Human posture detection for video surveillance

Jin, Tao

2009

Jin, T. (2009). Human posture detection for video surveillance. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/15285>

<https://doi.org/10.32657/10356/15285>

Human Posture Detection for Video Surveillance

JIN TAO

SCHOOL OF COMPUTER ENGINEERING

2009

Human Posture Detection for Video Surveillance

Jin Tao

School of Computer Engineering

A thesis submitted to Nanyang Technological
University in fulfillment of the requirement for the degree of
Doctor of Philosophy

2009

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Dr. Maylor Karhang Leung. It has been an honor to be his Ph.D. student. I appreciate for his enthusiasm and encouragement that assisted me to pass through tough times. I am also thankful for his insights, suggestions and patience that helped me to shape and consolidate my research skills. Furthermore his valuable instruction and feedback enable me to complete this dissertation. His refusal to accept any lowering of standards will let me benefit a lot for the rest of my life.

The Center for Advanced Media Technology (*CAMT*) provided space and equipment to work with, and the staff at *CAMT* gave me kindness technical support and comfortable work environment. My friends, X. Chen, Q. Liu, C. Chen, X. Zhang, Y. Liu, J. Qiu, W. Ng, Y. Cao, Y. Chi, T. Xiang, J. Lv, L. Li, helped to lighten the atmosphere at work and beyond. To all, my heartfelt thanks. In the process of data collection and survey of human behavior recognition for this research, many people contributed to these task and I am grateful for their contributions. I would particularly like to thank all the friends involved: C. Fu, L. Wang, J. Xu, L. Zhang, T. Liu, J. Chen, J. Du, H. Jiang, T. Ma, H. Lu, B. Liu, W. Liu, Y. Chen, F. Wei, J. Ning, Y. Na, etc.

Finally, I would like to give my special thanks to my parents and my wife N. Lu for always being there when I needed them most.

TABLE OF CONTENTS

Abstract	I
List of Figures	III
List of Tables	X
Chapter 1 Introduction	1
1.1 Motivation	2
1.2 Application.....	3
1.3 The Proposed Approach	4
1.4 Objectives.....	8
1.5 Input Data	10
1.6 Thesis Organization	10
Chapter 2 Literature Review.....	12
2.1 Human Motion Segmentation	13
2.1.1 Temporal-Based Segmentation	14
2.1.2 Spatial Approach	29
2.2 Human Pose Estimation	32
2.2.1 2-D approaches with explicit shape models	33
2.2.2 3-D approaches.....	37
2.2.3 2-D approaches without explicit shape models.....	45

2.2.4	Discussion	53
Chapter 3	System Overview	55
3.1	Pre-processing.....	56
3.2	Moving Object Outline	58
3.3	Motion Pattern Analysis	59
3.4	Convex Shape Modeling	61
3.5	Convex Shapes Combination	62
Chapter 4	Moving Outline Generation.....	64
4.1	Process Overview.....	65
4.2	Temporal Difference Edge.....	68
4.2.1	Image Loading.....	68
4.2.2	Temporal Difference Edge	68
4.3	Temporal Difference Outline	69
4.4	Moving Outline.....	73
4.4.1	Complement	74
4.5	Moving Outline Representation.....	76
4.5.1	Contour Extraction	76
4.5.2	Dynamic Two-strip Algorithm (Dyn2S)	76
4.6	Experimental Results	79
4.7	Summary and Discussion.....	90
Chapter 5	Moving Pattern Representation and Motion Vector Generation	91

5.1	Motion History Image (MHI).....	92
5.2	Front Motion History Image (fMHI).....	95
5.3	Inner MHI (iMHI).....	96
5.4	Fast Motion Measurement.....	102
5.5	Sedimentary MO	106
5.6	Motion Vector Generation.....	107
5.6.1	Motion Vector Extraction.....	107
5.6.2	Motion Vector Application	111
5.7	Summary and Discussion.....	113
Chapter 6	Convex shape modeling	115
6.1	Assumptions for Convex Shape Modeling	116
6.2	Overview of Convex Shape Modeling	117
6.3	Convex Shape Generation	118
6.3.1	Image Space Partition.....	119
6.3.2	Convex Shape Composition	122
6.3.3	Convex Shape Optimization using Dynamic Programming ...	124
6.3.4	Computation of Radius R and deviation σ	127
6.4	Convex Shape Combination	128
6.5	Summary and Discussion.....	130
Chapter 7	Human Body Modeling.....	132
7.1	Head and Torso Combination.....	134

