TRACKING HUMAN MOTION IN MONOCULAR VIDEO SEQUENCE WITH THE DE-MC PARTICLE FILTER

by

MING DU
B. Sc., Beijing Institute of Technology,
China 2002

A thesis
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2005

Author's Declaration

I hereby declare that I am the sole author of this thesis.
I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

(Ming Du)

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

(Ming Du)

Borrower's Page

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Name	Signature	Address	Date

ABSTRACT

Tracking Human Motion in Monocular Video Sequences with the DE-MC Particle Filter

Master of Applied Science
Department of Electrical and Computer Engineering
Ryerson University

Tracking human motion from monocular video sequences has attracted a great deal of interests in recent years. The difficulty in solving this problem is largely due to the nonlinear property of human dynamics and the high dimensionality of the state vector space required to model human motion. Traditional particle filtering methods usually fail in this situation because the distributions they sample from are ill-defined. In this thesis we propose a novel tracking algorithm, namely the Differential Evolution - Markov Chain (DE-MC) particle filtering. It is based on the particle filter framework but makes substantial changes to its core, i.e. the sampling strategy. In this new approach, the Differential Evolution algorithm and the Markov Chain Monte Carlo algorithm are integrated, aiming at improving both the accuracy and efficiency in approximating the posterior distribution. Global optimization and importance sampling are spirits of the proposed method. To apply the DE-MC particle filter to articulated model-based human motion tracking, we also integrate multiple image cues including the area of silhouettes, color histograms and boundaries to measure the image likelihoods. We find the Fourier Descriptor (FD) to be a new and effective image feature in human motion tracking applications. Our other contributions, such as a modified color cue-based measurement function and a simple adaptive strategy for sampling, also help to improve the performance of the human tracker. Experimental results including the comparison with the performance of other particle filtering methods demonstrate the power of the proposed approach.

Acknowledgments

First and foremost I would like to acknowledge my supervisor, Dr. Ling Guan at Ryerson University. I would like thank him for his fantastic guidance and support I have received throughout the course of my graduate study. Without his instructions and encouragement, accomplishment of this work would have been impossible.

I would like to thank Research Chair Program, Canada for providing funding to support my research work. I would also like to thank Canada Foundation for Innovation (CFI) and the Electrical and Computer engineering Department of Ryerson University for providing equipments. My thanks are due to the Graduate School of Ryerson University for providing me the Graduate Student Scholarship.

I would like to thank many professors and graduate students at Ryerson University for their help with my work. I would like to give my thanks especially to my colleagues in the Ryerson Multimedia Research Lab. It is my pleasure to work in such a friendly and collaborative environment.

My thanks are also due to my parents and my sister who consistently give me confidence and support me. Their love is invaluable treasure to me. I would also thank Ms. Xiaolin Zhu and Ms. Miao Liu.

Contents

1 Introduction 1
- 1.1 General Background . . . 1
- 1.2 Human Motion Tracking . . . 2
- 1.3 Application Scenarios . . . 3
- 1.4 Difficulties in Human Tracking . . . 4
- 1.5 Contributions . . . 6
- 1.6 Outline of thesis . . . 6

2 Related Work 8
- 2.1 Non-articulated-model-based Approaches . . . 8
- 2.2 Articulated-model-based Approaches . . . 9
  - 2.2.1 Bottom-up Methods . . . 10
  - 2.2.2 Top-down Methods . . . 10
- 2.3 Summary . . . 19

3 Particle Filter 21
- 3.1 Problem Formulation for Visual Tracking . . . 21
- 3.2 The Monte Carlo Simulation . . . 23
- 3.3 Importance Sampling . . . 24
- 3.4 Sequential Monte Carlo Sampling . . . 25
- 3.5 Some Implementation Issues . . . 28

4 From 3D World to 2D Images 30
- 4.1 Rigid Geometric Transformations . . . 30
- 4.2 Homogeneous Coordinates and the Change of Coordinates . . . 32
- 4.3 Perspective Projection . . . 33
- 4.4 Camera Calibration . . . 35
- 4.5 3D Articulated Human Body Model . . . 37
- 4.6 Summary . . . 40

5 Fusing Multi-Cue for Tracking 42
- 5.1 Data Acquisition and Silhouette Extraction . . . 42
- 5.2 Measurement Function . . . 44

5.2.1 Area of Silhouette 44
5.2.2 Color Histogram 44
5.2.3 Boundary 46
5.2.4 Combination 49

6 The DE-MC Particle Filter **51**
6.1 Markov Chain Monte Carlo 52
6.2 The Differential Evolution Algorithm 55
6.3 The Differential Evolution Markov Chain 57
6.4 The DE-MC particle filter 59

7 Experimental Results **62**
7.1 General Experiment Results 62
7.2 Comparison Experiment Results 71

8 Conclusions **77**
8.1 Summary 77
8.2 Open Issues 78

Bibliography **81**

A List of Publications **89**

List of Figures

1.1 Man ascending stairs, photograph from Eadweard Muybridge's 'Animal Locomotion'(1887) 2
1.2 Illustration of self-occlusion, depth ambiguity, motion blur, and loose fittings. 4

3.1 Illustration of particle filtering process 28

4.1 Rotations about three coordinate axes and arbitrary axis. 32
4.2 The perspective projection 34
4.3 Derivation of perspective transformation 34
4.4 Camera calibration 36
4.5 Extrinsic parameter estimate result 38
4.6 The proposed 3D articulated human body model and its hierarchical structure 38
4.7 Connected body segments and their local coordinate systems 39
4.8 The image formation process 41

5.1 Silhouette extraction result. 43
5.2 Boundary extraction result (left) and the boundary reconstructed from the first 100 (middle) and the first 50 (right) FD coefficients. 48
5.3 Euclidean distance between Fourier Descriptors extracted from ground-truth image and that extracted from hypothesis image. 49
5.4 The measurement function surface for a human motion video frame w.r.t. 2 DOFs when other DOFs are ground-truth data. 50

6.1 The Metropolis-Hasting algorithm implementation for MCMC 54
6.2 The Differential Evolution Algorithm 56

7.1 General experiment: tracking result for Sequence 1(I). 64
7.2 General experiment: tracking result for Sequence 1(II). 65
7.3 General experiment: tracking result for Sequence 1(III). 66
7.4 General experiment: tracking result for Sequence 1(IV). 67
7.5 General experiment: tracking result for Sequence 2(I). 68
7.6 General experiment: tracking result for Sequence 2(II). 69
7.7 Estimated joint angles for the two sequences 70
7.8 Reference joint angles and directions 70

7.9 Comparison Experiment 1: comparison of the performance of the DE-MC particle filter with different tracking algorithms. 72
7.10 Comparison Experiment 2: comparison of the performance of the DE-MC particle filters with different layer number. 73
7.11 Comparison Experiment 3: comparison of the performance of different measurement function 75
7.12 Comparison Experiment 4: result before (top) and after (bottom) adopting relative weights for color-cue based measurement function 76

List of Tables

2.1 Features of 3D articulated human body model used in the previous research. 11
2.2 Image features used in previous human tracking research 14

4.1 Range of joint angles 40

7.1 Computational cost for different algorithm used in Comparison Experiment 1 71

Chapter 1

Introduction

1.1 General Background

One of the most important characteristics of human resides in their extended ability to communicate. After computer was invented, communication between human and computer has become a critical research issue. For decades many technological breakthroughs have been made to create increasingly powerful computers, but the ability of computers to understand human behavior is still limited. In other words, the effective bandwidth of information flow going from computers to humans was increased by the multimedia platform; however, comparably less advancement has been made the other way. Usually, computers receive signals from humans through low bandwidth devices such as a keyboard or mouse. These devices have turned out to be the bottleneck in communications between human and computer. This problem becomes even more apparent with the emergence of novel technology such as virtual reality. Because of those problems people started to develop Human Computer Interaction (HCI) technology.

We can list numerous applications for HCI, such as electronic entertainment, information retrieval, security and surveillance, interactive education. In these applications, the computer analyzes the physiological or behavioral information of the human to make an appropriate response. Physiological properties, including face, fingerprint, iris and anthropometric measurement, are often used in verification and identification. On the other hand, behavioral properties, including gesture, speech, emotion, gait and other types of motion,

can function not only as the input to a biometrics system, but also to control the computer in the context of artificial intelligence.

1.2 Human Motion Tracking

Analysis of human motion, or more precisely, analysis of a human's full-body movement, is an important component of HCI. The history of motion analysis from image can be traced back to 1878, when Eadweard Muybridge, an English photographer, used a row of cameras to snap more than a dozen photographs of a passing horse. Although his original intention was to verify that there exists a moment at which a trotting horse would have all its four hooves off the ground simultaneously [2], his work was then extended to capture many human and other animal motions. In his famous books *The Horse in Motion* (1878), *The Human Figure in Motion* (1901) and *Animals in Motion* (1899), thousands of pictures of men, women, children, amputees, and many domestic and wild animals are captured in action. Figure 1.1 is one of the examples. His work sparked the scientific research on animal locomotion, which was published in 1887, and later, inspired the research on human motion analysis.

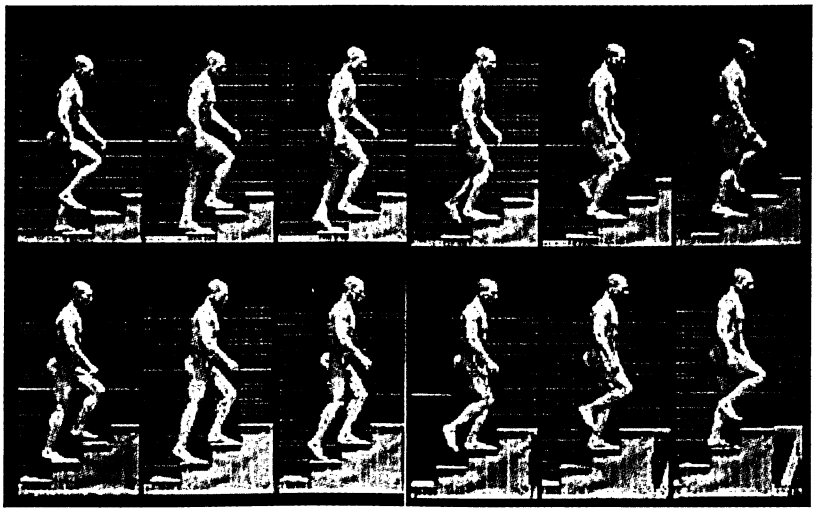


Figure 1.1: Man ascending stairs, photograph from Eadweard Muybridge's 'Animal Locomotion'(1887)

The goal of human motion analysis is to accurately describe the properties of different kinds of actions and correctly recognize them, but we can not expect a computer to understand high-level concepts directly. As a prerequisite for successful analysis, human motion must be quantified so as to be understood by the computer, which presents the problem of motion tracking. Although the human vision system can locate another person and his body parts easily, accurately and quickly, the same task remains arduous for computers. Early work on human motion tracking makes extensive use of sensors [3][4]. Wearable markers report the position of main body parts in a comparably high frequency. Although a precise human motion configuration can be recovered from the received signals, this should not be considered as a general solution in that the markers will cause not only distorted motions but also confined application scenarios in which cooperative subjects must be available. Therefore, tracking human motion from videos, especially monocular videos, has an irreplaceable position in HCI.

1.3 Application Scenarios

There are a bundle of direct potential application scenarios for human motion tracking, the most important of which, are highlighted as follows.

In an electronic reference system for gymnastics or diving competitions, motions of athlete are captured and tracked, and the quantified data is then compared to various criteria for a fair evaluation, thus removing errors to a large extent. Similarly, an automatic sports training system will be able to rectify the users' mistakes after tracking their motion and comparing the results with standard actions stored in the system. In the future, computer games or software are expected to receive users' input via human tracking equipment, responding according to the analysis result of the input. Also, next generation surveillance systems will be able to uncover suspicious actions by tracking all the people within its scope.

1.4 Difficulties in Human Tracking

Although the outlook for future applications of human tracking is very attractive, we still need to face the status quo in this area. Besides the many difficulties which are in common with tracking normal objects, there are some particular obstacles to tracking human movement. We must be confronted with both the complex nature of the human motion and the wide gap between low-level image features and high-level concepts. Those difficulties lie in the following aspects:

Figure 1.2: Illustration of self-occlusion, depth ambiguity, motion blur, and loose fittings.

Depth Ambiguity: Videos are 2-dimensional descriptions of 3-dimensional real world. The depth information cannot be solved from projection equations if no prior knowledge about the object size is available. A direct consequence of depth ambiguity is that similar poses which are only different with each other in the direction of limbs may yield indistinctive observations. This can only be solved by observations from additional view angles. In the leftmost image of Figure 1.2, the subject's right lower arm is almost perpendicular to the projection plane of camera and causes depth ambiguity. If we do not know the size of this lower limb, it will be very hard to predict its actual position.

Loose-body Clothes: The appearance of humans in a video is largely determined by clothing. When loose-body clothes such as skirts are dressed, the shape and contour of human are distorted to a great extent; hence the observable motion may no longer follow the

patterns of normal human motion. The changes of shape and contour across time are also difficult to tackle since any deformable shape analysis needed remains an unsolved problem in computer vision. Moreover, loosely fitting clothes may render one or more body parts totally invisible in the scene. For example, in the rightmost image of Figure 1.2, it is almost impossible to directly locate the upper legs of the subject because they are covered by skirt.

Motion blur: Motion blur usually accompanies fast human motion. When it occurs, the pixels in a blurred image area have almost uniform intensity values. As a result even with the naked eye, it is very difficult to locate the exact position of a blurred object. In practice, tips of moving limbs are more prone to blurring in videos.

Self-occlusion: In monocular video sequences, occlusions are unavoidable. Self-occlusion: part of the body being shaded by other parts of the body, is especially troublesome. In this case, observation will be incomplete during a certain period of time. This is the major cause of a large number of tracking failures. Basically, for monocular video tracking we should not expect to predict the position of the occluded body parts at each frame accurately. Instead, we set the target as being able to recover from temporary tracking failure when the occluded body parts reappear in the scene. The image in the middle of Figure 1.2 is an example with serious motion blur.

Non-linear Dynamic Model: In a tracking problem, when the movements of the object are in accordance with a linear dynamic model, and if the measurement model is also linear, then a Kalman filter can be applied. However, even small non-linearity in the dynamic model will lead to a substantial increase of peak numbers in the posterior [5], in which case both the Kalman filter and its extended version (the Extended Kalman Filter, EKF) lose their efficacy. In the leftmost image of Figure 1.2, most of the subject's left arm is occluded.

Multi-modality of the Measurement Function: The relationship between human motion and image appearance is too complex to be formulated mathematically. To this day, the image processing society has not solved the problem of exactly describing objects in images, as such, we can only select some simple image features to represent measurement information. Normally, the measurement function is not directly related to state variables

and is multimodal with many peaks.

The above difficulties have to be addressed by any human motion tracker design. For many of them, there has not been and will never be perfect solutions. But there indeed exist some partially successful methods.

1.5 Contributions

In this thesis a novel human motion tracker with top-down structure is proposed based on the particle filter. By noticing the fact that the traditional CONDENSATION algorithm ignores the most recent observation and therefore produces unreliable human motion tracking results, we introduce the Differential Evolution Markov Chain (DE-MC) algorithm from statistics and optimization theory to address the problem. The proposed DE-MC particle filter incorporates both the advantage of the Differential Evolution algorithm in global optimization and the ability of the Monte Carlo Markov Chain in reasonably sampling a high-dimensional state space. It evidently boosts the performance of the traditional tracking system in terms of more accurate motion vector prediction. In the implementation of the DE-MC particle filter we also fuse region, color and boundary information to build a robust measurement function. Among them, the boundary information represented by the Fourier Descriptors (FD) sets up a new and effective connection between the estimated model parameters and the image likelihoods. Compared with the previously used boundary or contour cue, FD has many noticeable advantages. Our other contributions include improving color cue utilization and introducing a simple adaptive strategy for particle filter implementation. Based on all of these novelties, our human motion tracking system achieves great improvement over the traditional particle filtering methods.

1.6 Outline of thesis

The remainder of this thesis is organized as follows:

Chapter 2: Related Work Previous work in the field of human motion tracking is reviewed from different aspects. Although we try to solve human tracking problem for monocular video sequences, some multi-camera tracking work will also be covered.

Chapter 3: The Particle Filter The theory of particle filtering is introduced.

Chapter 4: From 3D World to 2D Images A 3D articulated human body model is proposed. Relationships between 3D scene and 2D videos are addressed.

Chapter 5: Fusing Multi-Cue for Tracking In this chapter we design a robust multi-cue based measurement function which describes the resemblance between hypothesis and image observations.

Chapter 6: The DE-MC Particle Filter The background knowledge about the Differential Evolution Monte Carlo (DE-MC) algorithm is introduced. Based on this algorithm and the measurement function presented in Chapter 5, a novel extension of the particle filter is proposed.

Chapter 7: Conclusions Experiments are carried out on several monocular video sequences based on the proposed method. The results are shown and analyzed.

Chapter 2

Related Work

Previous human motion tracking research follows two different paths: an articulated-model-based approach and a non-articulated- model-based approach. The most evident difference between them is that the former tries to fit a pre-defined articulated model to image observations, but the latter treats the human body as a whole during feature extraction and then marks different body parts, or processes the features directly even without labelling. Note that although some non-articulated-model-based work claims that a model is used, these models are not articulated models in a real sense and only function as a connection map.

2.1 Non-articulated-model-based Approaches

Non-articulated-model-based approaches usually extract features such as contours and silhouettes. The Ghost system proposed by Haritaoglu et al. detects the convex and concave hulls on silhouettes [6] . Then it first searches for the head within a region constrained by two lines which intersect at the principle axis of silhouettes with a certain angle. Other convex and concave hull vertices are labelled according to the topology of the human body. Leung et al. extracts human body outlines from human motion video sequences and employs extensive knowledge about human body structure to label body parts with ribbons and circles [10]. Ramoser et al. extract blobs from video and switches between two structures - normal torso structure and long torso structure to label the blobs [55]. We notice that some work attempts to track humans with optical flow methods, such as Shi-Tomashi-Kanade tracker

in [38], 3D depth flow in [51], normal flow in [52] etc. However, they can only handle simple and short-duration motions for a small set of limbs. A more suitable application scenario for them is automatic initialization of model parameters, which we will discuss in this thesis.

Non-articulated-model-based approaches are normally easy to apply and are computationally efficient. Some of them can even achieve nearly real-time performance. However, lack of systematic interpretation for human body structure makes their results very unreliable for semantic processing. All the non-articulated-model-based tracking approaches we discussed above will have a high probability of failure in general application scenarios since silhouettes or outlines are rather a coarse description of human motion. When more reliable and advanced motion analysis is of concern, articulated-model-based approaches are the better choice.

2.2 Articulated-model-based Approaches

Articulated-model-based approaches have the advantage that they can give natural interpretation and description of human body motion. We are able to apply our knowledge about human motion directly to articulated models. The models also reasonably constrain the relative position of body parts. The vast majority of model-based work is in top-down style, namely in a scene the global position of the model is firstly determined, and then locations of other body parts are constrained according to their relative position with regard to the origin of the model coordinate system. On the contrary, in a bottom-up style method, candidates for individual limbs are located separately at first, then those impossible or less likely body part combinations are clipped out and a unique optimal solution is kept. The bottom-up approaches will be reviewed here first, and then the top-down approaches. The bottom-up approaches and the top-down approaches also share some commonalities. When discussing these issues in the top-down section we may mention some bottom-up work and vice versa.

2.2.1 Bottom-up Methods

Ramanan and Forsyth are the first to systemically discuss bottom-up human trackers [24] [53]. They model the 2D view of human body as a puppet of colored, textured rectangles. Parallel lines of contrast are detected as body segment candidates. The appearance template for each body part is learned by clustering candidate feature vectors. The clusters which do not accord with defined constraints are pruned. The factors considered in imposing constraints include human body structure, human kinematics and common sense in dynamics. Similarly, Sigal et al. proposed loose-limbed body models for tracking [36] [39]. This model is composed of tapered cylinders. Templates for head, upper arms and lower legs are learned from a database which contains multi-view images. Eigen-template detectors are then implemented to find these body parts. Spatial and temporal constraints and image likelihoods help to improve the initial hypothesis and locate other body parts. Compared with top-down methods, bottom-up methods make it easier to achieve automatic initialization and tracking failure recovery. The key issues are design of a reliable body part detector and reasonable constraints for clipping out false alarms.

2.2.2 Top-down Methods

When tracking failures happen, top-down methods are often trapped in the proximity of failure configuration in the state space, but once a global position of the human in the image is roughly determined, top-down methods save a lot of energy in searching for positions of individual body part. Important considerations for designing a practical top-down model-based human motion tracker include initialization strategy, human body model type, features for measurement and appropriate search algorithms.

