $r' \in \theta^{(l-1)}$ where $k^{(l-1)}_{r'}$ not in $\theta^{(l)}_{g-1}$. $\theta^{(l)}_{g-1} = \{\ldots, (k_r, m_r), \ldots\}_n \rightarrow \theta' = \{\ldots, (k^{(l-1)}_{r'}, m_r), \ldots\}_n$.

$$q(\theta'|\theta_{g-1}) = p(establish)q_{r1}(r, r')$$ (6.6)

We choose $q_{r2}(r, r') \propto \frac{1}{\|u_r - u_{r'}\|^2}$ for all the qualifying pairs.

6.3.1.2 Break correspondence

Break the correspondence of an object in $\theta^{(l)}_{g-1}$ with an object in $\theta^{(l-1)}$.

Randomly select $r \in [1, n]$ where $k_r$ is in $\theta^{(l-1)}$ and change $k_r$ to a new object (and same object in $\theta^{(l-1)}$ becomes dead). We use a uniform distribution to sample $r$.

$$q(\theta'|\theta_{g-1}) = p(break)\frac{1}{n'}$$ (6.7)

where $n'$ is the number of objects in $\theta^{(l)}_{g-1}$ that have correspondences in the previous frame.
6.3.1.3 Identities switch

Exchange the IDs of two close-by objects. Randomly select two objects \( r_1, r_2 \in [1, n] \) and exchange their IDs.

\[
\theta_{g-1} = \{ (k_{r_1}, m_{r_1}), \ldots, (k_{r_2}, m_{r_2}) \ldots \} \rightarrow \theta' = \{ (k_{r_2}, m_{r_1}), \ldots, (k_{r_1}, m_{r_2}) \ldots \}.
\]

\[
q(\theta' | \theta_{g-1}) = p(\text{exchange})q_{x_1}(r_1, r_2)
\]

(6.8)

We choose \( q_{x_1}(r_1, r_2) \propto \frac{1}{\|w_{r_1} - w_{r_2}\|} \). Note that we don’t require that \( r_1 \) and \( r_2 \) have correspondences in the previous frame. Identities switch can also be realized by the compositions of the first two dynamics. This is used to ease the traversal since establishing and breaking correspondences may lead to a big decrease in the probability and are less likely to be accepted.

6.3.2 Augmentations to proposal probabilities

6.3.2.1 Object addition: from estimate of previous frame

Besides the hypotheses generating methods of Sec.5.4.3.1-Sec.5.4.3.4, we also generate new hypotheses from the objects in \( \overline{\theta}^{(l)} \) (prediction of the current frame from previous frame). Different from the use of head candidate sets, we enforce that each object in \( \overline{\theta}^{(l)} \) can correspond to only one hypothesis in \( \theta^{(l)} \).

\[
q_{\alpha_5}(\mathbf{m}) \propto \sum_{i=1}^{N_{H\mathcal{C}_i}} \{ \mathcal{N}(\mathbf{m}_i, \mathbf{\Sigma}_i) \mathcal{N}(\hat{h}_i, \sigma_{\hat{h}}^2) \mathcal{N}(\hat{f}_i, \sigma_{\hat{f}}^2) \mathcal{N}(\tilde{\mathbf{r}}_i, \sigma_{\tilde{\mathbf{r}}_i}^2) P(l) \},
\]

(6.9)
where $\mathbf{u}_i$ and $\mathbf{\Sigma}_i$ are predicted mean and covariance matrix of the parameters as in the temporal prior (Equ. 6.4).

### 6.3.2.2 Proposing position update

Compared to segmentation, the mean shift used to propose diffusion now also needs to consider the object attraction term. Suppose the weight for each location $x_i$ is $w_i^b$ and $w_i^f$ for optimizing the background exclusion ((1) in Equ. 6.5) and object attraction ((2) in Equ. 6.5) individually. Due to the linearity of the mean shift technique, a new weight which will optimize the two at the same time ((1)+(2)) is $w_i = w_i^b + w_i^f$. (also see Appendix D.3)

$$u' = \frac{\sum_{j \in \hat{S}_i} u_j w_j^b}{\sum_{j \in \hat{S}_i} |w_j^b|} + w, \quad (6.10)$$

### 6.3.3 Temporal filtering with adaptive measurement noise

Since the Markov chain generates unbiased samples from the posterior distribution of $\theta^{(t)}$, besides obtaining the MAP estimation, we can also compute other statistics from the samples. Unlike in segmentation, the objects here have identities; therefore we can compute object specific statistics.

Assume we have $N - M$ samples of $\{\theta_{M+1}, \ldots, \theta_N\}$ from the posterior probability of $\theta^{(t)}$, with the first $M$ samples discarded. Object $k$ only appeared in $N_k$ of the $N$ samples $\{\theta_{k1}, \ldots, \theta_{kN_k}\}$. The expectation related to object $k$ can be computed as

$$E[f(m)] = \int f(m)P(m)dm \sim \frac{1}{N_k} \sum_{i=1}^{N_k} f(m_{ki}) \quad (6.11)$$
We compute the mean \( f(\mathbf{m}) = \mathbf{m} \) and the covariance (or variance, \( f(\mathbf{m}) = (\mathbf{m} - \mathbf{m})(\mathbf{m} - \mathbf{m}^T) \)) of the position, height, thickness and inclination.

We use Kalman filters to filter and predict the states of each object. We use a Kalman filter for each of the position, height, thickness, and inclination parameters. We assume that each quantity follows a constant velocity law as in Sec. 4.3. The previous use of Kalman filters usually assumes that the measurement noise is fixed (estimated or empirically given). In Sec. 4.3, we made the measurement noise adaptive according to the geometric transformation from 2D to 3D, but it does not embody the real measurement uncertainty given the image observations. Here, the covariance of the real measurement noise is estimated by the samples from the posterior probability distribution.

We have noticed in our experiments that the occurrences of tracking errors or highly ambiguous situations are usually accompanied by a noticeable larger covariance. This is a good indication that high-level analysis (e.g., trajectory-based analysis) should step in.

### 6.4 Implementation and Results

The above approaches are used to augment the system in Chap. 5 to perform tracking. In processing each frame, we choose the initial state to be a predicted state (the parameters of each object predicted by their Kalman filters) instead of a null state as in the segmentation. The Markov chain has more chances to perform diffusions and less to perform jumps since less structural change is expected. We use less number of iterations per frame compared to segmentation in the previous chapter. For example, we use 300 iterations
per frame for sequence “commons”, in contrast to 1000 iterations as in the segmentation. Since the temporal coherence resolves some ambiguities, less burden is put on accurate shape modeling, therefore we only use the standing model in tracking.