7.1.1	Compatible Measures	134
7.1.2	Pairing Head and Torso.....	139
7.2	Limb Modeling	141
7.3	Head-torso-limb Modeling Refinement.....	141
7.4	Experimental Results	142
7.5	Summary and Discussion.....	147
Chapter 8 Experimental Results		148
8.1	Input Data	150
8.2	Experimental Results	155
8.2.1	Segmentation and Perception Experiment	155
8.2.2	Motion Pattern Analysis.....	172
8.2.3	Human Pose Estimation	198
8.3	Summary	211
Chapter 9 Conclusion and Future Work		212
9.1	Work summary.....	212
9.2	Contributions.....	213
9.3	Future Work	217
9.4	Thesis conclusion.....	218
References		219
Appendix A Sequences Used in Our Experiment		233

Appendix B Motion Segmentation and Motion Patterns Results.....244

Abstract

This thesis has made 3 major contributions. First a novel, effective and efficient framework to segment and keep track of Moving object Outline (*MO*) is proposed. We employ only temporal difference that takes difference of two consecutive frames and difference of the edge images followed by thresholding to extract outlines of moving regions. Compared to background subtraction, it works without referring to the background image and hence it does not need to spend valuable resources to update the background. It is suitable and convenient to be set up at non-predetermined scenes (mobile surveillance) with complex background. It is faster than optical flow method and can be applied in real time. Its effectiveness is measured by human perception when the segmentation results of 4 different types of inputs (totally 81 sequences) are displayed consecutively to 30 observers. Compared with other researches on motion segmentation, the proposed techniques are simple in concept and can work well in different kinds of environments such as dim lighting, lights changing, inside elevator and low quality images from close-circuit cameras or TV. This advocates a new direction for motion segmentation and becomes the major contribution of this work.

To recognize human behaviors, pixel intensities are transformed into features to represent different actions. Hence, the analysis of feature pattern plays a vital role for

human action recognition. The second contribution of this thesis is the derivation and design of useful motion features. We generate the front/rear motion history images ($rMHI$ and $fMHI$) and motion vectors along the moving object boundary to measure motion and distinguish which side of outline belongs to moving region. In addition, Sedimentary Moving object Outline (SMO) for representing motion history is employed to record MO that stop moving. Hence, deposited objects can be discovered and tracked. To track movement of overlapping body parts, inner MHI ($iMHI$) is proposed to isolate the mixed moving parts from each other and then track the movement of each isolated part. The $iMHI$ can be applied to generate motion feature for each tracked moving part.

The third contribution proposes one approach for human posture computation, which include two sequential processes: convex shape modeling and convex shape combination. Convex shape modeling aims to generate convex shapes that can be the candidatures for human head and torso. Then the head, torso and limbs combination is located by combining two convex shapes according to the topology of the estimated body posture.

The last two contributions are built on top of the first contribution of motion segmentation. Finally, the implementation has built a motion analysis system that supports video surveillance of human body motion.

List of Figures

Figure 1-1: Flowchart of proposed system.....	5
Figure 1-2: Illustration of the terms used in our work	6
Figure 1-3: Examples of the input images	11
Figure 2-1: Architecture of the detection process	19
Figure 2-2: The output of the region-level layered detection algorithm	20
Figure 2-3: Result of the detection algorithm	20
Figure 2-4: An example of the various shifted difference images	23
Figure 2-5: The optical flow is not always the same as the true motion field	25
Figure 2-6: Overview of the motion estimation process	26
Figure 2-7: An estimate of the motion at any point	27
Figure 2-8: The two motion models	27
Figure 2-9: The flow results for the trajectory-based method.....	28
Figure 2-10: A 2D stick-figure model	34
Figure 2-11: A 2D stick-figure model fleshed out with ribbons.....	36
Figure 2-12: The recovered 3D models ELLEN and DARIU	40
Figure 2-13: 3D model of a person	42
Figure 2-14: 3D model structure of a person	43

Figure 2-15: 3D model structure of human body	44
Figure 3-1: Flowchart of human body modeling	56
Figure 3-2: An illustration of extracted edges	57
Figure 3-3: Human body modeling processing	59
Figure 3-4: Example outputs of motion history images with motion vector	60
Figure 3-5: SMO detection from deposited object.....	61
Figure 3-6: Inner motion extraction	61
Figure 3-7: Examples of convex shapes detection	63
Figure 4-1: MO segmentation result of walking activity	66
Figure 4-2: Flowchart of the proposed system.....	68
Figure 4-3: Differencing outputs of a running sequence.....	71
Figure 4-4: Differencing outputs of a walking sequence with occlusion.....	72
Figure 4-5: number of regions versus different threshold values.....	73
Figure 4-6: Illustration of the dynamic two-strip algorithm with the selection of minimum and maximum strip widths	78
Figure 4-7: Illustration of the dynamic two-strip algorithm	79
Figure 4-8: An example of Line Edge Map.....	79
Figure 4-9: MO segmentation results of person squatting down	81
Figure 4-10: MO segmentation results of person breaking door with occlusion	82
Figure 4-11: MO segmentation results of person running	82
Figure 4-12: MO segmentation results of person walking.....	83
Figure 4-13: MO segmentation results of four persons walking.....	84