Model Type

Models are used to generate hypothesis observations. Hence, on the one hand they should be designed to provide good fitting for ground-truth observations, on the other hand, computational cost must be taken into account so that complicated models may be avoided.

Authors	Number of body parts	Number of DOFs	Shape of body parts
Rohr [30]	14	8	Cylinders
Yamamoto [13]	12	66	Polyhedrons
Wachter & Nagel [32]	14	24	Right-elliptical cones
Sminchisescu & Triggs [40]	16	38	Superquadric ellipsoids
Moon & Chellapa [7]	15	Not reported	Truncated cones and ellipsoids
Deutscher et al. [15]	17	30	Tapered cones
Roberts et al. [12]	14	22	Super-quadrics
Lee et al. [22]	14	32	Tapered cones
Huang & Chung [35]	10	24	Cylinders
Sidenbladh [21]	12	25	Cylinders and sphere
Green & Guan [34]	11	38	Surface point sets associated to skeleton
Senior [33]	14	Not reported	Ellipsoid and cylinders

Table 2.1: Features of 3D articulated human body model used in the previous research.

Generally speaking, we can roughly categorize the human models proposed in previous work into 2D ones and 3D ones.

Hu et al. model the human as 10 connected rectangles and fit it to silhouettes [37]. This cardboard type model has many similarities with the 3D articulated models which will be discussed later. However, since this is a 2D model, motions in depth can only be reflected by varying the size of rectangles. Therefore the geometric parameters for each rectangle must be re-evaluated at each frame, as with the position parameters. We can see a trade-off here: namely, that a simple model design brings heavy burdens for the following processing. A similar type of cardboard body model is used in [48]. The model is only defined for strict front view and side view and can only cover a portion of possible motion. Lee et al. divide silhouettes of a side-view walking person into 7 regions, modelling each region as ellipse [49]. It is an application of tracking in side-view gait analysis and hence too specific to be generally applied.

3D models are able to handle more complicated motions. In a 3D model body parts

typically take the form of cylinders or truncated cones. At each important joint, one or more degrees of freedom (DOFs) are assigned to allow rotations around axes in different directions. Fine motions such as rotation of fingers and toes are usually not defined in these models. For clarity, we list the features of some previously proposed 3D articulated models in Table 2.1.

Initialization

The geometric parameters of a body model are usually assumed constant throughout the whole tracking process because they represent the anthropometric measurement of body parts. Small fluctuations of the parameters produced by the elasticity of muscles and stretching of joints are usually ignored. It is necessary to acquire the geometric parameters in the stage of initialization. If we know neither the initial state (motion parameters) nor the geometric parameters of the model, the dimension of search space for initialization will be prohibitively high, normally above 60. It provides a real challenge for any known algorithms. Conventionally the human motion trackers are manually initialized. This is a justified simplification which enables us to focus on more important issues of tracking though currently it yields impractical application.

However, there is still some partial success towards automatic initialization. There are basically two kinds of approaches frequently adopted. The first one is template matching, which tries to find structures that are similar to pre-defined body part templates. For instance, Oren et al. detect upright pedestrians with arms hanging at their side [56]; Poggio et al.and Rowley et al. detect people by looking for their faces [57] [58]; Lee et al. detect head and hands with contour templates [22]. Besides, template matching is also an indispensable step for any bottom-up tracking approach, as we have discussed above. This is the reason why bottom-up approaches can achieve automatic initialization more easily. The second method for practical automatic initialization is motion segmentation, which analyzes the first several frames to extract body segments according to motion cues. Gao et al. applied RANSAC algorithm for motion segmentation [8]; Krahnstoever et al. present a framework which is based on low-level motion segmentation and elaborately designed model assembly

iterations [54]. Their results are only shown on sets of two or three body segments and have not been extended to whole body due to complexity.

Besides template matching and motion segmentation, in [29] a promising contour-based approach for automatic initialization is proposed. It fits the contour of a 3D model into that of silhouette. When the two contours intersect each other, a technique called "Maxwell's demons" forces the contour of 3D model to move to the contour of silhouette, otherwise the Iteration Closest Points (ICP) algorithm is applied to make the contours intersect each other. The limitation is that for convergence, the start pose of 3D model must be somehow similar with the ground-truth pose.

Measurement of Fit

In a tracking problem, especially a particle filter based one (about the particle filter theory, please refer to Chapter 3.), an important step is to evaluate the correctness of proposed hypotheses. The criterion is similarity between two sets of image features - one extracted from hypotheses and one extracted from ground-truth. The choice for image features is thus critical for the performance of a human tracker. We are aware of the fact that every image feature alone just describes one aspect of the image observations and provides incomplete information about them. If we can utilize multiple image features, as far as the measurement function is reasonably designed, in most cases, the description of similarity will be more accurate than using one feature alone. However, we should also consider the issue of computation efficiency. In other words, the selected features should be easy to extract. It is not worth improving the accuracy of similarity measurement slightly at the cost of much larger amount of calculation. We summarize the choices for image features from previous work as shown in Table 2.2 .

As we can see from the table, the most popular features are edge, silhouette area and intensity (color). Robustness of these measurements is largely determined by the capture environment and the image processing techniques applied. Edges give good description for the limbs since their projections usually have evident and nearly straight borders. They can be detected even with the presence of cluttered background. Paradoxically, for the

Authors	Image information	Extraction method
Rohr [30]	Edge	
Hu et al. [37]	Area silhouette	Background subtraction
Zhao et al. [41]	Intensity	
Delamarre & Faugeras [29]	Contour	Geodesic active contour
Sminchisescu & Triggs [40]	Edge	Sobel operator
	Motion	Optical flow
	Intensity	
Moon & Chellapa [7]	Boundary	Shape filter
	Intensity	
Deutscher et al. [15]	Edge map	
	Area of silhouette	Background subtraction
Rui & Chen [23]	Edge map	Canny operator
Roberts et al. [12]	Intensity	
Lee et al. [22]	Boundary	
	Area of silhouette	Background subtraction
Huang & Chung [35]	Area of silhouette	Background subtraction
Sidenbladh et al. [21]	Intensity	
Green & Guan [34]	Edge	Gradient map
	Region	
Senior [33]	Area of Silhouette	Background subtraction
Sidenbladh & Black [14]	Edge and ridge	Multi-scale filter
	Motion	Optical flow
Ramanan & Forsyth [24]	Intensity	Color histogram
	Shape	Template matching

Table 2.2: Image features used in previous human tracking research

same reason spurious edges are often extracted due to the disturbance from the texture of clothing. Silhouettes can be easily obtained by background subtraction and subsequent measurement is rather straightforward, but they are helpless to a lot of forms of ambiguities since many observations generated by different hypothesis may yield almost the same area of silhouette. Moreover, non-static backgrounds and shadows will cause segmentation errors for silhouettes. The intensity (color) cues are robust to spatial rotation, non-rigidity and partial occlusion. However, variance of lighting conditions often causes troubles for this feature. Motion cue based on optical flow computation is less frequently adopted now because it leads to a relatively heavy burden of calculation.

Physical constraints are often imposed to reweight the results obtained by the measurement function ([7] [36] [39] [24] [21]), so unrealistic solutions can be avoided. We should notice that this imposition of joint limits should not be confused from a similar implementation in the sampling or prediction stage of particle-filtering-aided human tracking. There, the physical constraints help to directly improve the prior distribution.

For the work in which multiple image cues are combined, the overall measurement function is often defined as the multiplication of individual measurement functions (or the sum of their logarithmic version). It is based on the assumption of statistical independence (naive Bayesian model) of these cues. Roth et al. argue the reasonability of this assumption by showing the strong dependence existing between different measurements [42]. In place of the naive Bayesian model, they develop new image likelihood model based on Gibbs sampling theory. Their experimental results show that the measurement function obtained with the Gibbs model yields a distribution much more approximative to the ground-truth distribution than naive Bayesian model.

Democratic integration [43] fuses different cues together and adaptively varies their weights through evaluating tracking errors caused by each of them. However, since a reliable measurement of tracking error is not available (otherwise there is no need to track at all), most multi-cue fusion methods are still based on constant weight schemes [11] [44]. Even for the determination of fixed weights there has not been a perfect solution. A possible

approach is to learn this knowledge as a prior by testing all candidate features on exemplar motion video sequences. However there is no guarantee that an image feature which is good as a measurement for the training database will be suitable for all videos because of the complexity of motion and the capture environment.

Sampling and Search Strategy

The core of a tracking algorithm is its mechanism of searching for configurations which interpret observations best. For this reason a rich body of technical literature was devoted to designing an efficient sampling and search strategy. Basically the two tasks are towards the same goal and are closely linked: a good sampling strategy will substantially increase the efficiency of searching and a proper search strategy will increase the possibility of finding extrema. Therefore we regard them as an integrated aspect.

Most human motion trackers are based on particle filters (we will give detailed introduction for the particle filter in Chapter 3). Application of particle filters (or Sequential Monte Carlo Sampling) in the computer vision society can be traced back to the CONDENSATION algorithm [17]. Although it does not touch the topic of articulated-model-based human tracking, it does encourage much successive work in this field.

To improve the performance of particle filters, the central issue lies in developing an efficient way to do searching in high dimensional state space. Commonly, physical constraints like joint angle limits [40] are imposed on state vectors to help crop the unreasonable regions from search space, as mentioned in the last subsection, but more advanced ideas are necessary. A natural consideration is to decompose the state space. Some work locates human body parts with evident features in each frame separately by body template matching [22] [45]. It is a more general case of the discussed template matching for bottom-up trackers and automatic initialization: When all the body parts are detected with template matching, it is for the bottom-up tracker; when template matching is only applied to the first several frames it is for automatic initialization. Since the cost of exploring a state space increases exponentially with the space's dimensionality, we can expect to save search cost greatly with this method. However, as we have mentioned previously, the cues used to detect and label

the body parts are not always available or robust, so the performance of the detectors is not reliable enough.

From a different aspect, researchers have considered exploring the relationship of the target function with state space so as to sample more reasonably. The efforts diverge into two branches. Some are based on the original CONDENSATION framework: trying to define a robust dynamical model, i.e. a state transition distribution. An extensively used dynamical model is the general constant velocity model [21]. The prototype of this model seldom yields acceptable tracking results without revisal or help from outside. A refined version which derives the process noise for this model from an uncertainty description matrix is available in [32]. Zhou and Chellappa propose an adaptive velocity model [50] but actually this provides only limited improvement in performance. To particularly track human walking, Moon and Chellappa utilize much explicitly prior knowledge about this motion type [7], including left-right symmetry and periodic patterns, etc. In [31], Sidenbladh et al. explore human motion patterns by learning data from a motion capture database as an extension of their earlier efforts in learning only a walking model. This learning, as opposed to Moon and Moon's approach, is an implicit one. Urtasun and Fua [26] also develop a similar approach. By adopting PCA they were able to reduce the dimension and match the motion history in the target video sequence with motion templates. The closest template will then guide the tracking procedure. Similar in spirit, Hidden Markov Model (HMM) is utilized to segment human motion into fixed states. Zhao and Nevatia [41] defines the locomotion model for standing, running and walking and allows switching between the three modes. There are 16 states for running, 16 states for walking and 1 state for standing. Lan and Huttenlocher [27] also learn a motion template for walking, but since their model is a 2D one, they add the states for a certain number of different views as well. However, the methods of learning motion models or templates impose too strong motion constraints to be used for tracking general human motion, which can never be thoroughly included in a typical motion capture database.

As such, there has been more of a move toward borrowing ideas from sampling theory.

In [16], Isard et.al. proposed ICONDESATION based on importance sampling, which forced particles to be generated from the "important area" of the state space. Those areas which produce samples with low importance are neglected since they "waste" the "energy" of particle filters. The approach helps to improve efficiency of sampling when auxiliary information about the state-space distribution is available as a form of importance function. The unscented particle filter (UPF) [23] is an application of the "importance sampling" concept, albeit much more general than the work in [16]. It draws particles from a proposal distribution which is determined by the calculation result of the unscented Kalman filter (UKF). Sminchisescu and Triggs develop a proposal density based on local parameter estimation uncertainty [40]. Along the lowest few covariance eigen-directions, sampling is implemented with covariances scaled by a factor from 8 to 14. The generated samples are refined by a deterministic Hessian-based optimization algorithm.

A typical problem frequently encountered when doing high-dimensional state space searching is that samples are often trapped in local extrema and fail to escape from them. Deutscher et. al. developed the annealed particle filter (APF) [15], which originates from the simulating annealing algorithm of optimization theory. The algorithm pushes the samples gradually to the global maximum of weighting function by progressively adjusting the sensitivity of the weighting function. This work was then extended to an amended annealed particle filter [19] by making the drift of joint angle configurations adaptive and by introducing crossover operator from another optimization algorithm the Genetic Algorithm (GA). GA is also applied in [37] although there it is not based on particle filtering framework. In a similar research direction, Sminchisescu and Triggs presented hyperdynamics importance sampling which is motivated by computational chemistry theory [18]. The algorithm can guide samples moving to low-cost negative curvature regions which may lead to neighboring cost basins, and hence avoids local minima trapping.

2.3 Summary

In this chapter we gave an overview on previous major work and recent progress in the area of human motion tracking. We focus on the articulated-model-based trackers, especially those aided by the particle filters, because they are able to handle general full-body motions and have promising future for practical applications. They are also the ones which have been relatively thoroughly investigated. Besides particle filters, bottom-up approaches are becoming more and more popular in that they require relatively less complicated algorithm design procedure and that they are easier to be initialized.

For the selection of model, as we see from Table 2.1 a 3D articulated one with 10-17 segments and 22-32 DOFs is usually competent for the task. It can handle motion in arbitrary directions while avoiding a totally intractable high-dimension vector space. We must also manage the balance between the simplicity of the segment shape and achievable geometric fit. The model which is built by attaching a highly accurate 3D mesh surface to articulated skeletons may be widely adopted in the future although it provides a huge challenge for initialization. 2D models can accomplish the tracking task only in a few particular scenarios, and they are losing their advantage with the emergence of more and more powerful computers.

Initialization for articulated 3D human body models is basically an unresolved problem by far. Although partial success has been achieved, a general reliable automatic initialization algorithm is still in absence. We expect new breakthroughs to be made with the help from bottom-up approach and multi-view motion segmentation approaches.

Multi-cue fusion is absolutely necessary for robust tracking in monocular image sequences. Most of the previous work focuses on discussing what image features to track rather than on exploiting how different features affect each other when combined. For further improvement, a method to adaptively adjust relative weights of different cues is necessary. To this end we hope to know how to evaluate the performance of each individual cue on-line in the future.

Most significant research on human tracking takes place in the field of designing an efficient search and sampling strategy. The simplification of the state transition distribu-

tion based on the Markov process assumption has shown its incapability when dealing with complicated human poses. Trade-off has to be made between the ability for handling an abrupt change of motion pattern and the efficiency of searching. Knowledge of statistics and optimization theory has been widely applied and we believe they will guide the future of vision-based human tracking research.

Chapter 3

Particle Filter

Particle filters take a lot of forms across a variety of literature. In the statistics community, the method is know as Sequential Monte Carlo Sampling; In the artificial intelligence community, sometimes it is called survival of the fittest; In the field of computer vision, its name becomes the CONDENSATION algorithm. It provides a robust Bayesian framework for the visual tracking problem and therefore is widely adopted. We will start from formulating the articulated model-based human motion tracking problem, and then introduce the statistics theory related to solve this problem before moving to the application of particle filtering.

3.1 Problem Formulation for Visual Tracking

In an articulated model-based human motion tracking problem, joint angles together with global translation and rotation parameters constitute a state vector. This vector gives a complete description for the pose of human. Therefore, the tracking problem can be formulated as recursively estimating state at each time step according to a posterior distribution $p(\mathbf{X}_{0:k} \mid \mathbf{Y}_{1:k})$ where $\mathbf{X}_{0:k} = \{\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_k\}$ are the state vectors up to and including time k and $\mathbf{Y}_{1:k} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_k\}$ are the observations in the same period of time. In addition, we are particularly interested in $p(\mathbf{X}_k \mid \mathbf{Y}_{1:k})$, which is the so-called filtering process. By Bayesian Inference [17]:

$$p(\mathbf{X}_k \mid \mathbf{Y}_{1:k}) = \lambda_k p(\mathbf{Y}_k \mid \mathbf{X}_k) p(\mathbf{X}_k \mid \mathbf{Y}_{1:k-1})$$

$$
\begin{aligned}
&= \lambda_k p(\mathbf{Y}_k \mid \mathbf{X}_k) \int \frac{p(\mathbf{X}_k, \mathbf{X}_{0:k-1}, \mathbf{Y}_{1:k-1})}{p(\mathbf{Y}_{1:k-1})} d\mathbf{X}_{0:k-1} \\
&= \lambda_k p(\mathbf{Y}_k \mid \mathbf{X}_k) \int \frac{p(\mathbf{X}_k, \mathbf{X}_{0:k-1}, \mathbf{Y}_{1:k-1})}{p(\mathbf{X}_{0:k-1}, \mathbf{Y}_{1:k-1})} \frac{p(\mathbf{X}_{0:k-1}, \mathbf{Y}_{1:k-1})}{p(\mathbf{Y}_{1:k-1})} d\mathbf{X}_{0:k-1} \\
&= \lambda_k p(\mathbf{Y}_k \mid \mathbf{X}_k) \int p(\mathbf{X}_k \mid \mathbf{X}_{0:k-1}, \mathbf{Y}_{1:k-1}) p(\mathbf{X}_{0:k-1} \mid \mathbf{Y}_{1:k-1}) d\mathbf{X}_{0:k-1} \quad (3.1)
\end{aligned}
$$

where λ_k is a normalization constant that is independent of $\mathbf{X}_k$.

We can further simplify Equation 3.1 if the human motion dynamics are assumed to form a temporal Markov chain, then the new state is conditioned directly on the immediate preceding state and independent of the earlier history:

$$
p(\mathbf{X}_k \mid \mathbf{X}_{0:k-1}) = p(\mathbf{X}_k \mid \mathbf{X}_{k-1}) \quad (3.2)
$$

and if observations are assumed to be independent, both mutually and with respect to the dynamical process, then:

$$
\begin{aligned}
p(\mathbf{X}_k, \mathbf{Y}_{1:k-1} \mid \mathbf{X}_{0:k-1}) &= p(\mathbf{X}_k \mid \mathbf{X}_{0:k-1}) \prod_{i=1}^{k-1} p(\mathbf{Y}_i \mid \mathbf{X}_i) \\
&= p(\mathbf{X}_k \mid \mathbf{X}_{0:k-1}) p(\mathbf{Y}_{1:k-1} \mid \mathbf{X}_{0:k-1}) \quad (3.3)
\end{aligned}
$$

Now:

$$
\begin{aligned}
p(\mathbf{X}_k \mid \mathbf{X}_{0:k-1}, \mathbf{Y}_{1:k-1}) &= p(\mathbf{X}_k, \mathbf{Y}_{1:k-1} \mid \mathbf{X}_{0:k-1}) \frac{p(\mathbf{X}_{0:k-1})}{p(\mathbf{X}_{0:k-1}, \mathbf{Y}_{1:k-1})} \\
&= p(\mathbf{X}_k \mid \mathbf{X}_{0:k-1}) \\
&= p(\mathbf{X}_k \mid \mathbf{X}_{k-1}) \quad (3.4)
\end{aligned}
$$

and it follows that Equation 3.1 is reduced to:

$$
\begin{aligned}
p(\mathbf{X}_k \mid \mathbf{Y}_{1:k}) &= \lambda_k p(\mathbf{Y}_k \mid \mathbf{X}_k) \int \int p(\mathbf{X}_k \mid \mathbf{X}_{k-1}) p(\mathbf{X}_{0:k-1} \mid \mathbf{Y}_{1:k-1}) d\mathbf{X}_{0:k-2} d\mathbf{X}_{k-1} \\
&= \lambda_k p(\mathbf{Y}_k \mid \mathbf{X}_k) \int p(\mathbf{X}_k \mid \mathbf{X}_{k-1}) p(\mathbf{X}_{k-1} \mid \mathbf{Y}_{1:k-1}) d\mathbf{X}_{k-1} \quad (3.5)
\end{aligned}
$$

Hence, some specifications are necessary for a recursive solution to human tracking problem:

- $P(\mathbf{X}_0)$: leads to the implementation of initialization.
- $P(\mathbf{X}_k \mid \mathbf{X}_{k-1})$: leads to the definition of a dynamic model.
- $P(\mathbf{Y}_k \mid \mathbf{X}_k)$: leads to the definition of a measurement function or image likelihoods.