We show in Fig.6.3 the selected frames from the result of sequence “commons”. The identities of the objects are show by their ID numbers displayed on the head. The results show more consistency than the results obtained by segmentation. For example, the three people in dark clothes walking in a group (ID 11, 22, 23) are sometimes segmented as 3 people and sometimes segmented as 2 people in the previous results, here they have 3 trajectories once they are fully in the scene. Tracking also handles temporary severe occlusions such as object with ID 44 in frame 661.

We perform the frame-based evaluation following the same protocol as described in Sec.5.5.3 to compare the results with those obtained only with segmentation. The detection rate and the false alarm rate is 98.13% and 0.27% respectively. It can be compared with the evaluation result of the segmentation of the same sequence (92.82% detection rate and 0.18% false alarm rate, as in Tab.5.1). With tracking, the detection rate increased by a large margin while the false alarm rate only increased a little bit (still very low). We also evaluate the results by the trajectory-based errors. Trajectories whose lengths are less than 10 frames are discarded in our results and therefore not counted in the evaluation. Among the 33 human objects, trajectories of 3 objects are broken once (ID 28→ID 35, ID 31→ID 32, ID 30→ID 41, all between frame 387 and frame 447, as marked with colored arrows in the images), and the rest of the trajectories are correct. The trajectory-based error rate is 9.1%. Usually the trajectories are initialized once the humans are fully in the scene, some even start when the objects are partially in. Only
the initializations of three objects (objects 31, 50, 52) are noticeably delayed (by 50, 55, 60 frames respectively after they are fully in the scene). Partial occlusion (objects 31, 50, 52) or/and lack of contrast with the background (object 31, 52) are the causes of the delays.

6.4.1 Remarks

6.4.1.1 Comparison with tracking approach in Sec.4.3

The relationship of the approach described in this chapter and the approach in Chap.4 is summarized as below.

- They both use object shape model, object appearance and constraints provided by camera model. The object shape model used in Sec.4.3 is very simple and the parameter (i.e. height) is fixed; in this chapter it has more parameters which are continuously optimized. Sec.4.3 uses a template to model appearance, here we use color histograms. Efficient algorithms exist to optimize the likelihood based on color histograms.

- Due to computational reason, the likelihood in Sec.4.3 is factored into that of each individual objects. The single object likelihood is carefully designed to take into account some information of the background and other objects, but it is not enough in crowded situations like Fig.6.3. In this chapter, we use the joint likelihood of the entire image (all the objects and the background). The best solution is computed by a more powerful (but slower) computational engine.
Figure 6.3: Selected frames of the tracking results from “commons”. The numbers on the heads show identities. (Please note that the two people who are sitting on two sides are in the background model, therefore not detected.)
6.4.1.2 Comparison with particle filter-based tracking

There has been work for tracking multiple humans using particle filter [109] and [48]. They bear some similarities as well as differences from our approach as discussed below.

- They are similar in that both make use of a joint likelihood of all objects, even though the design of the likelihood may be different.

- Particle filter tries to keep a distribution of the solution using a non-parametric technique (i.e., with a large number of samples). The evolution of such a distribution is made possible by the “group behavior” of all the samples (i.e., the history of a sample does not necessarily correspond to a consistent trajectory of an object). Therefore, it is difficult to incorporate an appearance model which is useful in tracking (e.g. the color histogram that we use). The techniques of [109] and [48] are similar to tracking using F/B likelihood model in this thesis. Using appropriate appearance models can resolve many ambiguities of using a generic foreground model.

- Particle filter is known to suffer from the dimensionality problem due to the poor scalability of non-parametric approaches in high dimension. The number of samples needed usually grows exponentially with the dimensions. The work of [109] and [48] only show results for up to 4 humans in the scene. Applying the same technique to scenes containing more people may be non-trivial. In contrast, our method explores the solution space with a single Markov chain for explicit optimization, which is less sensitive to the dimension of the space.
• The sequential nature of MCMC can make more in-depth analysis of the solution distribution. Various data-driven techniques can be incorporated in our framework which may not be suitable for the one-pass sampling in particle filtering [47]. [85] used a hybrid Monte Carlo filter to combine a particle filter with MCMC, which is equivalent of the use of multiple Markov chains in MCMC. However, the MCMC was only for diffusion in their work.

• The particle filter-based approaches were proposed to resolve temporary ambiguities in tracking with its multiple-hypotheses nature. Our approach does not keep multiple hypotheses. However, joint likelihood and object appearance are used to make the ambiguities minimal, and the covariance analysis of the samples can get an estimate of the uncertainty. We will reply on higher-level processing (trajectory analysis) to resolve remaining ambiguities.
Chapter 7

3D Tracking of Human Locomotion: a Tracking as Recognition Approach

In this chapter, we present an approach, which we call tracking as recognition, to robustly estimate the 3D human body postures during locomotion (i.e., walking, running, standing). It makes use of strong prior knowledge of the motion to resolve the ambiguities of single view, temporary occlusion and low image resolution. The experiments show very promising results and the approach is expected to apply to other types of motions.

7.1 Introduction and Motivation

Some applications require more detailed information of the body postures in addition to the positions and orientations. 3D human body posture is commonly represented by major joint angle values in a kinematics model of over 20 degree of freedoms. Some previous work has been summarized in Sec.2.2.5. A common way to recover them from video sequences is by tracking: to estimate the state at time \( t \) assuming the state at time \( t - 1 \) is known. This formulation may face a number of difficulties, which makes it sensitive to realistically
noisy situations. Firstly, the track may drift when the observations (measurements) are ambiguous; such ambiguity is common when tracking a high dimension model using two dimensional image observations. Secondly, the prior knowledge about the motion can only be used very locally for prediction (e.g., as in [99]). Finally, tracking requires knowledge of initial state; this is a difficult problem and a user often specifies it manually.

These difficulties call for stronger prior knowledge on the motion being studied and stronger temporal integration. We propose here a tracking as recognition approach where the estimation of body postures is accomplished by recognition in a locomotion model over a period of time. By doing so, short-term ambiguities of measurements are resolved by considering all measurements globally. Initial state is estimated in the same way as all the other frames by inference.

Human locomotion contains many modes, among which walking, running and standing are the three most common ones. Our locomotion model is composed of three modes: walking, running and standing. Walking and running are dynamic modes and are further decomposed into a number of phases corresponding to characteristic 3D body postures. The model is in the form of a hierarchical finite state machine. The states are matched with the image observations by the motion flow, which provides robust results especially when the humans are small in the image. The inference is done efficiently by the Viterbi algorithm [89]. Besides the body postures, high-level descriptions (i.e., the modes) are obtained concurrently. In Chap.4, the human objects were hypothesized only by shape analysis which could produce false alarm in case of non-human foreground region with a human-like shape. We use this technique to verify these human objects to see if they exhibit proper motion pattern.
7.2 The Locomotion Model

Human locomotion has many modes, among which walking, running and standing are the three most common ones in daily life. A human can switch between these modes. The relationship of the modes is naturally represented as a finite state machine shown in Fig.7.1.(a). The speed of the body is an important feature to help distinguish among these three modes. The modes and the associated feature constitute the first level of our hierarchical locomotion model. The transition probabilities between the modes are given by their observed frequencies. The prior probability distribution of the speed given the modes $P(v|m), m \in \{walk, run, stand\}$ is set according to previous research in [3].