Figure 4-14: MO segmentation results of object-throwing	85
Figure 4-15: MO segmentation results of person walking in dim light environments	86
Figure 4-16: Samples of MO segmentation results from test set in light changing environments	88
Figure 4-17: Samples of MO segmentation results from activities inside an elevator	89
Figure 4-18: Samples of segmentation results from robbery scenes.....	89
Figure 5-1: Example to illustrate the process of MHI generation.....	94
Figure 5-2: MHI result of walking activity	95
Figure 5-3: rMHI and fMHI results of another walking activity	96
Figure 5-4: rMHI and fMHI results of standing up activity.....	96
Figure 5-5: Histogram segmentation.....	97
Figure 5-6: Histogram of pixel intensity of moving objects	98
Figure 5-7: The relation between MEI and iMHI	99
Figure 5-8: One example of 6 consecutive segmented results on iMHI	101
Figure 5-9: Inner motion extraction	102
Figure 5-10: Dilation of a swinging rod.....	103
Figure 5-11: nop MHI result of another example of a swinging rod	104
Figure 5-12: Non-Overlapping Parts (NOP)	105
Figure 5-13: The computation of SMO.....	107
Figure 5-14: Two-layer and three-layer MHIs	108
Figure 5-15: Motion vectors extraction for a running action.....	109
Figure 5-16: Chamfer distances from TR2	110

Figure 5-17: Examples of Motion vector extraction	113
Figure 6-1: The flowchart of convex shape modeling	117
Figure 6-2: Two examples of convex and concave shapes	118
Figure 6-3: Examples of negative and positive regions	121
Figure 6-4: An example of a PP from a real image	122
Figure 6-5: Convex shape definition.....	123
Figure 6-6: Illustration of the regional constraint	123
Figure 6-7: Line Chain Map Array.....	125
Figure 6-8: One example of a circle model.....	127
Figure 6-9: An example of two similar convex shapes	129
Figure 6-10: An example of the intersection of two regions.....	130
Figure 7-1: The flowchart of human body modeling	133
Figure 7-2: The link distance model	135
Figure 7-3: Three scenarios of link distances.....	136
Figure 7-4: Shape of equation 7-3.....	137
Figure 7-5: Example of rectangles enclosing head and torso.....	138
Figure 7-6: Shape of equation 7-5.....	139
Figure 7-7: Example in expanding the torso	140
Figure 7-8: Example of the three rectangles models.....	141
Figure 7-9: Head-torso-limb models	142
Figure 7-10: Human pose estimation results of people walking.....	144
Figure 7-11: Human pose estimation results of people falling down.....	145

Figure 7-12: Human pose estimation results of two persons walking	146
Figure 7-13: Convex shapes extraction for two people walking.....	147
Figure 8-1: Examples of test images; Set 1	151
Figure 8-2: Examples of test images; Set 2.....	152
Figure 8-3: Examples of test images; Set 3.....	152
Figure 8-4: Examples of test images; Set 4.....	153
Figure 8-5: MO segmentation results of two persons fighting.....	157
Figure 8-6: MO segmentation results of three persons walking	158
Figure 8-7: Four MO segmentation results of two persons fighting	159
Figure 8-8: Four MO segmentation results of three persons walking.....	159
Figure 8-9: Test scenes with less perfect perception rates	160
Figure 8-10: Samples of MO segmentation results from battering sequence in light changing environments	165
Figure 8-11: Motion segmentation comparison with relevant work	168
Figure 8-12: Comparisons of two object detection results.....	170
Figure 8-13: Reference motion vector generation for human running sequence.....	174
Figure 8-14: Fast motion pattern for the battering sequence	176
Figure 8-15: Motion patterns with vector results of people running.....	180
Figure 8-16: Motion patterns with vector results of people threatening	181
Figure 8-17: Motion patterns with vector results of people waving arm.....	182
Figure 8-18: Motion patterns with vector results	183
Figure 8-19: Motion patterns with vector results	184