3.2 The Monte Carlo Simulation

The Monte Carlo simulation tries to approximate a target distribution $p(\mathbf{X})$ defined on a high-dimensional space χ by drawing a set of independent and identically distributed samples $\{\mathbf{X}^{(i)}\}_{i=1,\ldots,N}$ from it and representing it as an empirical point-mass function:

$$p_N(\mathbf{X}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\mathbf{X}^{(i)}}(\mathbf{X}) \tag{3.6}$$

where $\delta_{\mathbf{X}^{(i)}}(\mathbf{X})$ denotes the delta-Dirac mass located at $\mathbf{X}^{(i)}$. Actually, in many cases, describing a probability distribution is not our primary objective; we wish to obtain a representation of the expectation or some other statistical property for a state. For instance, in the human tracking problem we are interested in computing the expectation of current human pose state. To this end a sampled representation of integrals must be developed.

Now we use the sums $I_N(f)$ to approximate the integrals $I(f)$ as:

$$I_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{X}^{(i)}) \rightarrow I(f) = \int_{\chi} f(\mathbf{X})p(\mathbf{X})d\mathbf{X}. \tag{3.7}$$

According to the strong law of large numbers, this estimate will converge to the target expression when the number of samples N approaches ∞. We can also obtain the maximum of an objective function as follows:

$$\hat{\mathbf{X}} = \arg\max_{\mathbf{X}^{(i)};i=1,\ldots,N} p(\mathbf{X}^{(i)}) \tag{3.8}$$

Actually the expectation and maximum of the sample set are two alternative options for making decisions in the human tracking problem, as we will see later.

The direct Monte Carlo integration method is simple to use and understand, but also has its limitations. When the target distribution is too complex to allow easy generation of independent samples, this method is not applicable anymore. For example, in our tracking problem we wish to get a representation of $p(\mathbf{X}_k \mid \mathbf{Y}_{1:k})$ by sampling, but obviously direct Monto Carlo integration is not possible.

3.3 Importance Sampling

Importance sampling is a Monte Carlo method which is used for approximating distributions of interest by drawing samples from available auxiliary density [60]. It improves the efficiency of sampling by concentrating on regions of state space which contain most information about the posterior distribution. Assume that there is a distribution for $\mathbf{X}$ we wish to sample from. It is with probabilities or probability densities that are proportional to the function $p(\mathbf{X})$. Assume also that the value of $p(\mathbf{X})$ for any $\mathbf{X}$ can be evaluated, but that we are unable to directly sample from it because of its complexity, which is exactly the situation addressed in Section 3.2. As long as we are able to directly sample from another distribution which approximates the one defined by $p(\mathbf{X})$, and this distribution is with the probabilities or probability densities proportional to the function $g(\mathbf{X})$, we will be able to rewrite the integral $I(f)$ as:

$$I(f) = \int f(\mathbf{X})w(\mathbf{X})g(\mathbf{X})dx \tag{3.9}$$

where $w(\mathbf{X}) = \frac{p(\mathbf{X})}{g(\mathbf{X})}$ is called the importance weighting function. It is now clear that if we obtain a set of i.i.d. samples $\{\mathbf{X}^{(i)}\}_{i=1,\ldots,N}$ from the distribution defined by $g(\mathbf{X})$, then according to the Monte Carlo simulation, the sum:

$$\hat{I}_N(f) = \sum_{i=1}^{N} f(\mathbf{X}^{(i)})w(\mathbf{X}^{(i)}) \tag{3.10}$$

will converge to $I(f)$ when $N \rightarrow \infty$ provided $g(\mathbf{X}) \neq 0$ and $p(\mathbf{X}) \neq 0$. Here the correction term $w(\mathbf{X}^{(i)}) = \frac{p(\mathbf{X}^{(i)})}{g(\mathbf{X}^{(i)})}$ compensates for the uneven distribution of sample positions. By examination we can also regard this method as a sampling strategy in which the posterior density function $p(\mathbf{X})$ is approximated by

$$\hat{p}_N(\mathbf{X}) = \sum_{i=1}^{N} \delta_{\mathbf{X}^{(i)}}(\mathbf{X})w(\mathbf{X}^{(i)}) \tag{3.11}$$

It becomes obvious that $\hat{I}_N(f)$ is nothing but the function $f(\mathbf{X})$ integrated w.r.t. the empirical measure $\hat{p}_N(\mathbf{X})$. In the inference of above equations, we make a default assumption that $p(\mathbf{X})$ and $g(\mathbf{X})$ are exactly the distribution densities themselves, i.e. $\int p(\mathbf{X})d\mathbf{X} = 1$

and $\int g(\mathbf{X})d\mathbf{X} = 1$. In a more general case, where $p(\mathbf{X})$ and $g(\mathbf{X})$ are not necessarily the density functions but proportional to them, Equation 3.9, Equation 3.10 and Equation 3.11 should be rewritten as:

$$I(f) = \frac{\int f(\mathbf{X})w(\mathbf{X})g(\mathbf{X})d\mathbf{X}}{\int w(\mathbf{X})g(\mathbf{X})d\mathbf{X}} \tag{3.12}$$

$$\hat{I}_N(f) = \frac{\sum_{i=1}^{N} f(\mathbf{X}^{(i)})w(\mathbf{X}^{(i)})}{\sum_{i=1}^{N} w(\mathbf{X}^{(i)})} \tag{3.13}$$

$$\hat{p}_N(\mathbf{X}) = \frac{\sum_{i=1}^{N} \delta_{\mathbf{X}^{(i)}}(\mathbf{X})w(\mathbf{X}^{(i)})}{\sum_{i=1}^{N} w(\mathbf{X}^{(i)})} \tag{3.14}$$

The accuracy of the estimation by importance sampling depends on the variability of $w(i)$. When $w(i)$ have large variance, it means that the estimation will be effectively based on only a few samples with the largest weights, and hence the approximation of $g(\mathbf{X})$ to $p(\mathbf{X})$ is not valid. When x is high-dimensional and $p(\mathbf{X})$ is multi-modal, the specification of a usable auxiliary distribution $g(\mathbf{X})$ usually becomes very difficult. In the ICONDENSATION algorithm [16], Isard and Blake do not combine two available measurements to evaluate the resemblance between hypothesis samples and ground-truth, which is a strategy widely adopted by most other visual tracking work. Instead, they make use of one of the measurements to generate the desired auxiliary distribution $g(\mathbf{X})$. This is a reasonable method to realize importance sampling for complex distributions. The disadvantage is that some independent information of different measurements is discarded and can not be utilized to weight the samples.

3.4 Sequential Monte Carlo Sampling

Based on the discussions in Section 3.2 and Section 3.3, and the problem formulation in Section 3.1, the Sequential Monte Carlo Sampling method is ready to apply. We wish to use N samples (or particles) at time step $k-1\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^{N}$which are approximately distributed according to the distribution $p(\mathbf{X}_{k-1} \mid \mathbf{Y}_{1:k-1})$ to compute the sample (or particle) set at time step $k\{\mathbf{X}_k^{(i)}\}_{i=1}^{N}$. In an ideal situation, we can represent the posterior according to Equation by

drawing $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$ from it. Unfortunately directly sampling from the posterior is impossible, so we will introduce an appropriate importance proposal distribution $g(\mathbf{X}_k \mid \mathbf{Y}_{1:k})$ from which we can draw samples to help accomplish the task. The posterior will be represented according to Equation 3.11 and the weights are given by:

$$w(\mathbf{X}_k^{(i)}) = \frac{p(\mathbf{X}_k^{(i)} \mid \mathbf{Y}_{1:k})}{g(\mathbf{X}_k^{(i)} \mid \mathbf{Y}_{1:k})} \tag{3.15}$$

From Equation 3.4 it follows that:

$$p(\mathbf{X}_k^{(i)} \mid \mathbf{Y}_{1:k}) = p(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)}) p(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}) p(\mathbf{X}_{k-1}^{(i)} \mid \mathbf{Y}_{1:k-1}) \tag{3.16}$$

and we know:

$$g(\mathbf{X}_k^{(i)} \mid \mathbf{Y}_{1:k}) = g(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_k) g(\mathbf{X}_{k-1}^{(i)} \mid \mathbf{Y}_{1:k-1}) \tag{3.17}$$

Substituting the above equations into Equation 3.15 ,we obtain:

$$\begin{aligned} w(\mathbf{X}_k^{(i)}) &= \frac{p(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)}) p(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}) p(\mathbf{X}_{k-1}^{(i)} \mid \mathbf{Y}_{1:k-1})}{g(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_k) g(\mathbf{X}_{k-1}^{(i)} \mid \mathbf{Y}_{1:k-1})} \\ &= w(\mathbf{X}_{k-1}^{(i)}) \frac{p(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)}) p(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)})}{g(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_k)} \end{aligned} \tag{3.18}$$

A method to recursively evaluate the posterior is now available, but we still need to specify the proposal distribution $g(\cdot)$. The optimal proposal distribution is given by

$$g(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_k) = p(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_k) \tag{3.19}$$

However, for simplification, the traditional particle filters use the transition prior to generate samples:

$$g(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_k) = p(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}) \tag{3.20}$$

Then the weights become:

$$w(\mathbf{X}_k^{(i)}) = w(\mathbf{X}_{k-1}^{(i)}) p(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)}) \tag{3.21}$$

Although attractively simple for implementation, this approach usually results in important weights with high variance. The reason is obvious: by comparison of $g(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)})$

and $p(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)})$, we can see that the observation at time step k - $\mathbf{Y}_k$ is lost by this simplification.

We summarize a typical particle filter step as follows:

The Particle Filtering Algorithm

At time step k, starting with a sample set:$\{\mathbf{X}_{k-1}^{(i)}, w(\mathbf{X}_{k-1}^{(i)})\}_{i=1}^{N}$:

1 **Selection:** select a new set of samples $\{\hat{\mathbf{X}_k^{(i)}}\}_{i=1}^{N}$ from $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^{N}$ according to $w(\mathbf{X}_{k-1}^{(i)})$. The samples with a larger weight should be selected with a higher probability.

2 **Prediction:** Sample from the importance function :

$$\{\mathbf{X}_k^{(i)}\} \sim g(\mathbf{X}_k^{(i)} \mid \hat{\mathbf{X}_k^{(i)}}, \mathbf{Y}_k) \; i = 1, 2, \ldots, N \tag{3.22}$$

3 **Measurement:** Evaluate the weight for each sample:

$$w(\mathbf{X}_k^{(i)}) = \frac{p(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)}) p(\mathbf{X}_k^{(i)} \mid \hat{\mathbf{X}_k^{(i)}})}{g(\mathbf{X}_k^{(i)} \mid \hat{\mathbf{X}_k^{(i)}}, \mathbf{Y}_k)} \; i = 1, 2, \ldots N \tag{3.23}$$

where the $p(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)})$ is the image likelihood and $p(\mathbf{X}_k^{(i)} \mid \hat{\mathbf{X}_k^{(i)}})$ is the dynamical model. Then normalize the weight so that $\sum_{i=1}^{N} w(\mathbf{X}_k^{(i)}) = 1$

4 **Representation:** Estimate the state at time step k as:

$$\widetilde{\mathbf{X}}_k = \arg\max_{\mathbf{X}_k^{(i)}; i=1,\ldots,N} w(\mathbf{X}_k^{(i)}) \tag{3.24}$$

or:

$$\widetilde{\mathbf{X}}_k = E[\mathbf{X}_k] = \sum_{i=1}^{N} w(\mathbf{X}_k^{(i)}) \mathbf{X}_k^{(i)} \tag{3.25}$$

Here the measurement represented by Equation 3.23 does not include the item $w(\mathbf{X}_{k-1}^{(i)})$ since this factor has already been considered when we do the selection. Note that in the prediction stage, if we sample from dynamical model according to Equation 3.20 and in the measurement stage, weight the samples according to Equation 3.21 instead, then the algorithm reduces to a traditional CONDENSATION particle filter. We illustrate the particle filtering process with a 10-sample example in Figure 3.1.

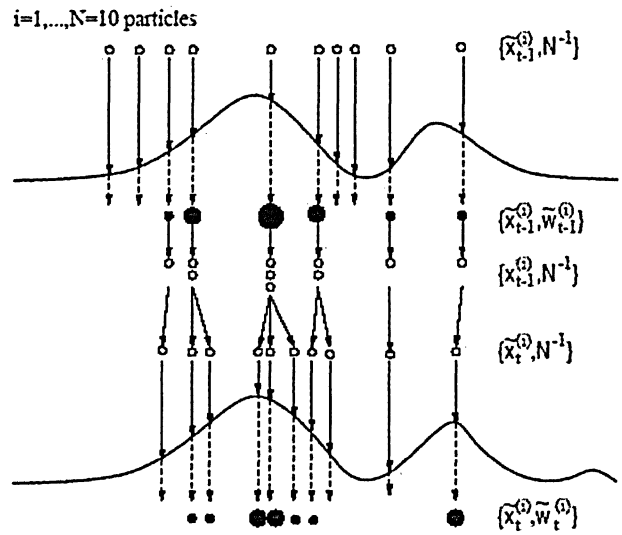


Figure 3.1: Illustration of particle filtering process

3.5 Some Implementation Issues

Sample Selection: At the beginning of each time step, a new set of particles need to be selected from the posterior particle set of last time step according to their weights. The following is a simple and efficient selection strategy:

1. generate a random number from uniform distribution $\beta \in [0, 1]$.

2. based on binary subdivision, find the smallest i which satisfies $c(\mathbf{X}_{k-1}^{(i)}) \geq r$. Then $c(\mathbf{X}_{k-1}^{(i)})$ is determined after $w(\mathbf{X}_{k-1}^{(i)})$ are normalized in the last time step by

$$c(\mathbf{X}_{k-1}^{(0)}) = 0 \tag{3.26}$$

$$c(\mathbf{X}_{k-1}^{(i)}) = c(\mathbf{X}_{k-1}^{(i-1)}) + w(\mathbf{X}_{k-1}^{(i)}) \tag{3.27}$$

3. select $\mathbf{X}_{k-1}^{(i)}$ as $\hat{\mathbf{X}}_{k}^{(i)}$ for successive processing.

Decision Making: The ultimate goal of human motion tracking is to find the pose configuration from video rather than to approximate a distribution. There are several ways to

develop a representation for a desired state vector. As we mentioned in Section 3.2., the two most popular choices are the maximum likelihood representation and the expectation representation as in Equation 3.24 and Equation 3.25. The expectation decision is a more stable estimation. It ensures that there will not be any too nonsensical or ridiculous result. But if there are multiple maxima in the posterior distribution, or even if there is only one maximum but the distribution is quite flat, the result obtained by this approach will be greatly biased. Therefore, we prefer the maximum likelihood decision in our work since it represents the optimal solution available at the current iteration. The flaw of this method is that sometimes the estimate tends to "jump around" over time when the number of particles is limited because the condition of law of large numbers is not well satisfied.

Number of Particles: According to the law of large numbers, we know that the approximation to the posterior gets improved when the number of particles increase. However, we can not use too many particles since the factor of computational cost must be considered. It is reported by some literature that for the traditional CONDENSATION algorithm the number of particles N will be sufficient if:

$$N \geq \frac{D_{\min}}{\alpha^d} \tag{3.28}$$

where $D_{\min}$ is the survival diagnostic and α is the survival rate [45]. These two parameters evaluate the effective number of particles after one filtering iteration. Since $\alpha \ll 1$, for a high-dimensional problem like human motion tracking, the required number will be prohibitively large. But with some improvement strategy, this anxiety can be relieved to a large extent.

Chapter 4

From 3D World to 2D Images

Human tracking from monocular video sequences is basically a 3D reconstruction problem based on 2D image information. It involves a lot of projections and geometric transformations. In this chapter we will discuss the spatial relationships between the articulated human model, the scene and the camera.

4.1 Rigid Geometric Transformations

According to the knowledge of kinematics, individual motions are roughly categorized into rigid motions and non-rigid motions. Non-rigid motions can be regarded as deformations which change the shape of an object. The deformable motions of any point on the object are independent of those of other points and must be determined separately. Therefore many control points have to be set to describe this kind of motion. In case of rigid motions, the relative distance between any two sets of points on the same object remains invariant, and then the 3D structure of objects can be modelled as a non-deformable surface [59]. A very important result from this assumption is that the motion parameters (displacement, velocity, acceleration, direction etc.) of each single point on a rigid object are sufficient to represent the motion status of the whole object. In our work, although the human body as a whole is deformable, its every single part can still be assumed as a rigid object and to undertake rigid motions. Except those caused by extremely loose fittings, in most cases the non-rigid motions associated with body segments are trivial and thus can be ignored. Actually we are

more interested in describing human motions as different combinations of the state of each body part, so we only consider rigid motions here.

Although rigid geometry transformations are developed to describe rigid motions in a 3D coordinate system, it can also be utilized to describe the relative positions of two objects. Actually the latter case is encountered even more frequently in our work. When we design an articulated human body model, we must consider the fact that each body part has different effects on others. For instance, when the torso moves, the positions of all the other body parts are unavoidably changed with it. However, when the upper arms are in motion, only the lower arms and hands are affected. The movement of hands can change the position of no other body parts but itself. Therefore, it is usually convenient to build the human body model hierarchically and describe the spatial location of body parts hierarchically. On the other hand, to simulate the image formation process we often need to switch between multiple different coordinate systems. Both of these implementations require handling rigid transformations.

Only two kinds of rigid transformations are defined: translation and rotation. A rigid transformation can be fully described by a translation matrix $\mathbf{T}$ and a rotation matrix $\mathbf{R}$:

$$\mathbf{P}' = \mathbf{R}\mathbf{P} + \mathbf{T} \tag{4.1}$$

where $\mathbf{P} = [x, y, z]^T$ and $\mathbf{P}' = [x', y', z']^T$ are the coordinates of the point before and after the transformation. $\mathbf{R}$ is a 3×3 Matrix and $\mathbf{T} = [T_x, T_y, T_z]^T$ is the translation vector. A rigid rotation can be regarded as a combination of rotations around three coordinate axes. Assume that the Eulerian angles of rotations about x axis, y axis and z axis are θ_x, θ_y and θ_z, respectively, the individual rotation matrices are:

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}, \mathbf{R}_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix}, \mathbf{R}_z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

and thus the overall rotation matrix is given by:

$$\mathbf{R} = \mathbf{R}_x\mathbf{R}_y\mathbf{R}_z = \begin{bmatrix} \cos\theta_y\cos\theta_z & -\cos\theta_y\sin\theta_z & \sin\theta_y \\ \sin\theta_x\sin\theta_y\cos\theta_z + \cos\theta_y\sin\theta_z & \cos\theta_y\cos\theta_z - \sin\theta_x\sin\theta_y\sin\theta_z & -\sin\theta_x\cos\theta_y \\ -\cos\theta_x\sin\theta_y\cos\theta_z + \sin\theta_x\sin\theta_z & \cos\theta_x\sin\theta_y\sin\theta_z + \sin\theta_y\cos\theta_z & \cos\theta_x\cos\theta_y \end{bmatrix} \tag{4.3}$$

It is also shown in [46] that the matrix for a rotation around an arbitrary axis through the origin is:

$$\mathbf{R}_{\vec{\mathbf{n}}} = \begin{bmatrix} n_x^2 + (1-n_x^2)\cos\alpha & n_xn_y(1-\cos\alpha) - n_z\sin\alpha & n_xn_z(1-\cos\alpha) + n_y\sin\alpha \\ n_xn_y(1-\cos\alpha) + n_z\sin\alpha & n_y^2 + (1-n_y^2)\cos\alpha & n_yn_z(1-\cos\alpha) - n_x\sin\alpha \\ n_xn_z(1-\cos\alpha)0n_y\sin\alpha & n_yn_z(1-\cos\alpha) + n_x\sin\alpha & n_z^2 + (1-n_z^2)\cos\alpha \end{bmatrix} \tag{4.4}$$

where $\vec{\mathbf{n}}$ is a vector from (0,0,0) to (n_x, n_y, n_z) and α is the Eulerian angle of rotation around $\vec{\mathbf{n}}$. We plot the two kinds of rotations in Figure 4.1.

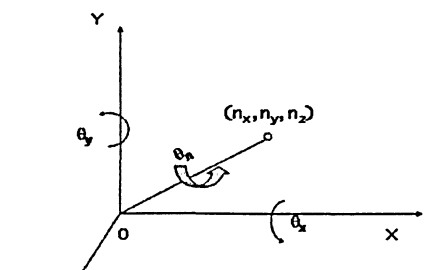


Figure 4.1: Rotations about three coordinate axes and arbitrary axis.