A more detailed model is needed to estimate the motion of the limbs of walking and running. Walking and running are both periodic motions. We define a cycle to be the minimum repetitive unit, which equals two steps. Each cycle of the mode is represented by a number of characteristic phases. The phases are the second level of the locomotion model (Fig.7.1.(b)).

For each of walking and running modes, several 3D motion capture sequences are gathered to compute an average cycle by aligning the phase when right leg crosses the left leg while moving forward. 3D motion capture data, consisting of a human kinematics model and a sequence of joint angle values, is a concise representation of motion of an articulated body and such data are now easily available. The average cycle trajectories are uniformly quantized into 16 discrete phases. The quantization level is chosen considering the response property of the likelihood measurements. The body postures corresponding to the phases of walking and running are shown in Fig.7.2.
Figure 7.1: The human locomotion model. (a) the first level of hierarchy; (b) the second level of hierarchy (self-cycles omitted to save space). Notations: dotted ellipses and arrows-high level model; solid circles and arrows-low level model (specifically, dark circle-starting state; double circle-ending state); solid ellipses and solid lines-feature and its association.

Figure 7.2: The 2D projections of the 3D body postures corresponding to the phases of walking and running.

Each phase corresponds to a state. A state has a link back to itself (i.e., self-cycle, not shown in Fig.7.1.(b) for simplicity), a link to the state of next phase and a link to the state of phase after next (i.e., by-pass link, e.g., the link from $R_1$ to $R_9$) to handle fast motion. The start and end states in the low level state machine are defined so that the transition between modes can only happen at the start and end of each step. The transition probabilities within each mode are set according to average walking/running cycle duration. The initial state probabilities are set to be uniform for all possible states.
7.3 The Observations

A similarity measure of the states and the image observations is needed for inference. Here we propose to use a motion template which is the template of model flow or image optical flow. It encodes both the shape and motion information. Other possibilities include edges [94], foreground mask and moments of the foreground mask [6]. These static features are less discriminative since they do not contain motion information. Edge information may be noisy when the objects are small. Foreground mask and its moments are sensitive to the inclusion of background pixels. Moments cannot be used when a human is overlapped with other moving objects. Motion templates are advantageous in these aspects, therefore it is suitable as a representation of instantaneous motion in our case. Besides, its response field is relatively wide so that fewer quantization steps (e.g., 16 for each locomotion mode) are needed. It is different from the Motion Energy Image (MEI) and Motion History Image (MHI) introduced in [10] which are templates of accumulated motion over a period of time. We choose to use only the motion of the legs since it is salient and stable compared to the motion of other parts (e.g., arms).

The motion template of the model can be easily computed from the 3D motion data given a camera model and 3D position and orientation of the human (as shown in Fig 7.3(a)). We assume that the human faces forward in the direction of its motion. The kinematics model is rescaled according to the human’s physical height. The motion model is placed at the given position facing the given orientation. Forward kinematics [76] is used to compute 3D positions of all joints and the camera model projects the 3D positions into 2D image positions.
With the 2D positions of the joints, the motion templates are computed as follows. For simplicity, we only show the procedure for one segment of the leg. We approximate the projected shape of the segment as symmetrical (equicrural) trapezoid. Rotation along the limb direction is not considered since its effect is hardly visible in the image. Suppose we wish to compute the model flow at frame \( t \) with backward differencing. Let points \( \mathbf{a}_t, \mathbf{a}_{t-1} \) and \( \mathbf{b}_t, \mathbf{b}_{t-1} \) be the projected 2D positions of two ends of a limb segment in frame \( t \) and \( t-1 \) respectively. Their motions are \( \Delta \mathbf{a} = \mathbf{a}_t - \mathbf{a}_{t-1}, \Delta \mathbf{b} = \mathbf{b}_t - \mathbf{b}_{t-1} \).

A point \( \mathbf{p}_{t-1} \) in frame \( t-1 \) is transformed to its correspondence \( \mathbf{p}_t \) in frame \( t \) in the following way (Fig.7.3.(b)). First it is translated by \( \Delta \mathbf{a} \) (\( T_{\text{translation}} \)), then it is rotated around \( \mathbf{a} \) by \( \theta \) (\( T_{\text{rotation}} \)), and then it is stretched along the direction of \( \mathbf{a}_t \mathbf{b}_t \) by a factor of \( s \) (\( T_{\text{stretch}} \)). \( \theta \) is computed by cosine rule and \( s = |\mathbf{a}_t \mathbf{b}_t|/|\mathbf{a}_{t-1} \mathbf{b}_{t-1}| \). Therefore, the motion of the point \( \mathbf{p} \) is computed by Eqn.7.1.

\[
\Delta \mathbf{p} = \mathbf{p}_t - \mathbf{p}_{t-1} = \mathbf{p}_t - T^{-1}(\mathbf{p}_t),
\]

(7.1)

where \( T \) is defined as in

\[
\mathbf{p}_t = T(\mathbf{p}_{t-1}) = T_{\text{stretch}} T_{\text{rotation}} T_{\text{translation}}(\mathbf{p}_{t-1}).
\]

(7.2)

The two legs may occlude each other, therefore depth order is considered in generating the templates. 32 model motion templates (16 for walking and 16 for running) are generated this way in each frame. Image motion or optical flow is computed only for
the foreground objects from the incoming frames to avoid unnecessary computation. A block matching based optical flow algorithm (in [110]) is employed.

We define normalized template distance to be the normalized sum of the vector differences of the two templates given by

\[
D = \sum_{i \in \Omega} \frac{|v_i - m_i|}{|v_i||m_i|}
\]  \hspace{1cm} (7.3)

where \( \Omega \) is the area within the bounding box of both legs, \( v_i \) and \( m_i \) are the vectors of image optical flow and the predicted motion template at \( i \) respectively. The distances are computed for the 32 model motion templates and the image optical flow aligning the feet of the model and the human hypothesis in the image.

### 7.4 Inferences in the Model

To estimate the modes and phases, we need to buffer the observations from frame 1 to frame \( T \) and to compute inferences in the motion model to get an optimal path which maximize the likelihood of the observations—both body speed and the motion template responses.