4.2 Homogeneous Coordinates and the Change of Coordinates

Equation 4.1 is not a convenient representation for rigid transformation. It is more effective to use a single matrix describing both the translations and rotations so as to increase the implementation efficiency. This goal can not be achieved in the old Cartesian coordinates because of the existence of addition operation. A more appropriate coordinate representation,

homogenous coordinates, is then introduced to address the problem. The definition of the homogenous coordinates for a point with Cartesian coordinates $[x, y, z]^T$ is:

$$\mathbf{P}_h = \begin{bmatrix} sx \\ sy \\ sz \\ s \end{bmatrix} \tag{4.5}$$

where the s denotes a scale factor. Note that although the mapping from the Cartesian coordinates to the homogeneous coordinates is unique this is not true inversely. For example, the homogenous coordinates $[x, y, z, 1]^T$ and $[x/2, y/2, z/2, 1/2]^T$ refer to the same point in Cartesian coordinates. Now the rigid transformation can be described by a single 4×4 matrix $\mathbf{M}$ when $\mathbf{P}$ and $\mathbf{P}$' are represented by homogenous coordinates:

$$\mathbf{P}' = \mathbf{MP} \tag{4.6}$$

or explicitly:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4.7}$$

where $\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ is the rotation matrix. Now if we want to change coordinate system, all the points in one system can be mapped to the other through Equation 4.6 provided matrix $\mathbf{M}$ is known.

4.3 Perspective Projection

The pinhole perspective projection model, which is firstly proposed in the 15th century by Brunelleschi is mathematically convenient and provides acceptable accuracy for the approximation of the image formation process. In this model we regard the camera that is used to capture videos as an ideal pinhole camera. According to the principles of geometrical optics, all the rays reflected by an object will pass through the center of camera lens. For

this reason it is also called the central projection. Real perspective projection yields an inverted image, but for convenience usually a virtual image plane is assumed to be positioned in front of the camera and symmetric to the actual one with respect to the pinhole. Figure 4.2 illustrates the perspective projection. From Figure 4.3 we can derive algebraic relations

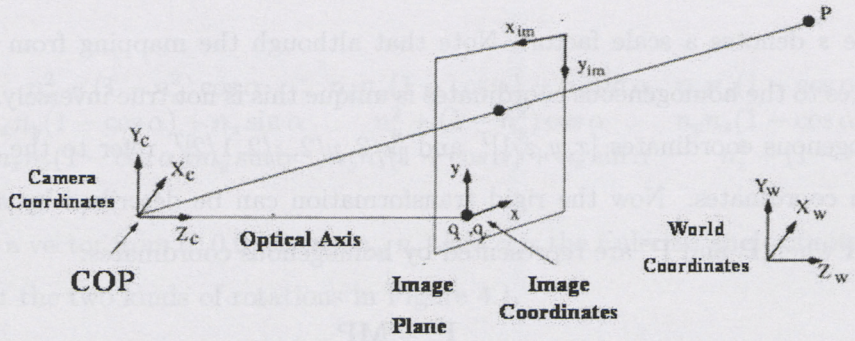


Figure 4.2: The perspective projection

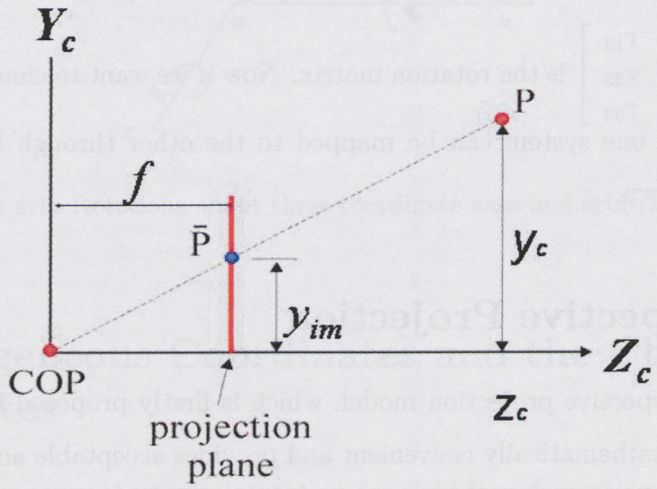


Figure 4.3: Derivation of perspective transformation

for the projection transformation easily with the knowledge of similar triangles:

$$\begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix} = \begin{bmatrix} f\frac{x_c}{z_c} \\ f\frac{y_c}{z_c} \end{bmatrix} \tag{4.8}$$

where (x_{im}, y_{im}) are the image coordinates of the projection of $\mathbf{P}$, (x_c, y_c, z_c) are the camera coordinates of $\mathbf{P}$ and f is the focal length of the camera. When the coordinates of $\mathbf{P}$ are given in the form of world coordinates (x_w, y_w, z_w), we need additional specification of the rigid transformation matrix for calculating the image coordinates. The transformation characterized by Equation 4.8 is nonlinear. However, we can make a linear mapping for perspective projection by transferring into the homogenous coordinate system:

$$\begin{bmatrix} lx_{im} \\ ly_{im} \\ l \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} kx_c \\ ky_c \\ kz_c \\ k \end{bmatrix} \tag{4.9}$$

where l, k are scale factors as the s in the definition of homogenous coordinates.

4.4 Camera Calibration

From perspective projection we see that to simulate image formation several parameters have to be known first. They include the center of image plane, the focal length of camera and the relation between camera coordinates and world coordinates. We classify these parameters as intrinsic ones and extrinsic ones. Intrinsic parameters are solely determined by the inherent properties of a camera, including the focal length, center of image plane, distortion of lens and skew coefficients. Extrinsic parameters describe the spatial characteristics of a camera, such as the position and orientation of the camera with reference to a defined world coordinate system. The process of estimating these camera parameters is called camera calibration. In this process we assume that some known features such as points or lines with known positions in a reference coordinate system are available. Hence the calibration can be considered as an optimization problem where the discrepancy between the observed image features and their theoretical positions is minimized with respect to the camera's intrinsic and extrinsic

parameters. There are two main steps for camera calibration. In the first step intrinsic parameters are obtained. Based on them, extrinsic parameters are then calculated in the second step.

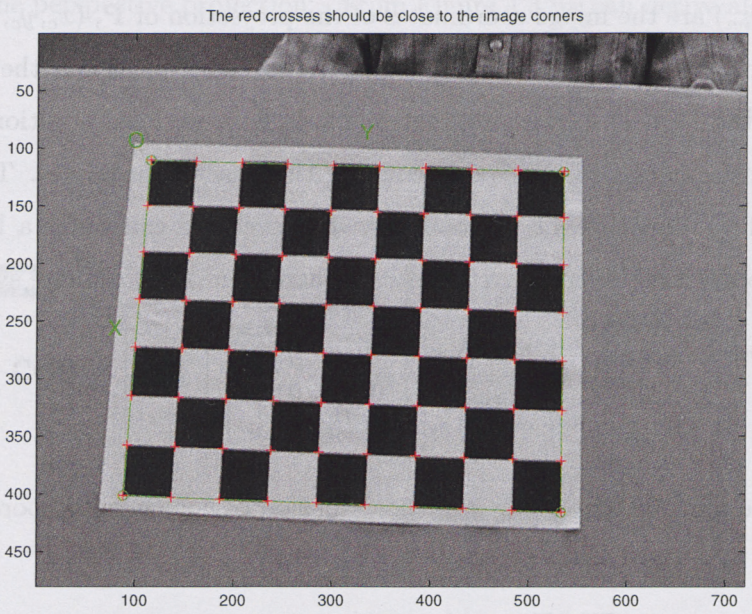


Figure 4.4: Camera calibration

We use the MATLAB camera calibration toolbox which is developed by Intel to aid our calibration. The toolbox is based on perspective projection theory and the calibration algorithm developed by Heikkila and Silven [46]. The procedure of calibration with this toolbox is as follows: Firstly, a checkerboard pattern with known size is created. We vary the orientation and position of the pattern and capture the process as a video sequence.Then a suitable number of frames were selected as the calibration images. Usually a sequence with 20-30 images is sufficient for the convergence of result. Too many images will only make the procedure unnecessarily burdensome. For each calibration image we mark the four corners of the checkerboard pattern and specify the size of the grids. Then the toolbox will try to extract corners of each grid. The estimate of the grid corners are subject to rectification from the user. In this manner the mean square error of the estimate can be calculated.

After a certain number of iterations, the estimate of grid corners will become accurate and the calibration result converges. The algorithm in [46] adopts an 8-DOF camera model. Since we are only interested in some of the parameters for the simulation of perspective projection, those parameters which describe the lens distortion and the skew of camera are ignored due to the fact that these effects are not evident in our work. The estimate of extrinsic parameters also follows the procedure of marking checkerboard pattern corners and automatically extracting grid corners, but it is only implemented for a single image (iteration) based on the known intrinsic parameters. The origin of the world coordinate system is specified as one of the checkerboard pattern corners. Displacement of the camera with regard to this corner is encoded in the form of translation and rotation matrixes. It is beneficial to select a fixed position in the shooting scene and make it coincide with the assumed world coordinate origin. If the subject starts from this position when we capture the human motion video, the initialization for the tracker will become much easier, since 6 DOFs are already determined by the extrinsic parameters. Figure 4.4 illustrates an intermediate step of our camera calibration procedure with the MATLAB toolbox and Figure 4.5 illustrates the extrinsic parameter estimate result.

4.5 3D Articulated Human Body Model

Following the typical human motion tracking method, we build a 3D articulated model which is composed of 14 segments to represent human body. These segments include head, neck, upper and lower limbs, torso and feet. All of them are modelled as truncated cones except the head, which is represented by a sphere. Geometric parameters such as the radius and height of the truncated cones correspond to anthropometric data of humans, so they are assumed to be constant during the whole tracking process. We have not realized automatic initialization by far so these values are obtained through a manual initialization procedure.

As we mentioned earlier, this model has a hierarchical structure so as to address the different effect yielded by the motion of each body segment on others. We illustrate this

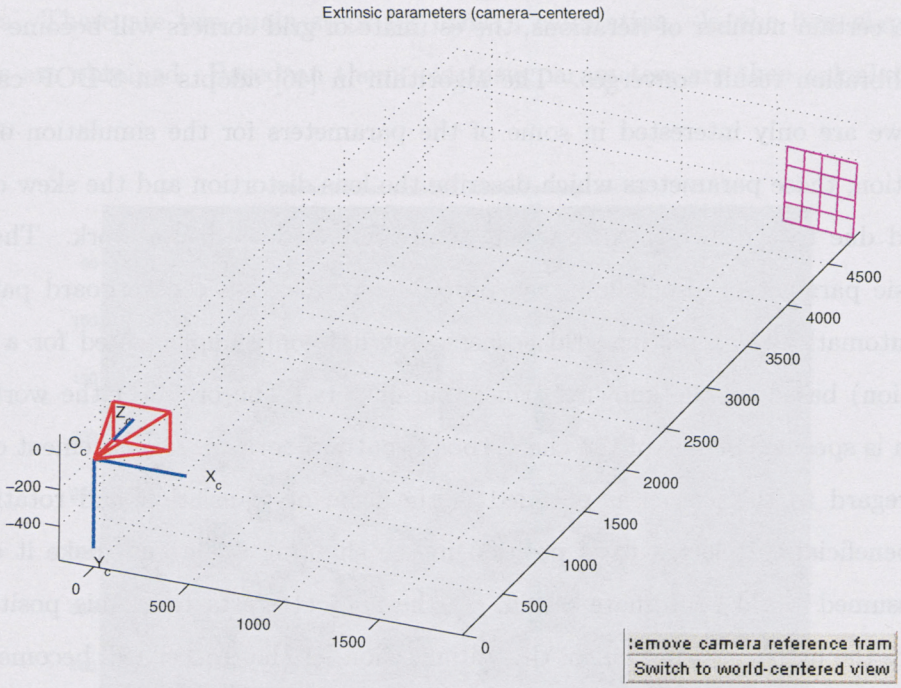


Figure 4.5: Extrinsic parameter estimate result

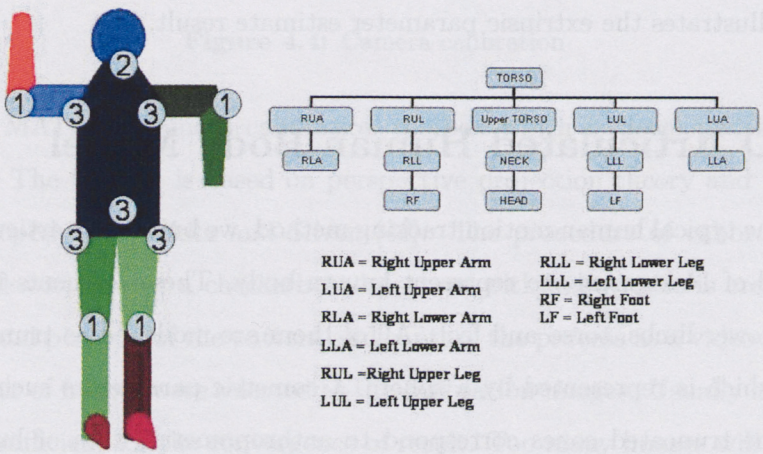


Figure 4.6: The proposed 3D articulated human body model and its hierarchical structure

structure in Figure 4.6. As we can see, the root segment is the torso; all the other body

parts form its child branches or grandchild branches. Thus, each body part can only affect its own children. We define a global coordinate system for the model with the origin located at the bottom center of torso. Each body part also has its own local coordinate system. Its origin coincides with the joint which connects this body part to its parent. Figure 4.7 illustrates three connected limbs: a lower arm, an upper arm and a torso. According to the rigid transformation we have discussed, the points on the lower arm have model coordinates which can be represented as:

$$\begin{bmatrix} \lambda X_m \\ \lambda Y_m \\ \lambda Z_m \\ \lambda \end{bmatrix} = \mathrm{M}_{1m} \begin{bmatrix} \gamma X_1 \\ \gamma Y_1 \\ \gamma Z_1 \\ \gamma \end{bmatrix} = \mathrm{M}_{1m}\mathrm{M}_{21} \begin{bmatrix} \tau X_2 \\ \tau Y_2 \\ \tau Z_2 \\ \tau \end{bmatrix} \tag{4.10}$$

where M_{1m} and M_{21} are 4×4 homogenous geometry transformation matrices which transfer upper-arm coordinates into model coordinates and lower-arm coordinates into upper-arm coordinates. Therefore, at any moment as long as we have the knowledge of the joint angles,

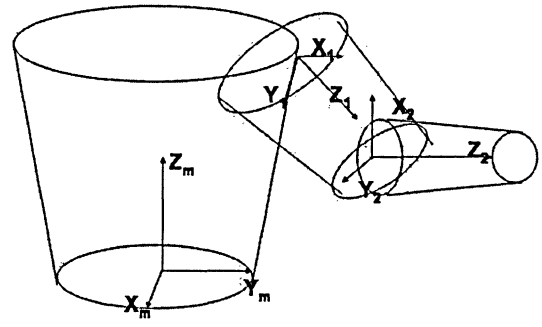


Figure 4.7: Connected body segments and their local coordinate systems

we will be able to determine the global model coordinates for any point on the body model. Armed with the information specifying the relation between the model coordinate system and the world coordinate system, we can further determine the position of the point in the world frame. It is based on this fact that joint angles plus global translation and rotation parameters are said to be "complete" for describing the pose and the position of a human. They together form the DOFs of a human body model. Note that the local translation

Joint	Rotation Axis	Range (Degree)
Neck	x	90
	y	90
Shoulder	x	225
	y	255
	z	180
Elbow	y	180
Hip	x	180
	y	135
	z	180
Knee	y	150

Table 4.1: Range of joint angles

matrix is already known in the form of geometrical parameters of body model segments. To reduce the dimension of state space we truncate some unimportant joint angles. For example, although feet are present in our model, no DOF is assigned to the ankle joints. Hands are not separately modelled but regarded as the extension of lower arms. As a result, there are 24 DOFs assigned to this model in total. Among them 18 are joint angles, 3 global translation parameters (T_x, T_y, T_z) which describe the position of hip joint in the world coordinate system, and 3 global rotation parameters (ψ_x, ψ_y, ψ_z). We illustrate the model in Figure 4.6 with all the DOFs marked.

Joint limits are widely adopted to avoid unreasonable configurations of articulated human model. The range of joint angles used in our work is summarized in Table 4.1. They are learned from common knowledge and are just rough. Further narrowing down the ranges is possible if we investigate on the statistics of human kinematics in the future.

4.6 Summary

In this chapter we have presented all the factors concerning the relationship between 2D images and the 3D world. Their roles in a human tracking system can be summarized as follows. Rigid transformation allows us to determine the spatial relationship between model coordinates, camera coordinates and world coordinates. The movement of any human body

part is first described in a model coordinate system to form a 3D pose together with motions of other body parts. Then the 3D pose is positioned in a 3D world frame and generates an image on the projection plane of a camera through the perspective projection. The necessary parameters for simulating the perspective projection come from camera calibration. The consideration about computation efficiency drives us to choose homogenous coordinates to facilitate implementations in these procedures. We show the connections by a block diagram in Figure 4.8.

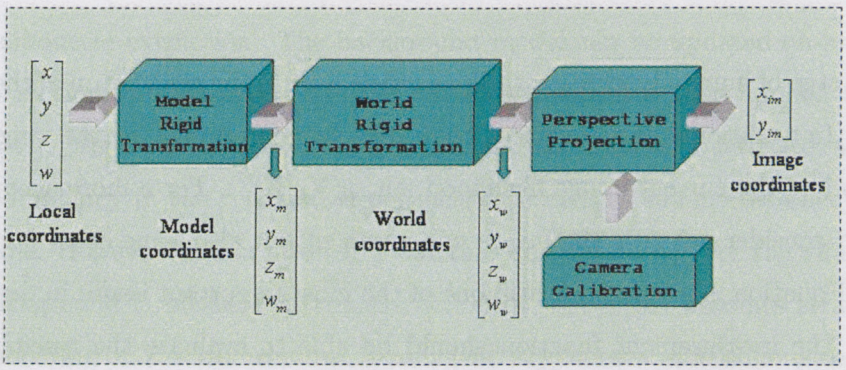


Figure 4.8: The image formation process

An image obtained through this process is called a hypothesis observation since it is generated by the articulated human model. In contrast, the captured human motion video sequence is called the ground-truth observation. Assume that there are no intolerable errors in the simulation of image formation, the model's pose which is similar to the ground-truth human pose should lead to a hypothesis image very close to the ground-truth image. This is the principle for judging the correctness of a tracking hypothesis, and it poses another problem which is to be discussed in the next chapter.

Chapter 5

Fusing Multi-Cue for Tracking

A critical step of human tracking with the particle filter is the calculation of the weights of the particles. In traditional particle filters, they are solely determined by the weights in previous iterations and the current image likelihood, i.e. $p(\mathbf{Y}_k|\mathbf{X}_k^{(i)})$. For a more accurate weighting, we should consider not only the image measurement but also some other factors, as we can see from Equation 3.18. But it is still one of the most important issues in designing a visual tracker. The measurement function should be able to evaluate the resemblance between image features generated by hypothesis and those generated by ground-truth human pose, as the criterion for judging the correctness of hypothesis. An ill-conditioned measurement function will produce an undesirable effect on the tracking result or may even lead to total failure. In this chapter, realizing the importance of the measurement function, we specify our method in building a robust measurement function which fuses multiple image cues.

5.1 Data Acquisition and Silhouette Extraction

The color video sequences used in our work are captured by a digital video camera in various indoor environments. During the capture process we ensure that there are no evident changes or moving objects other than the human subject in the scene. In some of them, a bluescreen background is used. The camera is located in a fixed position and there are no zoom effects in the video sequence. In other words, the intrinsic and extrinsic parameters remain invariant. The lighting condition is normal and the subject is equipped with no markers.

The color images can be directly used to generate intensity cues, but features such as area of silhouette and boundary usually need silhouette images as a starting point. The conditions of data acquisition make it possible to extract silhouettes by implementing background subtraction. To this end, we build a background model though an N-frame (usually $N > 100$ is sufficient) pure background video sequence under the assumption that the intensity value of each background pixel follows a Gaussian distribution. Pixel differencing is implemented between the human motion video frames and the mean of background model. Then a threshold with the value 2-3 times of standard deviation of background pixel is set for binary silhouette extraction. The background model can be updated on-line with each new human motion video frame coming in. Shadow removal is then applied to refine the silhouettes [47]. Figure 5.1 shows a video frame and the final silhouette extraction result. To avoid redundancy in the repeated experiments we extract silhouettes once for each sequence off-line. However we can also carry out the process on-line. For the video sequences in which background is bluescreen, the extraction of silhouettes becomes straightforward. Direct thresholding on the Hue channel is sufficient to generate good silhouettes.