Denote \( \lambda \) as the motion model, \( V = \{v_1,...,v_T\} \) as the speed, \( O = \{o_1,...,o_T\} \) as the motion template observations, and \( Q = \{q_1,...,q_T\} \) as the states in each frame. \( q_t \) is further decomposed into \([m_t, p_t]\) in which \( m_t \) is the mode and \( p_t \) (\( p_t \in \{1,...,16\} \) for walking/running and \( p_t = 1 \) for standing) is the quantized phase in that state. We also number \( q_t \) from 0 to 32 where states 0,...,15 are walking phases \( W_1,...,W_{15} \), states 16,...,31 are running phases \( R_1,...,R_{15} \) and state 32 is the standing state. \( A_{i,j} = P(q_t = j | q_{t-1} = i) \)
Figure 7.3: Computing model motion template. (a) The diagram to compute motion template from model; (b) How the motion of each pixel is determined; (c) Result motion templates for walking at certain condition.
is the state transition probability and we use $A_{0,i}$ to denote the initial state distribution for simplicity of notation. The normalized distance of motion template $[m_t,p_t]$ with the image optical flow is denoted as $D_t^{[m_t,p_t]}$. The optimal path is given by $Q^* = \arg\max P(O,V|Q,\lambda)$.

If we assume that the speed and the motion template responses are conditionally independent given the state $q_t$, we have

$$P(O,V|Q,\lambda) = \prod_{t=1}^{T} A_{q_{t-1},q_t}P(o_t,v_t|q_t,\lambda)$$
$$= \prod_{t=1}^{T} A_{q_{t-1},q_t}P(o_t|q_t,\lambda)P(v_t|q_t,\lambda)$$
$$= \prod_{t=1}^{T} A_{q_{t-1},q_t}P(o_t|[m_t,p_t],\lambda)P(v_t|m_t,\lambda) \quad (7.4)$$

By assuming a Gaussian noise model $\mathcal{N}(0,\sigma)$ where $\sigma$ is the standard deviation of the Gaussian distribution; it is identical for all states) for the model motion templates, the likelihood of walking/running motion templates is given by

$$P(o_t|[m_t,p_t],\lambda) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{D_t^{[m_t,p_t]}^2}{2\sigma^2} \right\} \quad (7.5)$$

$P(o_t|m_t = \text{stand},\lambda)$ is fixed to 1. Computing the optimal path is similar to the decoding problem in an hidden Markov model (HMM). We employ the Viterbi algorithm [89] based on dynamical programming for inference. A quantity $\delta_t(i)$ is defined as

$$\delta_t(i) = \max_{q_1,...,q_{t-1}} P(q_1,...,q_{t-1},q_t = i, o_1,...,o_t, v_1,...,v_t|\lambda), \quad (7.6)$$

which is the maximum probability of states $q_1,...,q_{t-1}$ with $q_t$ fixed to $i$. It can be computed from the first frame to the last frame by the induction rule
$$\delta_t(j) = (\max_{1 \leq i \leq N} \delta_{t-1}(i)A_{i,j})P(\alpha_t | q_t = j)P(\nu_t | q_t = j), \quad j = 1, ..., N. \quad (7.7)$$

Another quantity $\psi_t(j)$ is defined as the previous state of a best path passing through state $j$, which is $q_{t-1}$ in computing $\delta_t(i)$ in Eqn 7.6. Its induction rule is

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} (\delta_{t-1}(i)A_{i,j}), \quad j = 1, ..., N. \quad (7.8)$$

After $q^*_T$ is computed, the best path $Q^*$ is backtracked from frame $T - 1$ to the first frame by

$$q^*_t = \psi_{t+1}(q^*_t), \quad t = T - 1, ..., 1. \quad (7.9)$$

The resulting $Q^*$ gives both the high level modes and the detailed phases which corresponds to the body posture of each frame.

### 7.4.1 Optional post-processing

In many cases, the phase of walking or running changes approximately linearly with time. This linear relationship can be enforced in a post-processing stage. This can be used to deal with temporary missing/bad measurements due to occlusion, noise or other reasons. For each segment of walking or running, the linear relationship is written as

$$p_t = (p_t')_{MOD\,16} \text{ where } p'_t = k \times t + b.$$  

Direct estimation of parameters $k$ and $b$ is not straightforward due to the non-linear MOD operation. The structure of the circular state machine restricts that the state can only move forward and can move at most one or two steps at a time. This property enables us to compute the $p'_t$ before the $MOD$ operation as $p'_t = p_t + n \times 16$. $n$ is initialized to zero and is increased by one whenever $p_t$ is less than $p_{t-1}$ (e.g., $p_{t-1} = W_{16}$ and $p_t = W_1$).
Then $\hat{k}$ and $\hat{b}$ are estimated from $\{p'_t, t\}$ pairs using LMS fitting. $\hat{p}_t = (\hat{k} * t + \hat{b})_{MOD16}$ is recomputed and the pose is interpolated by the poses corresponding to $[\hat{p}_1]$ and $[\hat{p}_7]$.

7.5 Implementation and Results

7.5.1 Results on locomotion

We have tested the proposed method on a variety of datasets and found it to work very robustly despite of the difficulties of small image size, temporary occlusion, etc. In our current implementation, all the frames are used for inference. In a real-time system, we can buffer a fixed number of frames and use a sliding window.

Fig.7.4.(a)-(c) shows the result of a 477-frame sequence (BookStore) containing a human walking (frame 38-168, 291-476), running (frame 0-37) and standing (frame 168-290). The human has significant changes of orientation and size, and the trajectory is not linear. Fig.7.4.(a) shows the estimated state of each frame in the optimal path and Fig.7.4.(b) shows the states after the optional post-processing (note that the overlaid models are before the post-processing step). The 3D human pose corresponding to the state is projected and overlaid on the original frames in Fig.7.4.(c). It shows that the estimated motion parameters are accurate and the transitions between modes appear natural. The only inaccuracy occurs around frame 140 when the human’s direction is almost aligned with the camera optical axis and the walking speed is low and the motion of the legs is not very salient. This inaccuracy is resolved in the post-processing step (Fig.7.4.(b)). One deficiency of our method is that the human orientation when standing cannot be inferred from its speed which is close to zero. Another example (Leavy4b) is
shown in Fig.7.4.(d) with two humans passing by as well as other two humans. The result is not affected by the occlusion and the noisy blob due to the low contrast of one’s pants to the background (Fig.4.1.(c)).

The program is implemented in un-optimized C++ code and runs at about 8 Hz with 360*240 frame size on a Pentium 4 2.7G Hz PC. The main computation effort is spent on computing image optical flow and generating model motion templates. The time for inference part (i.e., the Viterbi algorithm) is negligible. As a trade-off for computation, simpler but not as powerful observation model (e.g., shape, edges, etc) could also be used.