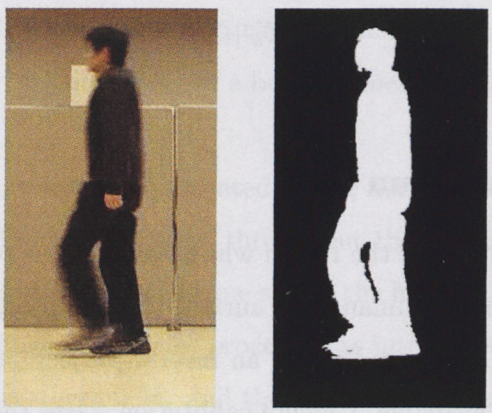


Figure 5.1: Silhouette extraction result.

5.2 Measurement Function

Our goal is to design a measurement function which has a significant peak corresponding to the ground-truth configuration. We must also reduce as much computational cost as possible since sample weighting is the most time-consuming step in particle filter iterations. We have learned the importance of multi-cue fusion for building a robust measurement function in Chapter 2. In our work we choose area of silhouette, color and boundary as image features to track.

5.2.1 Area of Silhouette

Given a silhouette S' which is extracted from the image projected from a hypothesis pose $\mathbf{X}_k^{(i)}$, we compare it with the observed silhouettes S, which is generated by ground-truth human pose. The pixels are categorized into 2 groups, R_1 and R_2 with $R_1 = S \cap S$ ' and $R_2 = S \cup S$ '. Let the number of pixels in R_1 and R_2 be N_1 and N_2, respectively, and then the silhouette area measurement density can be represented by:

$$p_1(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)}) = \frac{N_1}{N_2} \tag{5.1}$$

5.2.2 Color Histogram

We know that image pixels in the region which corresponds to the human subject are generated by the points on the human body surface through perspective projection. If we want to build correspondences between them, an inverse projection is necessary in the case that image observations are available. Camera calibration solves the focal length and the relation between camera coordinates and world coordinates, and the initialization of the human body model provides the relationship between the local segment coordinates, global body coordinates and the world coordinates. Note also that we assume the human body model coincides with the real human body after initialization. With the above knowledge and assumptions

we are able to make an inverse projection from the image to the 3D model:

$$\mathbf{P}_l = \mathbf{M}_{lm}^{-1}(\mathbf{X}_0, \mathbf{M}_{mw}^{-1}(\mathbf{Q}^{-1}(\mathbf{P}_i, \mathbf{E}))) \tag{5.2}$$

where $\mathbf{P}_m$ and $\mathbf{P}_i$ are the body part local coordinates of a point and the image coordinates of its projection, respectively; $\mathbf{Q}$ is the perspective projection; $\mathbf{E}$ is the extrinsic camera parameter learned from calibration; $\mathbf{M}_{mw}$ and $\mathbf{M}_{lm}$ are the rigid transformation relating the model coordinates to the world coordinates and that relating the body segment local coordinates to the model coordinate, respectively; and $\mathbf{X}_0$ is the initial pose learned from initialization.

After the correspondence between model points and image pixels is built through Equation 5.2, we can construct a reference appearance model for each individual segment of the human body. The normalized color histogram is calculated for the surface of the segment in RGB color space. The histogram has f bins (in our experiment $10 \times 10 \times 10$ bins are used) for each single color channel. Since the monocular videos are a 2D projection of the 3D scene, unavoidably observations for part of the human body surface will be missing. To handle this problem, we make some justified assumptions such as the left-right symmetry of the appearance. We also assume that in the initialization stage there are only limited occlusions existing and the occluded part of a body segment has the same color distribution as the visible part.

The reference appearance model represented by the normalized histogram is regarded as ground-truth and assumed almost constant throughout the tracking process. Starting from the first frame after initialization, at each time step k the histogram $\mathbf{H}_k^{(i)}$ is built for hypothesis state vector $\mathbf{X}_k^{(i)}$ by following a similar procedure as introduced above. Traditionally the difference of the reference histogram $\mathbf{H}_r$ and the hypothesis histogram $\mathbf{H}_k^{(i)}$ is measured by summing the Bhattacharyya distance of all the individual body segment histograms [46,47]:

$$D(\mathbf{H}_r, \mathbf{H}_k^{(i)}) = \sum_{m=1}^{14} \sum_{j=1}^{3} D(\mathbf{H}_r, \mathbf{H}_{k,(j,m)}^{(i)}) \tag{5.3}$$

where

$$D(\mathbf{H}_r, \mathbf{H}_{k,(j,m)}^{(i)}) = \sqrt{1 - \sum_{n=1}^{f} \sqrt{\mathbf{H}_{r,(n,j,m)} \mathbf{H}_{k,(n,j,m)}^{(i)}}} \quad , \tag{5.4}$$

k is time step index, n is the bin index, j is RGB channel index, and m is body segment index.

However, there is an apparent flaw in Equation 5.3: it does not consider the relative importance of different body parts. Actually errors in estimating the position of the torso almost always cause more trouble than errors in estimating the position of a foot. Therefore we propose to assign different weights for the histogram of each segment in summing, which turns Equation 5.3 into:

$$D(\mathbf{H}_r, \mathbf{H}_k^{(i)}) = \sum_{m=1}^{14} \alpha_m \sum_{j=1}^{3} D(\mathbf{H}_r, \mathbf{H}_{k,(j,m)}^{(i)}) \tag{5.5}$$

where α_m denotes the weights of histogram for each individual body segment. This weight is proportional to the area of the image patch projected by the body part of interest. We will show in Chapter 7 how this change improves the performance of the tracker significantly.

The color measurement distribution can then be formulated as:

$$p_2(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)}) = e^{-\beta D^2(\mathbf{H}_r, \mathbf{H}_k^{(i)})} \tag{5.6}$$

where β is a scalar which helps the result evaluated by Equation 5.6 more reasonably distributed in the range of (0,1). To make the reference appearance model adaptive to the variation of lighting conditions in video, an update process can be applied:

$$\mathbf{H}_{r,k}^{+} = \lambda \mathbf{H}_{r,k}^{-} + (1 - \lambda)\mathbf{H}_{r,k-1}^{+} \tag{5.7}$$

where the sign + and - distinguish the reference appearance model both after and before the update has occured.

5.2.3 Boundary

Boundaries are often confused with edges and contours. Here we define the boundary as the outer border of an object without any circles inside. Therefore we can not use typical edge and contour extraction method for boundary extraction. Instead, a morphology operator is applied to the silhouettes S [1]:

$$B = S - S \ominus M \tag{5.8}$$

where M is an structuring element and $\ominus$ signifies erosion: $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$. An example of a boundary extraction result is shown in Figure 5.2.

The boundary B can be represented by the image coordinates of pixels on it: $b(n) = [x(n), y(n)]$, for n = 0,1,2,....N-1. N is the total number of pixels on B. Treating $b(n)$ as a complex number:

$$b(n) = x(n) + jy(n) \tag{5.9}$$

we can implement DFT for $b(n)$:

$$B(f) = \frac{1}{N} \sum_{n=0}^{N-1} b(n) e^{-2\pi f \frac{n}{N}} \tag{5.10}$$

$B(f)$ is called the Fourier Descriptor (FD) of the boundary $b(n)$ [1]. The boundary information-based measurement density is then formulated as:

$$p_3(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)}) = e^{-\rho D(B(f)_k, B(f)_k^{(i)})} \tag{5.11}$$

where ρ has a similar function as the β in Equation 5.6 and $D(B(f)_k, B(f)_k^{(i)})$ is the Euclidean distance between the FD of the ground-truth boundary and the FD of the boundary generated by hypothesis $\mathbf{X}_k^{(i)}$.

For a boundary the number of pixels N is usually in hundreds or even thousands. However, in our work just the first 100 coefficients in $B(f)$ are already sufficient to capture the gross essence of a boundary and are able to roughly reconstruct the boundaries, as we can see from Figure 5.2. In fact this approximative representation is even more advantageous than the original one since high frequency components of the FD correspond to noise or trivial details, which should be eliminated. Using only low frequency components of the FD allows a strong emphasis to be laid on the relationship between the boundaries and human motions rather than on the errors of geometric fit caused by noise. Moreover, it can reduce the computational cost to a great extent with the help of the Fast Fourier Transform (FFT). An additional advantage of using the FD as a measurement is that it can be directly integrated into the human tracking framework. In this field, translations and rotations of

the model are estimated by the state vectors, so we do not need to do any modifications to make the FD insensitive to translation, rotation and scaling, a factor we would normally have to worry about in many shape analysis scenarios. On the contrary, we wish FD to be sensitive to those transformations. Otherwise the change of positions or poses of human can not be reflected by the FD. However, we do wish to avoid the FD's sensitivity to the starting point. To this end we can set a fixed corner of the boundary as the starting point. The FD has once been used for tracking in [28]. However, their work is significantly different with the function of the FD here in that they only use FD to determine motion parameters with respect to a known shape whilst we use the FD to calculate the degree of fit for what are initially unknown boundaries.

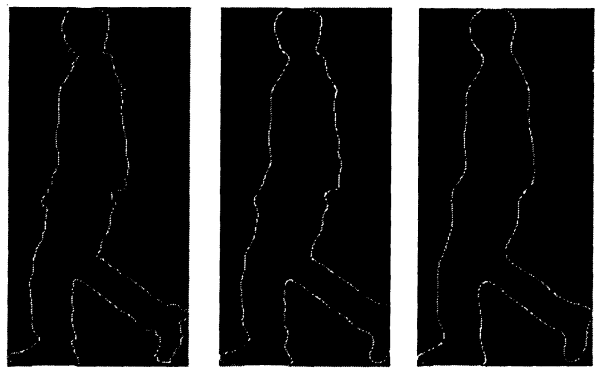


Figure 5.2: Boundary extraction result (left) and the boundary reconstructed from the first 100 (middle) and the first 50 (right) FD coefficients.

The Fourier Descriptor is not the only method for boundary representation. Shape signature is also aimed at describing boundary and shape [1]. Compared with it, the superiority of the FD is evident: Shape signature requires the origin to fall inside the shape, and for a uniform comparison, the origin is normally chosen as the point with mean x and y coordinates. However there is no guarantee that this point will always fall inside the boundary of a human silhouette. In contrast, the FD can be extracted from any boundary; Principle Component Analysis (PCA) is usually necessary to reduce the dimensions of shape signature, while with

FD we just select the first certain number of coefficients as mentioned above.

Figure 5.3 demonstrates the power of FD as a measurement feature for tracking. The images on top of the checkerboard are ground-truth observations and the images on the left of the checkerboard are generated by hypothesis. The gray-level values of those blocks are proportional to the Euclidean distance between the FDs of the ground-truths and the hypotheses. A dark block indicates a strong resemblance and a bright one indicates otherwise. As we expected the blocks along the diagonal axis are darkest among the row and the column they are located in. We can also observe that the block corresponding to the distance between the first and the third pose is rather dark as well. This can be explained by the similarity of boundaries generated by these two poses.

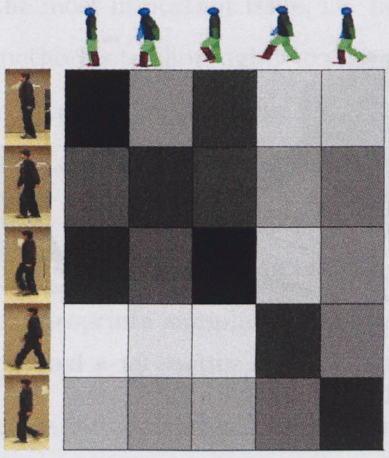


Figure 5.3: Euclidean distance between Fourier Descriptors extracted from ground-truth image and that extracted from hypothesis image.

5.2.4 Combination

We fuse the three image cues to build the overall measurement distribution:

$$p(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)}) = p_1(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)})^{\mu} p_2(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)})^{\nu} p_3(\mathbf{Y}_k \mid \mathbf{X}_k^{(i)})^{1-\mu-\nu} \tag{5.12}$$

The parameters μ and ν are used to adjust the relative weight of the 3 individual image likelihoods. Due to reasons discussed in Chapter 2, we still set them as constant. In our

experiments, equal weights $\mu = \nu = 0.33$ works well for all purposes. Figure 5.4 is an example of the proposed multi-cue fusion measurement function surface from two perspective. We illustrate it only in a two-dimensional space for the reason of tractability. The figure is plotted for a certain frame of a human motion sequence. The values are obtained by varying 2 DOFs of the human body model and keep all the other DOFs equal to the ground-truth data. The peak of the surface the global maximum we are searching for. Its distinction suggests the validity of our measurement function. In a 24-dimensional space the surface of a measurement function will be much more complicated, but exhibits similar properties.

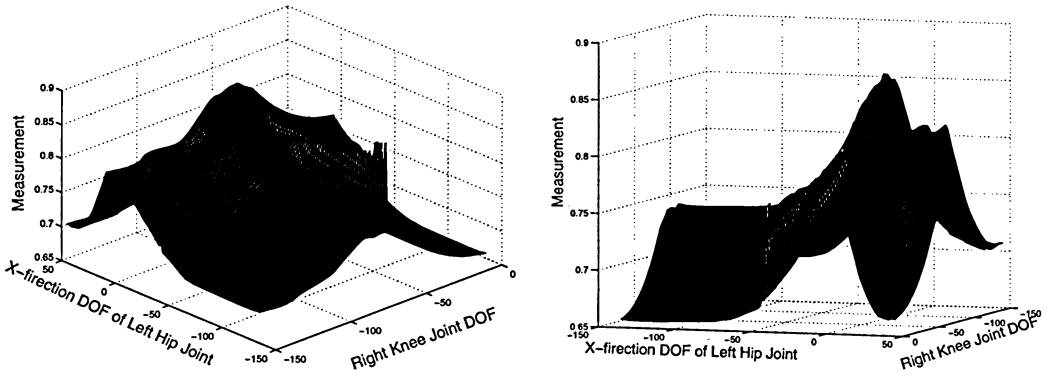


Figure 5.4: The measurement function surface for a human motion video frame w.r.t. 2 DOFs when other DOFs are ground-truth data.

Chapter 6

The DE-MC Particle Filter

We have addressed the problems on how to form images from hypotheses and how to evaluate the likelihoods for them, but the most important issue, i.e. how to generate the hypotheses, is still pending. The simplest method is to thoroughly exploit the state space. Unfortunately, for our articulated model-based tracking application there are 24 DOFs to be determined, which makes this idea totally unrealistic. We can also choose to follow the traditional CONDENSATION algorithm and some of its refined versions since they are much more efficient in exploiting the high dimensional space. However, in our experiments their performance are far from satisfactory due to inappropriate sampling strategies. Let us have a look back at the problem discussed in Chapter 3: Can we have a reasonable strategy for sampling from $g(\mathbf{X}_k \mid \mathbf{X}_{k-1}, \mathbf{Y}_k)$? The choice made by a general particle filter (CONDENSATION) is to sample from $p(\mathbf{X}_k \mid \mathbf{X}_{k-1})$ instead and that by the refined particle filters is to utilize an autoregressive dynamical model to sample from $p(\mathbf{X}_k \mid \mathbf{X}_{k-1}, \mathbf{X}_{k-2}, \ldots, \mathbf{X}_{k-m})$, but they both ignore the factor of observation. With a statistical technique which is called the Differential Evolution - Monte Carlo (DE-MC) algorithm, we are able to provide a better solution. We will start with the introduction to the related theoretical background, and then propose our novel DE-MC particle filter.

6.1 Markov Chain Monte Carlo

We informally described the definition of the Markov chain in Equation 3.2. For clarity we repeat it here: A stochastic process is called a Markov chain (MC) if the following condition holds:

$$p(\mathbf{X}_k \mid \mathbf{X}_{0:k-1}) = p(\mathbf{X}_k \mid \mathbf{X}_{k-1}) \tag{6.1}$$

That is, the current state of the process is conditionally independent of the past states except the most recent one. A Markov chain can be described by a transition matrix T in which:

$$\mathbf{T}_{mn} = p(\mathbf{X}_k = S_n | \mathbf{X}_{k-1} = S_m) \tag{6.2}$$

where S_n and S_m are two of the possible states. Regardless of which initial state the algorithm starts, the chain will always reach a steady state distribution $p(\mathbf{X})$ if $\mathbf{T}$ possesses the following two properties [50]:

1. Irreducibility: A MC is called irreducible (or indecomposable) if for all pairs of states (n, m) there exists an integer n such that $\mathbf{T}^{(l)}_{mn} > 0$. An irreducible MC can not be decomposed into parts which do not interact.

2. Aperiodicity: An irreducible chain is called aperiodic (or acyclic) if the period equals 1 or, equivalently, if for all pairs (m,n) of states there is an integer L_{mn} such that for all $l \geq L_{mn}$, the probability $p^{(l)}_{mn} > 0$. Here $p^{(l)}_{mn} = p(\mathbf{X}_{k+l} = n \mid \mathbf{X}_k = m)$. This property ensures that the chain will not get trapped in cycles.

They guarantee a finite path from every state to every other state with non-zero transition probability, which is the so-called ergodicity property.

Given a defined target distribution, Markov Chain Monte Carlo (MCMC) method takes aim at constructing a MC which has this distribution as its invariant distribution [25]. Normally we ensure the stationarity of the chain by designing it to satisfy the reversibility property (detailed balance):

$$p(\mathbf{X}_k)\mathbf{T}(\mathbf{X}_{k-1} \mid \mathbf{X}_k) = p(\mathbf{X}_{k-1})\mathbf{T}(\mathbf{X}_k \mid \mathbf{X}_{k-1}) \tag{6.3}$$

This is easy to see as long as we sum both sides of the equation above over $\mathbf{X}_{k-1}$:

$$p(\mathbf{X}_k) = \sum_{\mathbf{X}_{k-1}} p(\mathbf{X}_{k-1})\mathbf{T}(\mathbf{X}_k \mid \mathbf{X}_{k-1}) \tag{6.4}$$

The most frequently adopted MCMC method is the Metropolis-Hasting (MH) algorithm proposed by Hasting [51]. According to this algorithm the transition probability is given by:

$$\mathbf{T}(\mathbf{X}_k \mid \mathbf{X}_{k-1}) = \alpha(\mathbf{X}_{k-1}, \mathbf{X}_k) g(\mathbf{X}_k \mid \mathbf{X}_{k-1}) \tag{6.5}$$

where $g(\mathbf{X}_k \mid \mathbf{X}_{k-1})$ is a proposal distribution we can directly sample from and:

$$\alpha(\mathbf{X}_{k-1}, \mathbf{X}_k) = \min(1, \frac{p(\mathbf{X}_k) g(\mathbf{X}_{k-1} \mid \mathbf{X}_k)}{p(\mathbf{X}_{k-1}) g(\mathbf{X}_k \mid \mathbf{X}_{k-1})}) \tag{6.6}$$

is called the acceptance rate. It is easy to verify that Equation 6.6 has the property of reversibility. One condition that ensures the quality of convergence is that $g(\mathbf{X})/p(\mathbf{X}) > 0$ everywhere, so usually $g(\mathbf{X})$ is chosen such that it is similar in shape to $p(\mathbf{X})$, the target distribution. Figure 6.1 shows the results of a one-dimensional M-H algorithm implementation in which the proposal distribution is Gaussian: $N(X_{k-1}, 100)$ and the target distribution $p(X) \propto 0.3e^{-0.2x^2} + 0.7e^{-0.2(x-10)^2}$. In the figure i is the number of iterations. $g(\mathbf{X})$ determines how the state space is exploited. This is especially important to a high-dimensional problem such as human tracking. Gibbs sampling is one of the choices. Given a proposal distribution, Gibbs sampling repeatedly replaces each component of the vector with a value picked from a distribution conditional on the values of all the other components. However, a more general and more popular method is the Metropolis algorithm. In this algorithm, new samples are generated by varying some of the components of the vector by a symmetric random walker sampler, which means that the sampling proposal is determined only by the samples' separation from $\mathbf{X}_{k-1}$:

$$g(\mathbf{X}_k \mid \mathbf{X}_{k-1}) = g(|\mathbf{X}_k - \mathbf{X}_{k-1}|) \tag{6.7}$$

Thus the acceptance rate reduces to:

$$\alpha(\mathbf{X}_{k-1}, \mathbf{X}_k) = \min(1, \frac{p(\mathbf{X}_k)}{p(\mathbf{X}_{k-1})}) \tag{6.8}$$

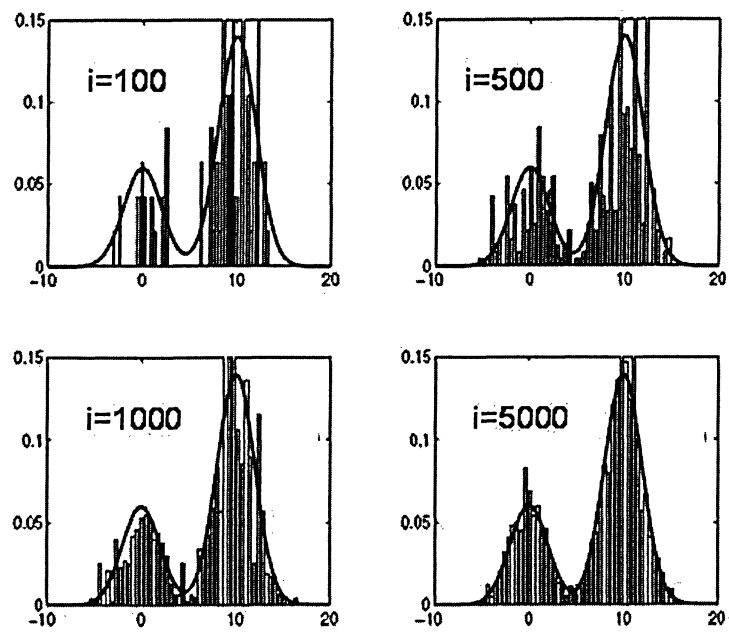


Figure 6.1: The Metropolis-Hasting algorithm implementation for MCMC

The calculations of the MH algorithm and the Metropolis algorithm are especially convenient because although we cannot directly draw samples from the target distribution, we know how to roughly evaluate the weights of them everywhere. This is precisely the case encountered when using the particle filter for visual tracking. Practically the symmetric random walker sampler is often chosen as a multivariate normal distribution $N(0, c^2\Sigma)$. Here Σ is the covariance matrix of the D-dimensional vector $\mathbf{X}$, and c is a scalar whose value is found to be optimal when $c = 2.38/(D)^{1/2}$.

6.2 The Differential Evolution Algorithm

The Differential Evolution Algorithm (DE) is an algorithm dealing with the problem of parallel searching for a global maximum through high dimensional state space [20]. Similar to other evolutionary program methods such as the Genetic Algorithm, it is also based on evolution theory and a competition mechanism. Stronger members of the population more easily survive to the next generation to guarantee that the new generation is better than the last one as a whole. Many people noticed the similarity between evolutionary optimization algorithms and the particle filter, just as Deutscher commented in [19]. Compared with the Genetic Algorithm, the Differential Evolution Algorithm is defined in real parameter spaces instead of binary code parameter spaces. So it is much simpler to implement. The DE algorithm is able to explore non-isotropic structures such as ridges in the target function because the vector differences are usually aligned with the direction of the ridges. Experiments also verify its excellent performance in convergence through comparison with other optimization methods [20].

Assume that a complicated function $f(\mathbf{E})$ is defined over a D-dimensional state space ε. Assume also that we do not know the analytical form of this function (or the analytical form is too complicated for a gradient-based method to apply), but can evaluate the value indirectly. We can use the DE algorithm to search the global maximum with an initial population $\mathbf{E}_{n,0}$, $n = 0, 1, \ldots, N-1$. N is the number of population. The simplest version

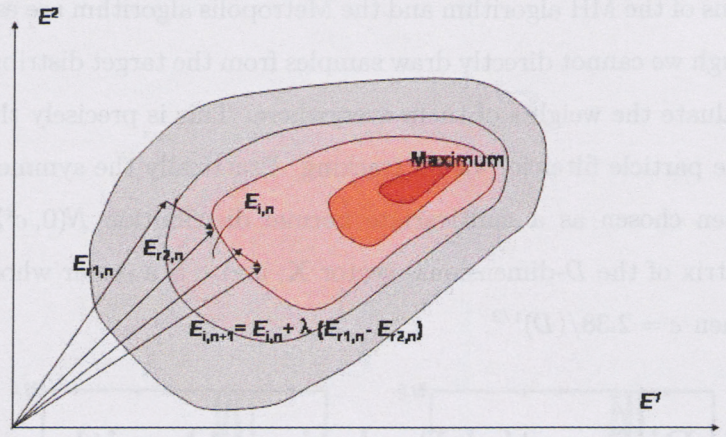


Figure 6.2: The Differential Evolution Algorithm

of the DE algorithm generates a new generation of population in time step k+1 according to:

$$\mathbf{E}^*_{n,k+1} = \mathbf{E}_{n,k} + \lambda(\mathbf{E}_{r1,k} - \mathbf{E}_{r2,k}) \tag{6.9}$$

where $r1$, $r2$ are random integers drawn from [0,1,2,...,N-1] and mutually different, and $\lambda > 0$. Figure 6.2 illustrates how the DE algorithm produces a new vector from the previous generation. In the literature, there are several variations of Equation 6.9 available, which can also be applied to the DE algorithm [20]. A popular one is given below:

$$\mathbf{E}^*_{n,k+1} = \mathbf{E}_{n,k} + \lambda_1(\mathbf{E}_{best,k} - \mathbf{E}_{n,k}) + \lambda_2(\mathbf{E}_{r1,k} - \mathbf{E}_{r2,k}) \tag{6.10}$$

where the $\mathbf{E}_{best,k}$ is the best member in the kth generation.

The use of a crossover operator is then optional for increasing the potential diversity of the perturbed state vectors:

$$E_n = (e_n^0, e_n^1, \ldots, e_n^{\langle l-1\rangle_D}, e_m^{\langle l\rangle_D}, e_m^{\langle l+1\rangle_D}, \ldots, e_m^{\langle l+V-1\rangle_D}, e_n^{\langle l+V\rangle_D}, \ldots, e_n^{D-1}) \tag{6.11}$$

Here e_n^d is the dth element of vector $\mathbf{E}_n$, and $\langle D\rangle$ denotes the modulo function with modulus D. The start position index l for crossover operation is randomly chosen from [0, D-1], and the crossover length V is drawn from [0, D-1] with the probability $p(V = v) = (CR)^v$,

$CR \in [0,1]$ is a control variable. Whether a newly generated state vector will be accepted is solely dependent on the value evaluated by the target function: In a global maximum search, if $\mathbf{E}^*_{i,n+1}$ yields a larger function value than $\mathbf{E}_{i,n}$ the state will be updated, otherwise it will be kept intact.

6.3 The Differential Evolution Markov Chain

By examining the characteristics of the MCMC and the DE algorithm we find that they aid each other in searching for an optimal solution. The acceptance rule in the DE part is controlled by the MCMC acceptance mechanism, whilst the step size and orientation of the random walk of the MCMC part is produced by the DE algorithm. By constructing multiple MCMCs in parallel, the state space is more efficiently explored since the state vectors can be more reasonably distributed than with a single chain. These chains can interact with each other, sharing information with the aid of the DE algorithm. Under the guidance of the DE algorithm, the MCMCs will gradually concentrate on the important regions of the posterior distribution without being easily trapped in local basins. The DE-MC algorithm is summarized as follows:

The DE-MC Algorithm

1 Start with a target function $f(\mathbf{E})$ and an initial population $(\mathbf{E}_{0,0}, \mathbf{E}_{1,0}, \ldots, \mathbf{E}_{N-1,0})$, whose members are D-dimensional vectors.

2 In the kth iteration For each member of the population $\mathbf{E}_{n,k-1}, n = 0, 1, \ldots, N-1$, randomly choose two integers $r1$ and $r2$ so that $r1 \neq r2 \neq n$.

3 Create a new member $\mathbf{E}^*_{n,k}$ by:

$$\mathbf{E}^*_{n,k} = \mathbf{E}_{n,k-1} + \lambda(E_{r1,k-1} - E_{r2,k-1}) + g. \tag{6.12}$$

λ is the same scalar as used in Equation 6.9 and g is drawn from a symmetric distribution with small variance compared to that of $\mathbf{E}$.

4 Compute the ratio:

$$R = \frac{f(\mathbf{E}^*_{n,k})}{f(\mathbf{E}_{n,k-1})}. \tag{6.13}$$

5 Choose a number h from $U(0,1)$, if $R > h$, $\mathbf{E}_{n,k} = \mathbf{E}^*_{n,k}$; otherwise $\mathbf{E}_{n,k} = \mathbf{E}_{n,k-1}$.

6 Repeat steps 2-5 for iteration $k+1$ until a convergence or a preset end point is reached.

Note that to make the DE-MC algorithm match our work, here we are searching for a maximum instead of a minimum.

The combination of the DE algorithm and MCMC appears to be natural and simple. However there are still some issues we should examine. Since the theoretical foundation for MCMC is built upon detailed balance we should avoid violating it. However, not every version of DE can retain detailed balance. Fortunately Equation 6.12 satisfies the detailed balance condition since if $\mathbf{E}^*_{n,k}$ is accepted:

$$\mathbf{E}_{n,k} = \mathbf{E}_{n,k+1} + \lambda(\mathbf{E}_{r2,k} - \mathbf{E}_{r1,k}) - g \tag{6.14}$$

and if it is rejected, the value remains unchanged. In contrast, Equation 6.10 can not be written in such a balance form. Therefore, we choose the original DE version for the DE-MC algorithm.

The same problem occurs when we use the crossover operator in the DE-MC. To ensure the detailed balance, when we pick two members from the population for crossover, we do not only use part of one member to replace that part of the other member, but interchange this part of the two members. Moreover, if one of the new members is accepted, the other new member created by interchanging must also be accepted. Therefore Equation 6.13 becomes:

$$R = \frac{f(\mathbf{E}^*_{m,k})f(\mathbf{E}^*_{n,k})}{f(\mathbf{E}_{m,k-1})f(\mathbf{E}_{n,k-1})}. \tag{6.15}$$

It is easy to verify that detailed balance is retained with the modified crossover operator. However, additional function evaluations are required. Users can make a choice whether to include the crossover operator in DE-MC by considering the overall performance gain. If it is included, it should be between Step 3. and Step 4. of the DE-MC algorithm.

6.4 The DE-MC particle filter

Based on the Differential Evolution-Monte Carlo (DE-MC) particle filter, we propose a novel sequential Monte Carlo sampling approach, namely the DE-MC particle filter. The DE-MC particle filtering iteration at time step k is shown below:

The DE-MC Particle Filter Algorithm

Starting from the set of particles which are the filtering result of time step $k-1$: $\{\mathbf{X}_{k-1}^{(i)}, w(\mathbf{X}_{k-1}^{(i)})\}_{i=1}^{N}$.

1 **Selection:** select a new set of samples $\{\hat{\mathbf{X}_k^{(i)}}\}_{i=1}^{N}$ from $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^{N}$ with the probability proportional to $w(\mathbf{X}_{k-1}^{(i)})$.

2 **Prediction and Measurement:** Apply a constant velocity dynamical model to the samples:

$$\mathbf{X}_k^{(i)-} = \hat{\mathbf{X}_k^{(i)}} + \mathbf{V}_{k-1} \tag{6.16}$$

where $\mathbf{V}_{k-1}$ is the velocity vector computed in time step k-1. The particle set $\{\mathbf{X}_k^{(i)-}\}_{i=1}^{N}$ then acts as the initial population for a T-iteration DE-MC processing. The processing follows the procedure we listed in the previous section. The fitness function is determined as the measurement function we developed in Chapter 5. Hence the weights of particles are subject to update by the DE-MC. For Equation 6.12 in step 3 of the DE-MC algorithm we choose $g \sim U(-c\sigma, c\sigma)$ and $\sigma = [\sigma_0, \sigma_1, \ldots, \sigma_{D-1}]^T$ is a vector with the elements equal to standard deviations for the elements in $\mathbf{X}$. Normal distribution can be used here instead of uniform distribution. c is a small number which can be flexibly chosen. Also in the same equation, the optimal value of λ is determined in literature [9] by

$$\lambda = (1-c) \times \frac{2.38}{\sqrt{D}} \tag{6.17}$$

In our experiments since $D = 24$ the value is around 0.437. At the end of this step, we take the output population as the particle set of current time step: $\{\mathbf{X}_k^{(i)}, w(\mathbf{X}_k^{(i)})\}_{i=1}^{N}$.

3 **Representation and Velocity Updating:** Estimate the state at time step k as:

$$\mathbf{X}_k = \arg\max_{\mathbf{X}_k^{(i)};i=1,\ldots,N} w(\mathbf{X}_k^{(i)}) \tag{6.18}$$

and calculate the velocity vector of current time step:

$$\mathbf{V}_k = \mathbf{X}_k - \mathbf{X}_{k-1} \tag{6.19}$$

We adopt a simple strategy to help the filter adapt to the changes of situations. The method is to calculate the value of σ in step 2 by:

$$\sigma_{k,t} \propto \frac{1}{\sum_{i=1}^{n}(w(\mathbf{X}_{k,t-1}^{(i)}))^2} \tag{6.20}$$

where t denotes the DE-MC iteration index. This strategy comes naturally from Equation 3.28. The step size of random jumping for current DE-MC iteration is reduced if the survival rate of the last DE-MC iteration is high and is increased the other way round.

The most evident improvement of the DE-MC particle filter with respect to the CONDENSATION algorithm is that the prediction (sampling) step and the measurement step are now integrated together instead of functioning separately. Please be reminded that at the beginning of this chapter we pointed out that generic particle filters simplify the distribution $g(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_k) = p(\mathbf{X}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)})$. Obviously, We lose the information about current observation during this simplification, which causes serious distortion in the sampling from posterior distribution. To be more specific, when we begin the measurement process, the samples are already drawn. If they are already trapped in the local cost basin of the state space, which frequently occurs in the human motion tracking applications, there is no way for them to escape. Errors are then accumulated and things get worse, until a total tracking failure takes place. However, with the DE-MC particle filter we are able to make a more reasonable sampling. The dynamical model is still necessary to accomplish part of the sampling task, as we can see from the DE-MC particle filtering step. However it is the DE-MC algorithm that really makes the proposed algorithm work successfully. In the DE-MC iterations, the measurement module provides necessary feedbacks to the sampling

module, and according to them the sampling moves to regions in the state space where it is more possible for the global maximum of the measurement function is to be found. Note that the generated samples are not necessarily in strict accordance with the ground-truth posterior distribution. Since we are interested in the global optimal state and only have limited number of samples, we place denser sampling grids in the region of interest. For the purpose of global optimization, this approach yields a result reasonably close to that obtained by sampling strictly according to the ground-truth posterior distribution, while at the same time it saves considerable computation cost.

Chapter 7

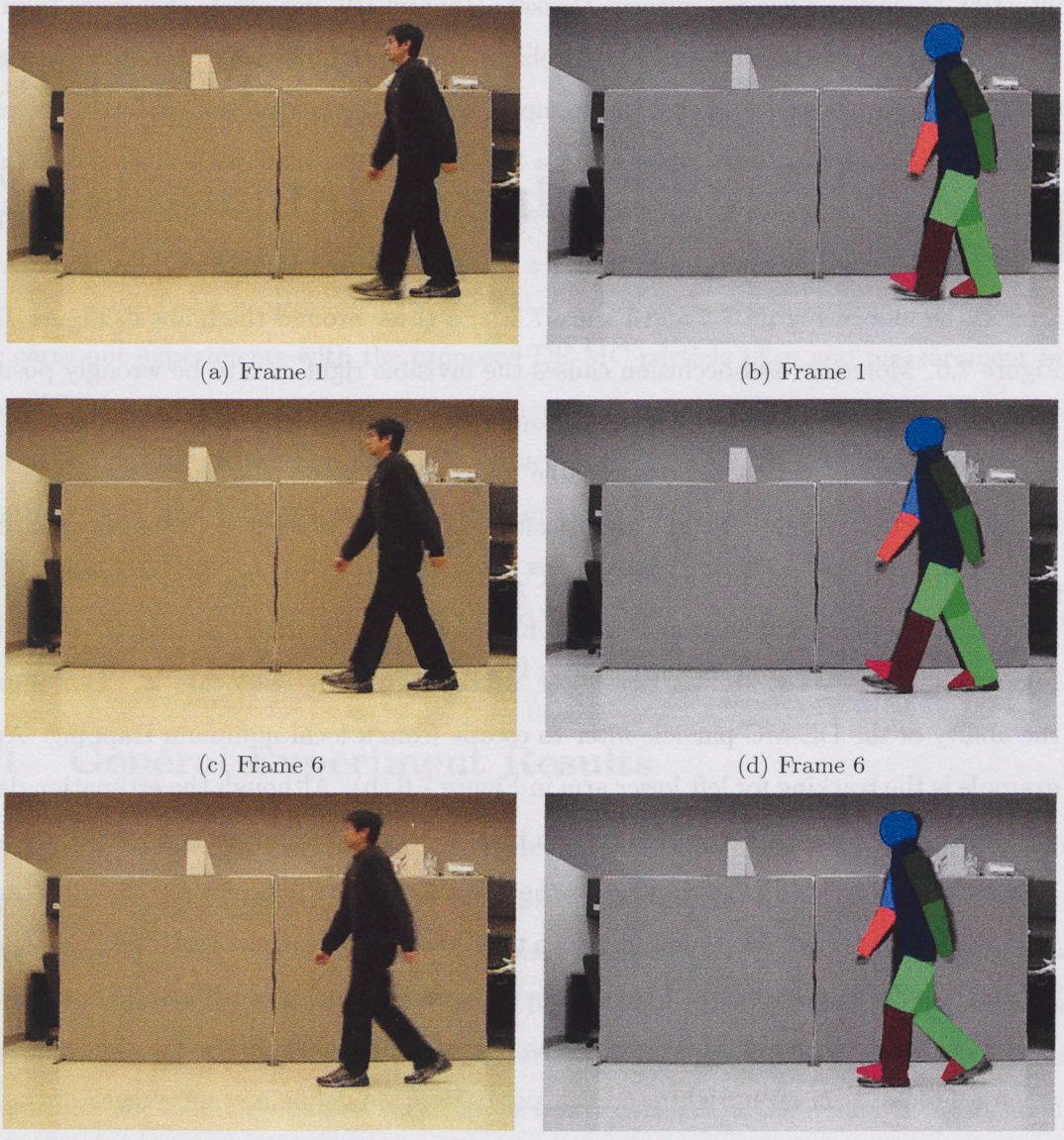
Experimental Results

We carry out experiments with the proposed DE-MC particle filter and measurement function. We use two monocular human motion video sequences: Most experiments are conducted on Sequence 1, which is a walking sequence; Sequence 2 is a hopping sequence. Their length are 1.8s and 1.2s, respectively. Both of them are in side view. The human subject wears loose fit clothings. The shooting environment has been introduced in Chapter 5. We will show the general tracking results first and then some comparison results concerning the performance of the DE-MC particle filter and proposed measurement function.

7.1 General Experiment Results

Figure 7.1 - 7.6 show part of the tracking results for the 2 monocular sequences. For tracking Sequence 1, a 7-layer DE-MC particle filter is used (Here we use "layer" instead of "iteration" so as to avoid being confused with the tracking iteration). The number of particles is 500, which can lead to a satisfactory balance between the reliability and the computational cost of the tracker. For Sequence 2, a 9-layer DE-MC particle filter with 600 particles is used. Since the side-view causes very serious self-occlusions in the scene, these monocular image sequences offer a huge challenge for any human motion tracking algorithm. Note that the uniform-color fitting in Sequence 1 provides additional difficulty since in many previous research works, clothing with varying texture is often utilized to label the body segments. To our knowledge there are very few successful tracking results reported under a

similar situation in the literature. [31], [26] and many other similar works tackle monocular video sequence-based human tracking by learning dynamical models, which departs from our intention to design a general-purpose tracker. [15] and [19] use image sequences from three views, which remove the self-occlusion problem and depth ambiguity to a large extent. Our tracking results are not perfect. For example: Due to the fast motion and the existence of much depth ambiguity, even when using a DE-MC particle filter with more computational power to track Sequence 2, the tracking result is still much worse than that of Sequence 1; We can find the tracking errors that are obviously caused by motion blur, such as those around the foot in Figure 7.2 and Figure 7.3, and those around the limbs in Figure 7.5 and Figure 7.6. Moreover, self-occlusion causes the invisible right arm to be wrongly positioned in Figure 7.2(f), Figure 7.3(b) and (f), and Figure 7.4(f). However, we should be aware that most of these errors are unavoidable for tracking monocular video sequences. Even a human eye can rarely tell the exact position of a limb when it is occluded or in motion blur. Even though, the proposed DE-MC particle filter still achieves an excellent overall performance. For instance, when the occluded arm in Figure 7.3(d) and (f) reappears in Figure 7.4 (b), the DE-MC particle filter is able to quickly reallocate it. This illustrates the ability of the DE-MC particle filter to escape from a local minimum trapping. Another example is the tracking for left lower arm in Figure 7.6 (b). Although the estimation deviates from arm's real position because of motion blur, it becomes accurate again when the motion blur alleviates in 7.6(d). According to the tracking results, we plot the shoulder and hip joint angles for the rotation around x-axis together with the elbow and knee joint angles in Figure 7.7. To help understand the values of these joint angles, in Figure 7.8 we illustrate how they change with limb movement. The reference positions are the locations of the limbs when a human is in an upright standing pose. Here we define any movement of the upper arms or legs in the direction that the human subject is facing as movement in the forward direction. α and β are positive.