### 7.5.2 Human verification by walking

In Sec.4.2, we form human hypotheses by shape analysis of the foreground blobs. These hypotheses need to be verified since non-human foreground pixels exist and it is possible that they are also hypothesized as humans (e.g., Fig.4.4.(g)). Human appearance varies significantly due to viewpoint, clothing and non-rigid motion. In some situations, even human observers have difficulty in telling the presence of humans from only a static image. Dynamical features can provide much more robust information. We observe that the motion of legs for walking people is a very salient feature, even for people projected to small sizes in the image. We use walking as the feature to verify the human hypotheses.

Previous researches used motion periodicity to recognize human motion (an overview is given in [21]). However, it should be noted that the motion of human shadow and reflection is also periodic. In [102], human motion is detected by mapping the motion of some feature points to a learnt probabilistic model of joint position and velocity of different body features in a small number of frames, however, joints are required to be
Figure 7.4: Result of estimating locomotion modes and phases. (a) (b) (c) Result on sequence BookStore: state output (0-15 walking; 16-31-running; 32-standing) before (a) and after (b) optional post-processing; (c) frame 15, 52, 116, 278, 358, 453 of the estimated stick-figure model overlaid on images, color showing mode (red-walking; blue-running; green-standing). (d) Selected frames of result on sequence Leavy4b, color showing different object identity.
detected as features and the experiments were conducted only on environments with limited clutters.

Human verification is only tested on a walking model. This model is only the part involved in walking (e.g., \( W_1 \sim W_{16} \)) in Fig.7.1. We set the verification process to execute for \( T \) frames \( (T = 40; \sim 1.3 \text{s}) \), which is the average length of a walking cycle. The verification is done once \( T \) frames of observation are obtained and the optimal path is obtained by the state machine inference as well as the optional post-processing. The hypotheses corresponding to humans exhibit valid walking speed (in phase space) and the summed correlation value is high. The example in Fig.7.5 shows the verification results of two hypotheses - a human and its reflection. The stick figure models corresponding to the phases are drawn according to the computed phases in Fig.7.5. As can be seen, for the valid hypothesis, the computed phase matches with the image very well, while the stick-figure does not move for the invalid one.

The walking pattern is a salient feature in most viewpoints except frontal/back viewpoints in which only a small amount of motion is visible in image. In these cases, since the human exhibits little motion in the image, the shape of human body can be used to verify the hypotheses. If a human’s legs are highly occluded during the verification period, the verification cannot be performed.

Besides the example shown in Fig.7.5.(a) (VerifyR), a wide variety of verification results were collected and some are shown in Fig.7.5. In our experiments, we set the verification process to start 4 frames after the hypothesis is formed to get a better estimate of its orientation.
Fig. 7.5 (b) (VerifyA) shows an example of two close-by walkers. The human height is about 25 pixels in the image and there are frames in which the walkers are occluded by the map board. (c) (VerifyB) shows an example of a woman in long dress. In all the sequences, the phase alignment is very accurate. The results are very promising since the small image size is difficult even for human observers and our method was not confused by left/right leg ambiguity.

We have tested the verification on several sequences which contain 45 distinct people. Before the walking verification, there are 12 false alarms and 1 missed detection. The verification cuts down the number of false alarms to 2, however, it also rejects an extra 5 real humans. This happens mostly when the walkers are walking towards or away from the view direction so that the motion of the leg is not salient.

7.5.3 Remarks

In this chapter, we have presented a tracking as recognition approach for human locomotion tracking and obtained robust results. It is a general formulation which we believe is applicable for other types of motions as well. The motion that this approach is likely to be applicable on is those having well-defined syntax (structured motion as in Zhao et al. [120]), in contrast to the stochastic texture like motion [97]. The structured motion can be decomposed into a few primitives (e.g., walking, running) and each primitive can be described with motion trajectories. Some apparent examples include dances, gestures, facial gestures.
Figure 7.5: Selected frames of human verification examples. (a) Real human was verified while his reflection was not; (b)(c) Other verified examples. For each subfigure, 1st column: original image; 2nd column: stick figure model overlaid on the image; 3rd column: stick figure model only.

The motion estimated from a state-based representation is inherently coarse. The result can be refined for purposes such as motion capture or gait analysis. Motion parameters and body parameters can be optimized locally to best fit the images. Having the initial states recovered from a model of global structure, the optimization is more likely to converge to the global optimum.
Chapter 8

Conclusion

In this thesis, we presented approaches to segment and track multiple humans and estimate their gross body postures with a stationary video camera and show satisfactory results, both visually and quantitatively, on very challenging situations.

Different from most previous work, we take a top-down approach and try to interpret the image observations with human (shape, motion) models. Towards the goal of segmenting and tracking the global motion of multiple humans, we presented two approaches. The first approach uses a simpler model and does simpler analysis which results in a real-time system which is capable of tracking in the presence of limited persistent occlusion (e.g. a few people moving together), temporary severe occlusion, shadow and reflections. The second approach follows a Bayesian formulation and solution is sought by computing the maximum of the posterior probability in a joint multi-object space. The computation is generally intractable in such a complex solution space and we employ a Markov chain Monte Carlo (MCMC)-based method. We design reversible Markov chain to explore the solution space in which various heterogeneous bottom-up techniques are incorporated to make the top-down search more efficient than traditional MCMC. This Bayesian-MCMC
approach is more general and can successfully handle the data where in the presence of large-scale persistent occlusions (e.g. a large group of people moving together). Towards the goal of estimating human body postures, we proposed a tracking as recognition approach to do the estimation by making inference in a prior motion model. This results in very robust result at low image resolution and with temporary occlusions. The framework and the approaches are general and extendable to a wide range of other problems.

8.1 Summary of Contributions

- The use of object shape model. A 3D shape model in conjunction with a camera model makes compact constraints the object shapes in the image.

- A Bayesian formulation of multi-object segmentation and tracking problem: the design of likelihood models for simultaneously segmentation and tracking.

- An efficient MCMC-based approach for compute optimal solution: the design of reversible dynamics to explore the solution space of the multi-object segmentation and tracking problem; the use of heterogeneous bottom-up techniques to accelerate the convergence of Markov chain.

- The extension of mean shift tracking to incorporate background information.

- A tracking as recognition framework to robustly estimate 3D articulated body postures.
8.2 Future Directions

There are a few directions which are interesting to explore starting from the thesis work.

Combining the simple and the MCMC approaches The MCMC-based approach gives very good results in complex situations, however, the computation involved is expensive. Given limited computational power, it is interesting to explore how the MCMC-based approach can be combined with the simple approach described in Chap.4.

Towards trajectory-based analysis Tracking is based on the analysis of the estimation of the previous frame (encoding some history information) and the current frame, which is a very local view and therefore inevitably contains ambiguities. There are also ambiguities that can only be resolved by future observations (e.g., whether a completely occluded object disappeared or is hiding behind other objects.) Trajectory-based analysis can look at the problem in a more global view, which may give more meaningful result. However the full trajectory-based analysis may have a solution space too large to handle, again some efficient bottom-up approaches should help. Tracking as an online system, it is ideal that the system delay is minimal. It has to be determined using how much history and future is optimal as a compromise of performance and system delay. The trajectory-based analysis will also unify the tracking as recognition approach in Chap.7.

Tracking indoor activities Currently our experimental domain is restricted to human locomotion where people are standing, walking or running. They usually cover most cases in outdoor human activities. However, other human motion such as sitting
down, reach for objects should also be considered especially in an indoor environment. The shape variations should be modeled. The context information (e.g. a chair) should assist in tracking. Modeling occlusion (by stationary scene object) process as well as occluding objects would be needed since frequent occlusions (e.g. by desks) happen. All these could be fit in different components of the framework.

**Other types of objects** Although proposed and tested for human segmentation and tracking, the framework is general and can be adapted for other objects (e.g. cars) as well. Sometimes different types of moving objects appear at the same time (e.g. people and cars in the street). This requires simultaneous segmentation, tracking and recognition. The mechanism for object switching enables objects of different types to compete for the best interpretation. As computational issues, Markov chain dynamics of switching (possibly including efficient splitting and merging) objects of different types should be added and data-driven methods for new object types should be designed for computational efficiency.

**Use of multiple cameras** All the approaches in this thesis are based on input from a single camera, however they are compatible with inputs from multiple cameras. The likelihood model of a single viewpoint can be easily extended to multiple viewpoints since we use 3D models. In case of a single camera, the perspective effect caused the objects far away from the camera to be less visible and have more occlusion, which creates a higher error rate. Multiple cameras can provide complementary information to each other and therefore the performance is expected to improve.
Integration with event recognition/behavior analysis modules The results of segmentation and tracking are stored in XML form and used by an event recognition program (i.e. [44]) to provide high-level descriptions of the video data. It is interesting that the two parts form a fully integrated system.
Reference List


[82] Data set provided by IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS2001), Kauai, Hawaii, (in conjunction with CVPR01), 2001,


Appendix A

Camera calibration from vanishing points/line

Lv et al. [67] have developed direct algorithms to compute the intrinsic camera parameters (focal length and principle point) and camera rotations (pan, tilt, yaw) from the vanishing points of 3 orthogonal directions. Approximating solutions can also be computed if the vanishing line of the planes parallel to the ground plane (i.e., horizon line) and the vertical vanishing point can be recognized. Camera height can be recovered with a reference object in the image with known height.

The vanishing points and vanishing line can be computed easily if the scene is abundant of man-made structures (e.g., buildings, roads, tiled floor, poles, etc); an example is shown in Fig.A.1.(a). In case the information from the static structures is not sufficient to compute the needed vanishing points and line (e.g. an empty square), we have developed method to compute the vertical vanishing point and the horizon line with humans walking in more than one direction. Fig.A.1.(b) shows graphically the computation from a human at 3 positions which are not on a straight line. Human walking is periodic; we recognize automatically the “leg-crossing” phases from video sequences (Fig.A.1.(c)). Humans at
Figure A.1: Camera calibration using vanishing points/line. (a) Computing vanishing points/line from static structures: in blue-lines to compute the vertical vanishing point, in yellow-the horizon line recognized directly from the image, in green-the reference object; (b) Computing the vertical vanishing point and the horizon line from humans at 3 non-collinear positions; (c) The “leg-crossing” phases recognized automatically from video sequences; (d) The vertical vanishing point and the horizon line computed from the positions in (c). Images courtesy of Lv et al. [67].

these positions are used to compute the vertical vanishing point and the horizon line (Fig.A.1.(d)). More details can be found in [67].
Appendix B

A blob tracker

We developed a blob tracker to compare with the model-based trackers that we proposed. It can find matches between two frames and also recognize splits and merges involving two blobs (i.e., one blob splits into two and two blobs merge). This blob tracker is by no means optimal, but it works reasonably well (as a “blob tracker”) in the experiments we performed.

In each frame, we match the blobs \( \mathbf{B}^{(t)} = \{ B_1^{(t)}, ..., B_n^{(t)} \} \) with the blobs of the previous frame \( \mathbf{B}^{(t-1)} = \{ B_1^{(t-1)}, ..., B_m^{(t-1)} \} \). First we found those \( B_i^{(t)} \) which has a perfect-match in \( \mathbf{B}^{(t-1)} \). Two blobs are declared as perfect-match if their centroids are sufficiently close (lower than a threshold) and their size difference is sufficiently small (smaller than a threshold). Once a perfect-match is found, the pair of blobs are removed from \( \mathbf{B}^{(t-1)} \) and \( \mathbf{B}^{(t)} \). Then we check whether a blob \( B_i^{(t-1)} \) in \( \mathbf{B}^{(t-1)} \) has split into two blobs in \( \mathbf{B}^{(t)} \). To do so, we find all pairs in \( \mathbf{B}^{(t)} \) which lies in a neighborhood of \( B_i^{(t-1)} \). We compute the best match of \( B_i^{(t-1)} \) with the pairs by their combined size and combined centroid. If \( B_i^{(t-1)} \) and its best-matched pair can form a perfect-match, we declare that a split is recognized. The one in the pair larger in size will inherit the ID of \( B_i^{(t-1)} \).
Similarly we recognize *merges* which is fully symmetrical to splits. After all these, the remaining the blobs in $B^{(l-1)}$ are declared as *dead* and the remaining blobs in $B^{(l)}$ are declared as *newly-created*. The dead blobs are saved for a number of frames to see if it can be matched with any newly-created blobs.
Appendix C

Image projection of multi-ellipsoid human model

A multi-ellipsoid human model is a composition of $k$ ellipsoids, in a local coordinate system (centered at the feet, vertical $z$, human facing $x$). Each ellipsoid is parameterized by the length of its 3 axis $l = (l_x, l_y, l_z)$ (aligned with the $x, y, z$ axes respectively), a translation $t = (t_x, t_y, t_z)$ and a rotation $\alpha = (\alpha_x, \alpha_y, \alpha_z)$ in the local coordinate system.

The multi-ellipsoid model is parameterized by feet position in 3D $(x, y)$, orientation on the ground plane $r$, 3D height $h$, relative fatness $f$ and 2D inclination $i$. When these values are fixed, we would like to compute the image projection of the model. We only show the transformation on one of the ellipsoid, and all the other follow the same set of formula.