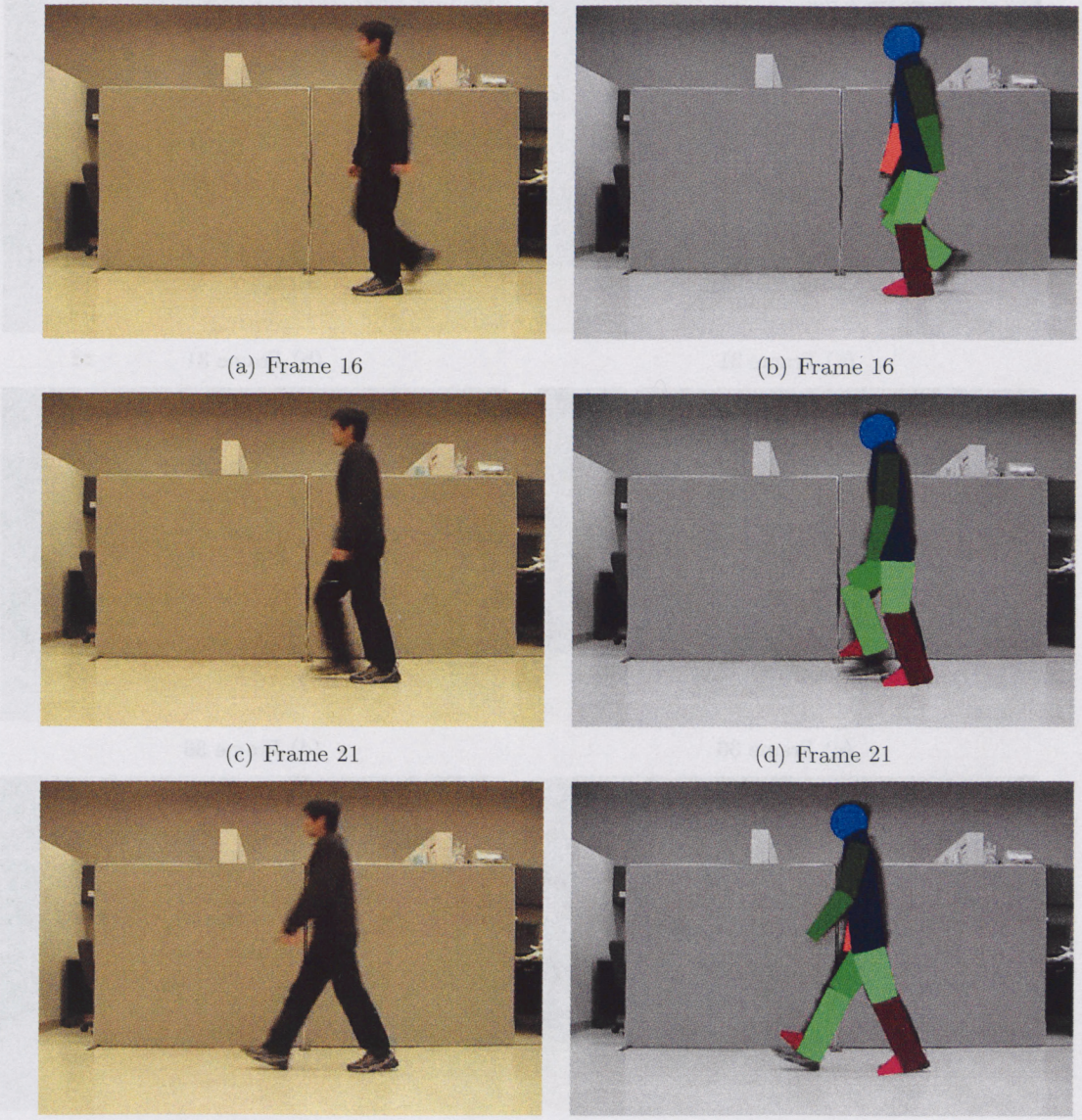


(a) Frame 1 (b) Frame 1

(c) Frame 6 (d) Frame 6

(e) Frame 11 (f) Frame 11

Figure 7.1: General experiment: tracking result for Sequence 1(I).

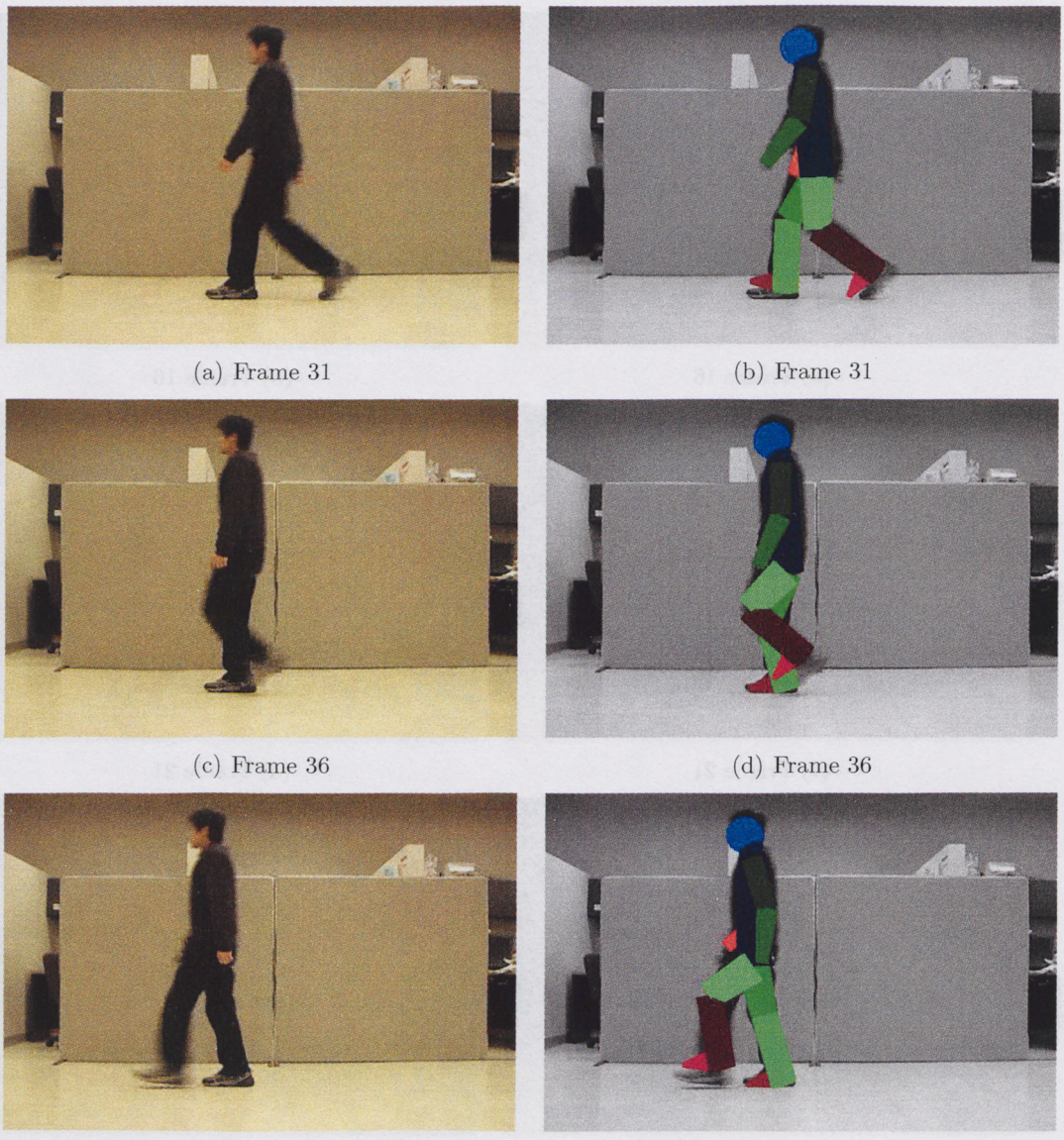


(a) Frame 16 (b) Frame 16

(c) Frame 21 (d) Frame 21

(e) Frame 26 (f) Frame 26

Figure 7.2: General experiment: tracking result for Sequence 1(II).



(a) Frame 31

(b) Frame 31

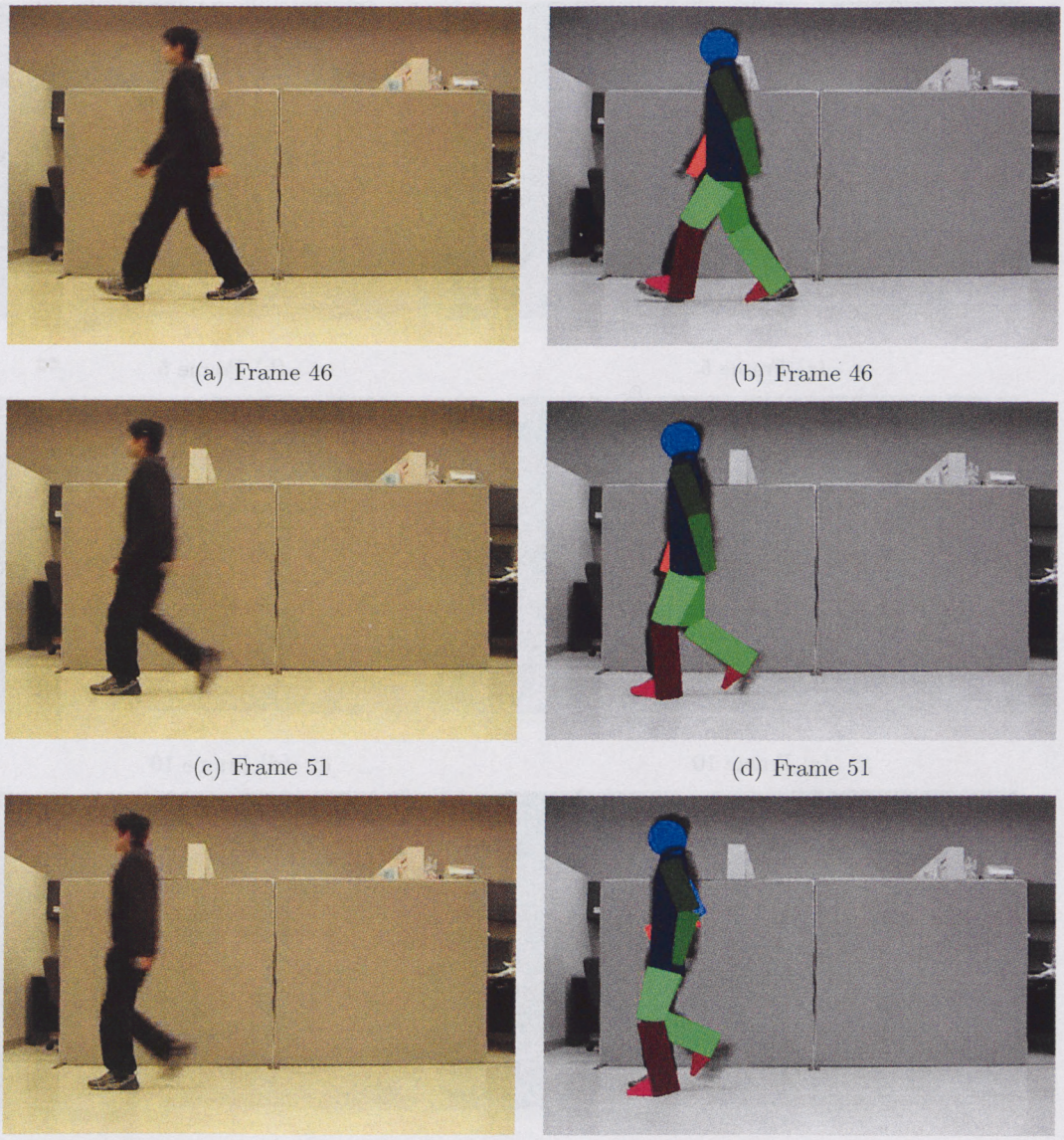
(c) Frame 36

(d) Frame 36

(e) Frame 41

(f) Frame 41

Figure 7.3: General experiment: tracking result for Sequence 1(III).

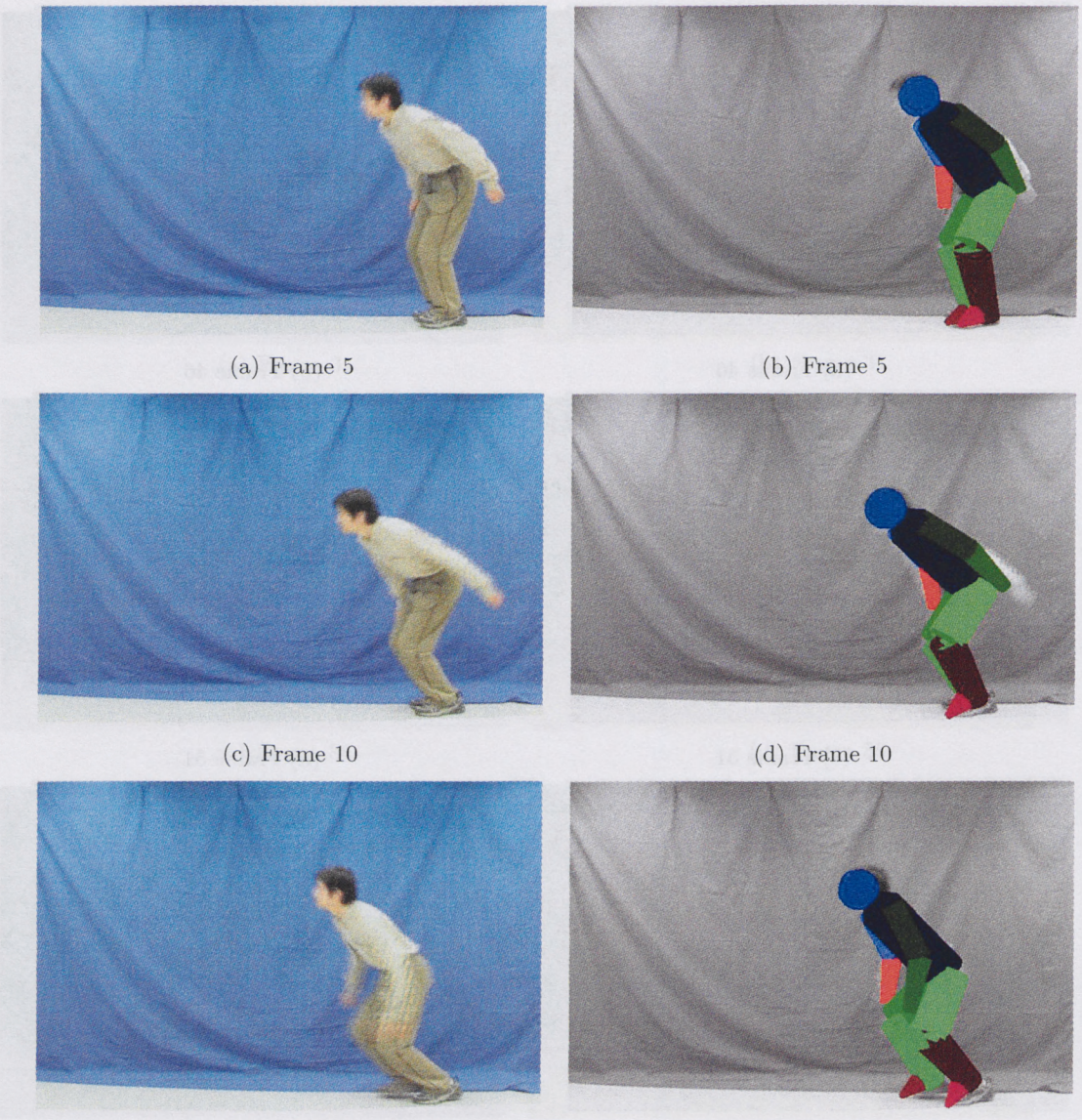


(a) Frame 46 (b) Frame 46

(c) Frame 51 (d) Frame 51

(e) Frame 56 (f) Frame 56

Figure 7.4: General experiment: tracking result for Sequence 1(IV).



(a) Frame 5

(b) Frame 5

(c) Frame 10

(d) Frame 10

(e) Frame 15

(f) Frame 15

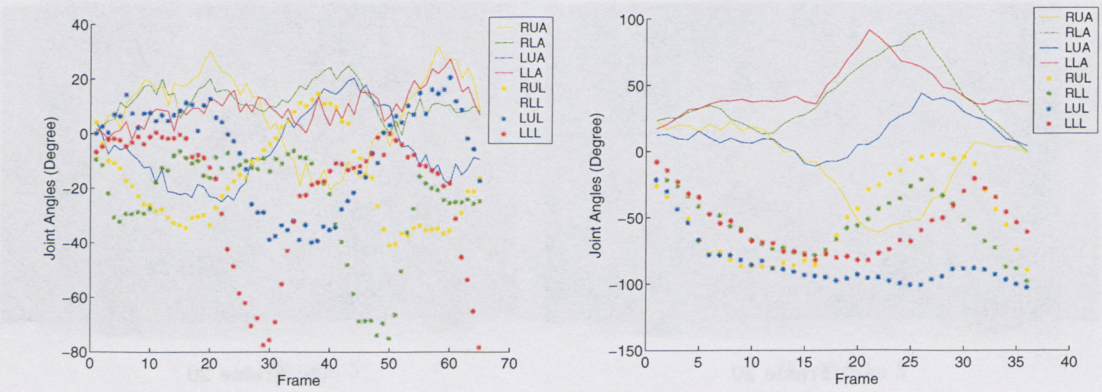
Figure 7.5: General experiment: tracking result for Sequence 2(I).

(a) Frame 20 (b) Frame 20

(c) Frame 25 (d) Frame 25

(e) Frame 30 (f) Frame 30

Figure 7.6: General experiment: tracking result for Sequence 2(II).



(a) Walking sequence (b) Hopping sequence

Figure 7.7: Estimated joint angles for the two sequences

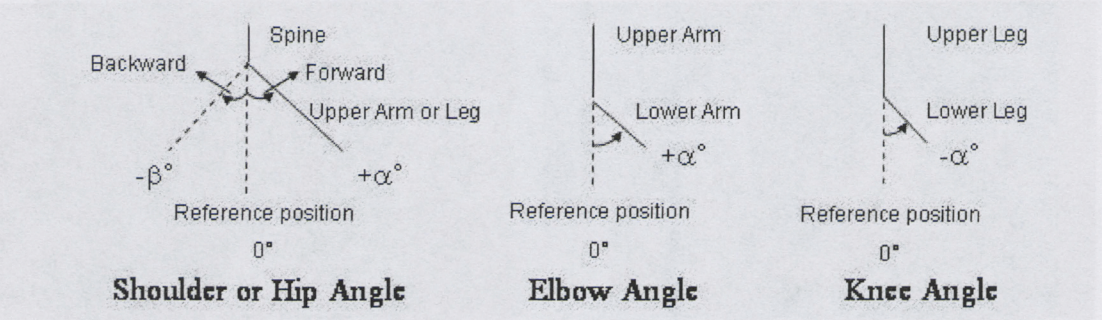


Figure 7.8: Reference joint angles and directions

Algorithm	Particle Number	Layer Number	Measurement Function Evaluations
CONDENSATION	5000	N/A	5000
Annealed Particle Filter	500	10	5000
DE-MC Particle Filter	500	7	3500

Table 7.1: Computational cost for different algorithm used in Comparison Experiment 1

7.2 Comparison Experiment Results

Comparison Experiment 1

In Figure 7.9, we compare the tracking result obtained by a 7-layer DE-MC particle filter with those obtained by other popular particle filtering-based algorithms. In this figure, the images at each row are (from the top to the bottom): the original video sequence, the results obtained by using the CONDENSATION algorithm, the results obtained by using the Annealed Particle Filter and the results obtained by using the DE-MC particle filter, respectively. In terms of fair comparison, the experiment is based on almost the same number of measurement function evaluations because it is the most time-consuming part for particle filtering. The computational expense for each algorithm is listed in Table 7.1. The other factors, such as the initialization result, initial standard deviation of the state vector, constant velocity model, adaptive strategy and measurement function are all given the same settings for each algorithm. As we can see, although consuming only 70% of the computations spent by the other two algorithms, the DE-MC particle filter still shows its superiority. It locks every human body part with relatively high accuracy. The Annealed Particle Filter, which is based on the Simulating Annealing algorithm, is shown to be able to successfully track human walking from multi-view video sequences in [15]. But in our monocular experiment, it just roughly captures the global location of the human subject but rarely makes invalid estimates to the motion of limbs. The classic CONDENSATION algorithm cannot even predict the global position of the human accurately. This experiment strongly demonstrate the power of our proposed approach.

Comparison Experiment 2

Figure 7.9: Comparison Experiment 1: comparison of the performance of the DE-MC particle filter with different tracking algorithms.