$Q_0$ is the ellipsoid in its own coordinate system.

$$Q_0 = \begin{bmatrix} \frac{1}{l_x^2} & 0 & 0 & 0 \\ 0 & \frac{1}{l_y^2} & 0 & 0 \\ 0 & 0 & \frac{1}{l_z^2} & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$
$Q_1$ is the ellipsoid in the local coordinate system of feet

$$Q_1 = T_{r_1}^{-T} T_r^{-T} T_{r_1}^{-1} Q_0 T_{r_1}^{-1} T_r^{-1}, \quad (C.1)$$

where $T_{r_1}$ is the matrix of rotating by $\alpha$, $T_r$ is the matrix of scaling by $h$ and $f$ and $T_{t_1}$ is the matrix of translating by $t$. $Q_2$ is the ellipsoid in the world coordinate system

$$Q_2 = T_{r_2}^{-T} T_{r_2}^{-T} Q_1 T_{r_2}^{-1} T_{r_2}^{-1}, \quad (C.2)$$

where $T_{r_2}$ is the matrix of rotating by $r$ and $T_{t_1}$ is the matrix of translating by $(x,y)$. $C_1$ is the projected ellipsoid (an ellipse) in the image coordinate system.

$$C_1 = (P Q_2^{-1} P^T)^{-1}, \quad (C.3)$$

where $P$ is the camera projection matrix. Finally $C$ is the ellipse after the inclination.

$$C = T_{r_3}^{-T} C_1 T_{r_3}^{-1}, \quad (C.4)$$

where $T_{r_3}$ is the 2D rotation matrix by $i$. 
Appendix D

Object detection/tracking using mean shift

Mean shift is a technique for mode seeking in a non-parametric distribution [14] [18]. It can be used to track an object represented with a color histogram [19], which is summarized in Appendix D.1. We extend the original mean shift tracker with a background exclusion principle (Appendix D.2) and combine the two (Appendix D.3) for object detection and tracking with a known background.

D.1 Object attraction using mean shift

This section summarizes the original mean shift color tracker [19]. It tracks an object by minimizing the color histograms difference of the target and an initialized model. The color histogram difference can be written as a non-parametric density form, and the mode-seeking can be performed efficiently using the mean-shift technique [18]. We call it object attraction in contrast to background exclusion introduced in the next subsection.

The target is represented by an elliptical region. Let \( \{x_i^t\}_{i=1,...,n} \) be the normalized 2-pixel locations in the region. An isotropic kernel, with a convex and monotonic

\footnote{Other features can also be used.}
decreasing kernel profile $k$ is used to assign smaller weights to the pixels farther away from the center, considering those closer to the boundary may contain more noise. An $m$-bin color histogram $q = \{q_u\}_{u=1...m}, \Sigma_{u=1}^m q_u = 1$, is constructed

$$q_u = C \sum_{i=1}^n k(||x_i||^2)\delta[b_f(x_i) - u]$$

(D.1)

where function $b_f()$ maps a normalized pixel location to the histogram bin of the color of that pixel location, and $\delta$ is the Kronecker delta function. Similarly, let $\{x_i\}_{i=1,...,n_h}$ be the normalized pixel locations of the target candidate centered at $y$. Its $m$-bin color histogram $p(y) = \{p_u(y)\}_{u=1...m}, \Sigma_{u=1}^m p_u(y) = 1$

$$p_u(y) = C_h \sum_{i=1}^{n_h} k(||y-x_i||^2/h)\delta[b_f(x_i) - u]$$

(D.2)

where the bandwidth $h$ reflects the possible change of the object size.

The Bhattacharyya coefficient of the two histograms

$$\rho(p(y), q) = \sum_{u=1}^m \sqrt{p_u(y)q_u}. \quad \text{(D.3)}$$

reflects how similar the two histograms are. Tracking the object is done by maximizing $L_1(y) = \rho(p(y), q)$. 

\(^2\text{The target region is normalized into a unit circle.}\)
Apply Taylor expansion around the value $p(y_0)^3$ and keep only the linear term,

$$L_1(y) = \rho(p(y), q)$$

$$\approx \frac{1}{2} \sum_{u=1}^{m} \sqrt{p_u(y_0) q_u} + \frac{1}{2} p_u(y) \sqrt{q_u \over p_u(y_0)}$$

$$= \frac{1}{2} \sum_{u=1}^{m} \sqrt{p_u(y_0) q_u} + \frac{C}{2} \sum_{i=1}^{n_h} w_i \cdot k(||y - x_i||^2),$$

where

$$w_i = \sum_{u=1}^{m} \sqrt{q_u \over p_u(y_0)} \delta[\hat{y}_i(x_i) - u]. \quad (D.4)$$

The second term of $L_1(y)$ is the density estimate computed with kernel profile $k$ at $y$, with weights that can be computed. The mean-shift algorithm applies and $L_1(y)$ will be increased with the new location moved to

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i \cdot g(||y_0 - x_i||^2)}{\sum_{i=1}^{n_h} w_i \cdot g(||x_0 - x_i||^2)}, \quad (D.5)$$

where $g(x) = -k'(x)$. In particular, by using the Epanechnikov profile

$$k(x) = \begin{cases} \frac{1}{2} x^{-1} (d + 2) (1 - x) & \text{if } x \leq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (D.6)$$

\footnote{Note that we need to treat $p$ instead of $y$ as the variable, because later we will use the gradient of the density for optimization.}
\( g = k' = \text{constant}, \) Equ. D.5 simplifies into a weighted average

\[
y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i}.
\]  

(D.7)

In each frame, Equ. D.5 is iteratively applied until convergence. The advantage of the mean shift over gradient decent method is its adaptive step size. The operational basin of attraction is at least equal to the size of the target model. In other words, if the object has any overlap in the consecutive two frames, it will be correctly tracked.

Binary (0/1 or black/white) representation and observation is a special case of the technique described above. The histograms now have only two bins. If we assume that the object is all white \((q_0 = 0, q_1 = 1)\), the formulas can be simplified as

\[
y_1 = \frac{\sum_{i=1}^{n_h} \delta[b_j(x_i) - 1|x_i]}{\sum_{i=1}^{n_h} \delta[b_j(x) - 1]} = \frac{\sum_{j \in S \cap F} u_j}{\sum_{j \in S \cap F}},
\]  

(D.8)

where \(S \ast F\) is the set of foreground ("white") pixels in the object region \(S\). \(y_1\) is the average of the positions of the "white" pixels.

Please note that the original mean shift tracking is derived for an elliptic shape. In our case, the object shape (a composition of ellipses) may not be an ellipse. In computation, we use an ellipse which can just cover the object shape. \(n_h\) is the number of pixels within the object shape. \(y\) is the centroid of the object shape.
### D.2 Background exclusion using mean shift

The known background model of a stationary camera can help in detection and tracking. An object can be detected where the feature histogram of the image is sufficiently different to that of the background within the object region. If the region of high difference with the background is followed in an image sequence, tracking is achieved.

We want to maximize the difference of the observed object color distribution (in the input image) and the object background distribution (in the background image) inside the corresponding region, which we call background exclusion. Denote the object position as \( \mathbf{y} \), and \( \{\mathbf{x}_i\}_{i=1}^{n_h} \) as the normalized pixel locations of the target candidate. Its \( m \)-bin color histogram \( p(\mathbf{y}) = \{p_u(\mathbf{y})\}_{u=1}^{m}, \sum_{u=1}^{m} p_u(\mathbf{y}) = 1 \)

\[
p_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k(||\frac{\mathbf{y} - \mathbf{x}_i}{h}||^2) \delta[b_f(\mathbf{x}_i) - u]. \tag{D.9}
\]

Its \( m \)-bin background color histogram \( \mathbf{d}(\mathbf{y}) = \{d_u(\mathbf{y})\}_{u=1}^{m}, \sum_{u=1}^{m} d_u(\mathbf{y}) = 1 \)

\[
d_u(\mathbf{y}) = C_1 \sum_{i=1}^{m} k(||\frac{\mathbf{y} - \mathbf{x}_i}{h}||^2) \delta[b_b(\mathbf{x}_i) - u], \tag{D.10}
\]

where \( b_b(\mathbf{x}_i) \) converts pixel location \( \mathbf{x}_i \) into the bin number of the background color at \( \mathbf{x}_i \); \( b_f(\mathbf{x}_i) \) converts the pixel location \( \mathbf{x}_i \) into the bin number of the input image color at \( \mathbf{x}_i \).

We would like to maximize \( L_2(\mathbf{y}) = -\rho(\mathbf{p}(\mathbf{y}), \mathbf{d}(\mathbf{y})) \) where \( \rho(\mathbf{p}(\mathbf{y}), \mathbf{d}(\mathbf{y})) \) is the Bhattacharyya coefficient. Following the derivation of the original mean shift tracker, by Taylor first order expansion at \( \mathbf{p}(\mathbf{y}_0) \) and \( \mathbf{d}(\mathbf{y}_0) \),

171
\[ L_2(y) = -\rho(p(y), d(y)) = -\rho(y) \]

\[ \approx -\rho(y_0) - \rho_p'(y_0)(p(y) - p(y_0)) - \rho_d'(y_0)(d(y) - d(y_0)) \]

\[ = -\rho(y_0) + \rho_p'(y_0)p(y_0) + \rho_d'(y_0)d(y_0) - \rho_p'(y_0)p(y) + \rho_d'(y_0)d(y) \]

\[ = c_2 - \sum_{u=1}^{m} \frac{d_u(y_0)}{p_u(y_0)}p_u(y) - \sum_{u=1}^{m} \frac{p_u(y_0)}{d_u(y_0)}d_u(y) \]

\[ = c_2 - \sum_{u=1}^{m} \frac{d_u(y_0)}{p_u(y_0)} \sum_{i=1}^{n} k(|| \frac{y - x_i}{h} ||^2) \delta[b_f(x_i) - u] \]

\[ - \sum_{u=1}^{m} \frac{p_u(y_0)}{d_u(y_0)} \sum_{i=1}^{n} k(|| \frac{y - x_i}{h} ||^2) \delta[b_h(x_i) - u] \]

\[ = c_2 + \sum_{i=1}^{n} k(|| \frac{y - x_i}{h} ||^2) \sum_{u=1}^{m} \left( -\frac{d_u(y_0)}{p_u(y_0)} \delta[b_f(x_i) - u] - \frac{p_u(y_0)}{d_u(y_0)} \delta[b_h(x_i) - u] \right) \]

This is a form of kernel “density” estimation with negative weights \((w_i^b \leq 0)\). The original mean-shift algorithm is modified slightly to accommodate the negative weights [17]. Assuming an Epanechnikov kernel, moving the object to a new position \(y_2\) will increase \(L_2(y)\), where

\[ y_2 = \frac{\sum_{i=1}^{m} x_i w_i^b}{\sum_{i=1}^{m} |w_i^b|}. \quad (D.11) \]

Tracking an object by background exclusion can prevent the object from drifting on the background when the appearance of the object has rapid change (e.g. due to sudden illumination change). We show an example of the background exclusion tracker in Fig.D.1. The background exclusion tracker successfully tracks an object from under the sun into a shadow area while the original mean shift tracker fails. But please note that background exclusion alone may not be sufficient to track an object with other moving object around
Figure D.1: Comparing the original mean shift tracker (a) and the background exclusion tracker (b) on an example.

(e.g., in a crowd) because the attraction basins produced by multiple objects are connected and the object may lock on an overlapping object whose color profile has the most difference from the background.

D.3 Combining object attraction and background exclusion

The object attraction and the background exclusion principles can be easily combined to provide more robust tracking performance. This time we would like to maximize

\[ L_3(y) = \lambda_f \rho(p(y), q) - \lambda_b \rho(p(y), d(y)), \]

where \( \lambda_f \) and \( \lambda_b \) are two parameters to control the relative importance of the two. By the linearity of the derivation,

\[ L_3(y) \approx c_3 + \sum_{i=1}^{n} k(\frac{y - x_i}{h}) (\lambda_f w_i^f + \lambda_b w_i^b) \]  \hspace{1cm} (D.12)
Figure D.2: Comparing the original mean shift tracker, the background exclusion tracker and the combined tracker on an example. (a) the original mean shift tracker is distracted by a pattern on the ground which has similar color profile with the object; (b) the background exclusion tracker is immediately distracted to the dark object which has more difference to the background; (c) the combined tracker successfully tracks. Note the images in each line may not correspond to the same frame.

Plugging $w_i$ in Eqn.D.11 will lead to a new position that increases $L_3(y)$. We show a simple example in Fig.D.2 where the original mean shift tracker fails due to the similarity of the object color profile with the background, the background exclusion tracker fails due to the reason described in Appendix D.2, and the combined tracker successfully tracks.
Publication notes

The thesis work started from the end of the year 2000. The initial version of the model-based segmentation and tracking described in Chap.4 was published in CVPR 2001 [119] which has been undergoing various modifications (including the complete abandon of blob tracking) ever since and Chap.4 describes the most recent and stable version. The tracking as recognition approach was firstly used in verifying human hypotheses by walking, also in [119], however a computational inefficient search was used for temporal integration. The idea was generalized to more complex motion (locomotion) and the Viterbi algorithm was used for temporal integration, published in ICPR 2002 [121]. The Bayesian formulation of the multiple human segmentation problem and its MCMC computational approach were firstly presented in WMVC 2002 [122], with very preliminary results. More work is done along the path and a more mature system was published in CVPR 2003 [123] with more convincing results. Both papers only address the foreground/background likelihood. The color-based likelihood and the extension to tracking have not been published by the time of the completion of this thesis.