In this experiment we compare the tracking results for Sequence 1 obtained by using 2-layer, 5-layer and 7-layer DE-MC particle filters, as shown in Figure 7.10 from the left to the right, respectively. We emphasize that the evident difference shown here is a result of accumulated difference for the three different particle filters up to the 36th frame. This should not be confused with the intermediate output of a 7-layer DE-MC particle filter obtained after the computation in the second, fifth and seventh layer. In the latter case, the difference is much smaller.

Figure 7.10: Comparison Experiment 2: comparison of the performance of the DE-MC particle filters with different layer number.

Comparison Experiment 3

In this experiment we compare the efficacy of different image cues. A 7-layer, 500-particle DE-MC particle filter is used. We compare the tracking results for Sequence 1 using particle filters based on 4 different measurement functions in Figure 7.11. The images at each row are (from the top to the bottom): the original video sequence, the results obtained by using the multi-cue fusion-based measurement function, the results obtained by using the color cue-based measurement function, the results obtained by using the silhouette area cue-based measurement function, and results obtained by using the boundary cue-based measurement function, respectively. From this figure we can see that superiority of multi-information fusion is apparent. It also demonstrates the power of individual cues: Color cue-based tracker yields relatively large error in locating the position of leg, mainly because color measurement function lacks the ability in describing the geometrical property of the objects. It tries to find the most possible region only in the sense of best color histogram fit, no matter

where this region is or whether this region actually corresponds to a body part. Ambiguity often occurs when the two legs of the subject are overlapped, since from then on there exists many regions with similar color histograms. Even after the legs depart from each other, the already updated histogram makes it very difficult to relocate them. On the contrary, the silhouette area information-based tracker and the boundary information-based tracker seldom produce geometrical fit error. What confuses them is the left-right ambiguity. This is especially evident in the third result image for the silhouette area information- based tracker and the fourth result image for the boundary information-based tracker. They appear to provide almost perfect geometrical fit, but through careful examination, we can find that they totally reverse the position of left and right arms. This is reasonable since both boundary and silhouette are unable to distinguish individual body parts. Instead, they treat the human body as a whole. On the other hand, the color information is able to mark each body segment and update their histograms individually. This experiment exactly demonstrates the necessity of fusing multiple image features for tracking.

Comparison Experiment 4

Figure 7.12 shows the results of tracking Sequence 1 with a DE-MC particle filter before and after adopting the proposed relative weight strategy for calculating Bhattacharyya distance of the two histograms as formulated in Equation 5.5. From the comparison we can conclude that the performance gain by adopting our method is significant and evident. Note that all the results in this experiment are based on the fused measurement function rather than the color-cue-alone measurement function.

Figure 7.11: Comparison Experiment 3: comparison of the performance of different measurement function

Figure 7.12: Comparison Experiment 4: result before (top) and after (bottom) adopting relative weights for color-cue based measurement function

Chapter 8

Conclusions

8.1 Summary

Tracking human motion from monocular video sequence is a difficult task. For years researchers have been committing a lot of efforts, trying to solve the problem. However, only limited progress has been made. In this thesis, we propose a novel approach for human motion tracking from monocular sequences. This approach is mainly based on the Differential Evolution algorithm, the Monte Carlo Markov Chain theory and the particle filtering, so we name it as DE-MC particle filtering. The most noticeable characteristic of the proposed method is its ability to incorporate the sampling and the measurement process which are separately implemented in most previous visual tracking work. We develop such a method based on the accumulated experience that in a human tracking problem, which can be formulated by Bayesian inference, the posterior depends on both the previous system state and the current observation. The advantage of combining sampling and measurement is that it will lead to a more reasonable approximation to the ground-truth posterior. Moreover, the power of the DE algorithm and the MCMC allows us to save a large amount of computations when simulating the posterior distribution, because their interactive combination has the property of "importance sampling". In term of searching for global optimal pose configuration in a high-dimensional state vector space, the DE-MC particle filter achieves very good balance between exploration and exploitation.

We also notice that human tracking is a complicated project which requires good coop-

eration from different aspects. We use a simple but effective articulated 3D human body model in our work. It can model most of the human poses without difficulty. We introduce an adaptive strategy for the random jumping of the DE-MC iterations as a fine adjustment method. Also, we design a robust measurement function which fuses multiple image cues. Among them, the boundary cue represented by FD to facilitate tracking is a novel approach. It has the merits of easy extraction and is computationally economical. Besides, the color cue representation is improved by adopting a weighted Bhattacharyya distance calculation, which is specially tailored for human tracking applications. The fusion of different image information yields a peaky measurement function, which contributes greatly to the stable performance of the DE-MC particle filter.

Both the general experiment results and the comparison experiment results suggest the validity of our approach. Similar or better results for tracking a full-body human motion are usually seen in the work which adopts trained dynamical models or which is based on videos captured by multiple synchronized cameras.

8.2 Open Issues

There is much work yet to be addressed within the field of human motion tracking. We expect more technological breakthroughs to be made in the near future. Below we outline several important research issues related to our work:

There is still substantial space for the sampling and searching strategy to be further improved. There are a lot of optimization and sampling algorithms available. Some of them may be more suitable for solving the human tracking problem. For example, Gibbs sampling is extensively used for multivariate distribution sampling. It is a time-consuming procedure when vectors are in high dimension, but with some modifications it may be incorporated into the DE-MC framework. The constant velocity model also accomplishes part of the sampling work. We may adopt an autoregressive model instead. It can help the particle filter to adapt to abrupt human pose change but will not bring too much additional burden since the

update only happens once at the end of each time step.

Finding a more reasonable fusion method for different image cues is an urgent issue. Currently they are simply multiplied and their individual characteristics are ignored. It will be greatly helpful if an online adaptive strategy for fusion is developed. We hope it can automatically decide the timing for changing the relative importance of different image cues according to their most suitable working environment. Some tries to realize this strategy by comparing the weights evaluated by individual measurement functions. It is our opinion that this may not be an effective method, since the measurement function itself is still to be evaluated and the weights generated by them cannot be used as a criterion.

Our human body model leads to acceptable accuracy of approximation to the real human body, but further improvement of the model will bring additional gains in the tracker's performance. For example, the thickness of human torso is often smaller than the width. It is thus impossible to obtain good hypothesis observations with the initialized truncated cones when the subject turns around his body. We can handle the problem in two directions in the future: 1. To use more complicated geometrical solids such as elliptical cones for modelling. The price is more parameters for determination in the initialization stage. 2. To make the body model reconfigurable. Therefore when the currently used model is no longer able to provide satisfactory geometrical approximation, the parameters of the model can be reinitialized. This is an attractive solution with great challenge: we must develop a reliable criterion to judge that whether the errors of geometrical fit of the model come from the inaccurate modelling (the case in which we should implement the reconfiguration) or from the tracking failure (the case in which we should not implement the reconfiguration). Moreover, it is obvious that this method will not be available until an acceptable automatic initialization algorithm emerges first.

At this point, we have no access to the equipments such as joint markers. Therefore, in our experiments no ground-truth data are available for us to carry out objective and quantitative analysis of the tracking results. It will be beneficial to all the researchers in this field if a universal human motion tracking video database is built. Camera calibration

results and ground-truth motion vectors should be provided as well as the videos. Moreover, the database should provide facilities so that image processing methods developed might be applied to it. For example, it should be accompanied with background images to ease silhouette extraction.

Another field that we should pay more attention to is the exploitation of prior knowledge about human kinematics. By far the furthest step taken in this direction is the application of joint limits. (Here we do not take account of training human motion models from a motion capture database since they are not helpful to tracking many general human motions, or may even cause negative effects.) In our belief there should be much more useful physical constraints which can be applied to facilitate human motion tracking. Researchers from computer vision society have been fighting with such difficulties alone for a long time. It is the time for more participation from experts in biology, athletics, performing arts, biomechanichal engineering and other related fields.

Bibliography

[1] R. C. Gonzalez, and R. E. Woods, *Digital Image Processing, 2nd Edition.* Prentice Hall, 2002.

[2] http://inventors.about.com/od/mstartinventors/a/Muybridge.htm

[3] G. Johansson, "Visual motion perception," *Scientific American*, vol. 6, pp. 76-88, 1975.

[4] N. Goddard, *The Perception of Articulated Motion: Recognizing Moving Light Displays*, Ph.D. Thesis, University of Rochester 1992.

[5] D. Foyrsyth, "Tracking with Non-Linear Dynamic Models," http://www.cs.berkley.edu/ daf/bookpages/pdf/particles.pdf, University of California, Berkley.

[6] I. Haritaoglu, D. Harwood and L. S. Davis, "Ghost: A Human Body Part Labeling System using Silhouettes," *Proceedings of International Conference on Pattern Recognition*, vol. 1, pp. 77-82, Brisbane, Australia, August, 1998.

[7] H. Moon, R. Chellappa and A. Rosenfeld, "Tracking of Human Activities using Shape-Encoded Particle Propogation," *Proceedings of International Conference on Image Processing*, vol. 1, pp. 357-360, Thessaloniki, Greece, Octobor, 2001.

[8] J. Gao, A. G. Hauptmann and H. Wactlar, "Combining Motion Segmentation with Tracking for Activity Analysis," *Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition*, vol. 1, pp. 699-704, Seoul, Korea, May, 2004.

[9] C. J. F. T. Braak, "Genetic Algorithms and Markov Chain Monte Carlo: Differential Evolution Markov Chain Makes Bayesian Computing Easy," http://www.biometris.nl/Markov%20Chain.pdf.

[10] M. K. Leung and Y. H. Yang, "A Model Based Approach to Labeling Human Body Outlines," *Proceedings of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects*, vol. 1, pp. 57-62, Austin, USA, November, 1994.

[11] P. Perez, J. Vermaak and A. Blake "Data Fusion for Visual Tracking with Particles," *Proceedings of the IEEE*, vol. 92, pp. 495-513, March, 2004.

[12] T. J. Roberts, S. J. McKenna and I. W. Ricketts, "Online Appearance Learning for 3D Articulated Human Tracking," *Proceedings of Interntional Conference on Pattern Recognition*, vol. 1, pp. 425-428, Quebec City, Canada, August, 2002.

[13] M. Yamamoto, A. Sato, S. Kawada, T. Kondo and Y. Osaki, "Incremental Tracking of Human Actions from Multiple Views," *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 2-7, Santa Barnara, USA, June, 1998.

[14] H. Sidenbladh and M. J. Black, "Learning Image Statistics for Bayesian Tracking," *Proceedings of Interntional Conference on Computer Vision*, vol. 2, pp. 709-716, Vancouver, Canada, July, 2001.

[15] J. Deutscher, A. Blake and I. Reid, "Articulated Body Motion Capture by Annealed Particle Filtering," *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 126-133, Hilton Head Island, USA, June, 2000.

[16] M. Isard and A. Blake, "ICondensation: Unifying Low-Level and High-Level Tracking in A Stochastic Framework," *Proceedings of European Conference on Computer Vision*, vol. 1, pp. 893-908, Freiburg, Germany, June, 1998.

[17] M. Isard and A. Blake, "Condensation: Conditional Density Propogation for Visual Tracking," *International Journal of Computer Vision*, vol. 29, pp. 5-28, January, 1998.

[18] C. Sminchisescu and B. Triggs, "Hyperdynamics Importance Sampling," *Proceedings of European Conference on Computer Vision*, vol. 1, pp. 769-783, Copenhagen, Denmark, June, 2002.

[19] J. Deutscher, A. Blake and I. Reid, "Automatic Partitioning of High Dimensional Search Spaces Associated with Articulated Body Motion Capture," *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 669-676, Hawaii, USA, December, 2001.

[20] R. Storn, "On the Usage of Differential Evolution for Function Optimization," *Proceedings of Biennial Conference of the North American Fuzzy Information Processing Society*, pp. 519-523, Berkeley, USA, June, 1996.

[21] H. Sidenbladh and M. J. Black, "Stochastic Tracking of 3D Human Figures Uusing 2D Image Motion," *Proceedings of European Conference on Computer Vision*, vol. 2, pp. 702-718, Dublin, Ireland, June, 2000.

[22] M. W. Lee, I. Cohen and S. K. Jung, "Particle Filter with Analytical Inference for Human Body Tracking," *Proceedings of IEEE Workshop on Motion and Video Computing*, pp. 159-165, Los Angels, USA, December, 2002.

[23] Y. Rui and Y. Chen, "Better Proposal Distributions: Object Tracking using Unscented Particle Filter," *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 786-793, Hawaii, USA, December, 2001.

[24] D. Ramanan and D. Forsyth, "Finding and Tracking People from the Bottom-Up," *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 467-474, Madison, USA, June, 2003.

[25] C. Andrieu, N. D. Freitas A. Doucet and M. I. Jorden, "An introduction to MCMC for Machine Learning," *Machine Learning*, vol. 50, pp. 5-43, 2002.

[26] R. Urtasun and P. Fua, "3D Human Body Tracking using Deterministic Temporal Motion Models," *Proceedings of European Conference on Computer Vision*, vol. 3, pp. 92-106, Prague, Czech Republic, May, 2004.

[27] X. Lan and D. P. Huttenlocher, "A Unified Spatio-Temporal Articulated Model for Tracking," *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 722-729, Washington, USA, June, 2004.

[28] D. L. Guidry and A. A. Farag, "Using Active Contours and Fourier Descriptors for Motion Tracking with Applications in MRI," *Proceedings of Interntional Conference on Image Processing*, vol. 2, pp. 177-181, Kobe, Japan, October, 1999.

[29] Q. Delamarre and O. Faugeras, "3D Articulated Models and Multi-View Tracking with Silhouettes," *Proceedings of Interntional Conference on Computer Vision*, vol. 2, pp. 716-721, Corfu, Greece, September, 1999.

[30] K. Rohr, "Incremental Recognition of Pedestrians from Image Sequence," *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, pp. 8-13, New York, USA, September, 1993.

[31] H. Sidenbladh, M. J. Black and L. Sigal, "Implicit Probabilistic Models of Human Motion for Syhthesis and Tracking," *Proceedings of European Conference on Computer Vision*, vol. 1, pp. 784-800, Copenhagen, Denmark, May, 2002.

[32] S. Wachter and H. H. Nagel, "Tracking of Persons in Monocular Image Sequences," *Computer Vision and Image Understanding*, vol. 74, pp. 174-192, June, 1999.

[33] A. Senior, "Real-Time Articulated Human Body Tracking using Silhouette Information," *Proceedings of IEEE Workshop on Visual Surveillance/PETS*, Nice, France, October, 2003.

[34] R. Green and L. Guan, "Quantifying and Recognizing Human Movement Patterns from Monocular Video Images - Part I: A New Framework for Modeling Human Motion.," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, pp. 179-190, February, 2004.

[35] C. L. Huang and C. Y. Chung, "A Real-Time Model-Based Human Motion Analysis System," *Proceedings of IEEE. conference on Multimedia and Expo*, vol. 2, pp. 477-480, Baltimore, USA, July, 2003.

[36] L. Sigal, S. Bhatia, S. Roth, M. J. Black and M. Isard, "Tracking Loose-Limbed People," *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 421-428, Washington, USA, June, 2004.

[37] C. Hu, Q. Yu, Y. Li and S. Ma, "Extraction of Parametric Human Model for Posture Recognition using Genetic Algorithm," *Proceedings of IEEE International Conference on Automatic Face Recognition* , pp. 518-523, Grenoble, France, March, 2000.

[38] J. J. Gonzalez, I. S. Lim, P. Fua and D. Thalmann, "Robust Tracking and Segmentation of Human Motion in An Image Sequence," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* , vol. 3, pp. 29-32, Hong Kong, China, April, 2003.

[39] L. Sigal, M. Isard, B. H. Sigelman, and M. J. Black, "Attractive People: Assembling Loose-Limbed Model using Non-parametric Belief Propagation," *Proceedings Of Conference on Advances in Neural Information Processing Systems 16*, Vancouver, Canada, December, 2003.

[40] C. Sminchisescu and B. Triggs, "Covariance Scaled Sampling for Monocular 3D Body Tracking," *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 8-14, Hawaii, USA, December, 2001.

[41] T. Zhao and R. Nevatia, "Tracking Multiple Humans in Complex Situations," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 26, pp. 1208-1221, September, 2001.

[42] S. Roth, L. Sigal and M. J. Black, "Gibbs Likelihoods for Bayesian Tracking," *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886-893, Washington, USA, June, 2004.

[43] J. Triesch and C. Malsburg, "Democratic Integration: Self-organized Integration of Adaptive Cues," *Neural Computation*, vol. 13, pp. 2049-2074, September, 2001.

[44] G. Taylor and L. Kleeman, "Fusion of Multimodal Visual Cues for Model-Based Object Tracking," *Proceedings of Australasian Conference on Robotics and Automation*, Brisbane, Australia, December, 2003.

[45] J. MacComick and A. Blake, "Partitioned Sampling, Articulated Objects and Interface-Quality Hand Tracking" *Proceedings of European Conference on Computer Vision* , vol. 2, pp. 3-19, Dublin, Ireland, June, 2000.

[46] J. Heikkila and O. Silven, "A Four-Step Camera Calibration Procedure with Implicit Imagecorrection" *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition* , pp. 1106-1112, San Juan, Puerto Rico, June, 1997.

[47] M. Du and L. Guan, "Human Recognition by Body Shape Features" *Proceedings of IS&T/SPIE Symposium on Electronic Imaging: Human Vision and Electronic Imaging X* , pp. 535-545, San Jose, USA, January, 2005.

[48] Y. Huang and T. S. Huang, "Model-Based Human Body Tracking" *Proceedings of Interntional Conference on Pattern Recognition*, vol. 1, pp. 552-555, Quebec City, Canada, August, 2002.

[49] L. Lee and W. E. L. Grimson, "Gait Analysis for Recognition and Classification" *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 148-155, Washington, USA, May, 2002.

[50] S. K. Zhou, R. Chellappa and B. Moghaddam, "Visual Tracking and Recognition using Appearance-Adaptive Models in Particle Filters" *IEEE Transactions on Image Processing*, vol. 13, pp. 1491-1506, November, 2004.

[51] S. Kagami, K. OKADA, M. Inaba and H. Inoue, "Real-Time 3D Depth Flow Generation and its Application to Track to Walking Human Being" *Proceedings of IEEE International Conference on Pattern Recognition*, vol. 4, pp. 197-200, Barcelona, Spain, September, 2000.

[52] Z. Duric, F. Li, Y. Sun and H. Wechsler, "Using Normal Flow for Detection and Tracking of Limbs in Color Images" *Proceedings of Interntional Conference on Pattern Recognition*, vol. 4, pp. 268-271, Quebec City, Canada, August, 2002.

[53] S. Ioffe and D. Forsyth, "Human Tracking with Mixtures of Trees" *Proceedings of Interntional Conference on Computer Vision*, vol. 1, pp. 690-695, Vancouver, Canada, July, 2001.

[54] N. Kranhnstoever and R. Sharma, "Articulated Models from Video" *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 894-901, Washington, USA, June, 2004.

[55] H. Ramoser, T. Schlögl, C. Beleznai, M. Winter and H. Bischof, "Shape-Based Detection of Humans for Video Surveillance Applications" *Proceedings of Interntional Conference on Image Processing*, vol. 3, pp. 1013-1016, Barcelona, Spain, September, 2003.

[56] M. Oren, C. Papageorgiou, P. Sinha and E. Osuna, "Pedestrian Detection using Wavelet Templates" *Proceedings of Interntional Conference on Computer Vision and Pattern Recognition*, pp. 193-199, San Juan, Puerto Rico, June, 1997.

[57] T. Poggio and K. K. Sung, "Finding Human Faces with a Gaussian Mixture Model Distribution-Based Face Model" *Proceedings of Asian Conference on Computer Vision*, pp. 437-446, Singapore, December, 1995.

[58] H. A. Rowley, S. Baluja and T. Kanade, "Neural Network-Based Face Detection" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 23-38, January, 1998.

[59] A. M. Tekalp, *Digital Video Processing.* Prentice Hall, 1995.

[60] A. Doucet, S. Godsill and C. Anderieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering" *Statistics and Computing*, vol. 10, no. 3, pp. 197-208, July, 2000.

Appendix A

List of Publications

- Ming Du and Ling Guan, "Human Motion Tracking with the DE-MC Particle Filter", submitted to International Conference on Computer Vision, March, 2005.
- Ming Du and Ling Guan, "Multi-Information Fusion for Human Motion Tracking by Particle Filter", to appear oin Visual Communications and Image Processing, Beijing, China, July, 2005.
- Ming Du and Ling Guan, "Human Recognition by Body Shape Features", Proceedings of IS&T/SPIE Symposium on Electronic Imaging: Human Vision and Electronic Imaging X, San Jose, USA, January, 2005.