Figure 8-20: Images for false positive and negative	188
Figure 8-21: Motion patterns with vector results of people running.....	190
Figure 8-22: Motion patterns with iMHI results of eight different human activities	192
Figure 8-23: Motion patterns with iMHI results of eight different human activities	193
Figure 8-24: Motion patterns with iMHI results of eight different human activities	194
Figure 8-25: The computation of SMO for the depositing object sequence	196
Figure 8-26: The computation of SMO for the knocking down people sequence	197
Figure 8-27: Convex shapes extraction for people walking.....	199
Figure 8-28: Convex shapes extraction for people bowing.....	200
Figure 8-29: Convex shapes extraction for people falling down	201
Figure 8-30: Convex shapes extraction from the robbery sequence	202
Figure 8-31: Convex shapes extraction for two people walking.....	203
Figure 8-32: Human pose estimation results of people bowing.....	206
Figure 8-33: Human pose estimation results of people falling down and jumping ..	207
Figure 8-34: Human pose estimation results of robbery sequence	210
Figure 8-35: Human pose estimation results of people waving arms.	211
Figure B-1: MO segmentation results of one person kicking	245
Figure B-2: MO segmentation results of one person sitting down	246
Figure B-3: MO segmentation results of three persons walking.....	247
Figure B-4: MO segmentation results of object-throwing	248
Figure B-5: MO segmentation results of one person kicking	249
Figure B-6: MO segmentation results of one person sitting down	250

Figure B-7: MO segmentation results of three persons walking.....	251
Figure B-8: MO segmentation results of two persons at night	252
Figure B-9: MO segmentation results of two persons at night	253
Figure B-10: Motion patterns with vector results of people falling down	254
Figure B-11: Motion patterns with vector results of people kicking	255
Figure B-12: Motion patterns with vector results of people waving arms.....	256
Figure B-13: Motion patterns with vector results of people bowing	257
Figure B-14: Motion patterns with vector results of people pushing a trolley	258

List of Tables

Table 4-1: Symbols and abbreviations	65
Table 5-1: Symbols and abbreviations	92
Table 5-2: Motion region classification	112
Table 8-1: Description of Input Data.....	154
Table 8-2: List of actions and acceptable descriptions.....	161
Table 8-3: Responses of Test Set 1	161
Table 8-4: Responses of Test Set 2.....	162
Table 8-5: Responses of Test Set 3.....	162
Table 8-6: Responses of Test Set 4.....	162
Table 8-7: Motion region classification for Figure 8-15	177
Table 8-8: Motion region classification for Figure 8-16.	177
Table 8-9: Motion region classification for Figure 8-17.	178
Table 8-10: Motion region classification for Figure 8-18.	178
Table 8-11: Motion region classification for Figure 8-19.	179
Table 8-12: Test result of motion pattern detection for Test Set I.....	185
Table 8-13: Test result of motion pattern detection for Test Set I.....	185
Table 8-14: Test result of motion pattern detection for Test Set I.....	186

Table 8-15: Test result of motion pattern detection for test set II	186
Table 8-16: Test result of motion pattern detection for Test Set III	187
Table 8-17: Test result of motion pattern detection for Test Set IV	187
Table 8-18: Human pose estimation Success Rate test results for Test Set I	208
Table 8-19: Human pose estimation Success Rate test results for Test Set I	208

Chapter 1 Introduction

Automatic visual analysis of human motion has recently become one of the most significant research topics in computer vision. A recognition system, which can learn explicit human behaviors, is desirable in many applications. To recognize human behaviors, high-dimensional data (e.g. pixel intensities in original frames) is transformed into a low-dimensional representation (feature) to facilitate abstracting and recognizing different actions. To facilitate learning and recognition, the processes on motion segmentation, feature (motion) pattern analysis and human posture estimation forms the basis for recognizing human behaviors. This thesis focuses on these supporting but important areas.

1.1 Motivation

With recent advances in the technologies and developments of computer processing, producing video sequence to record human behavior is becoming increasingly easy. In recent years, automatic visual surveillance and monitoring of human activity have become the keys to enhance security. Many researchers [1-7] have shown great interest in these areas. In human vision, human beings use a cognitive process that involves activation of sense organs, perception, abstraction, and memory to aid the recognition and interpretation of human behaviors from the video sequence involving human. The physiological counterpart of these processes is still not completely clear. By analogy with the human cognitive process, the human behavior recognition from visual information by a computer can be decomposed into three consecutive steps: human detection, human posture estimation and human behavior understanding. The technical issues of human motion analysis touch on many of the core topics of computer vision, pattern analysis, machine learning, and artificial intelligence.

One of the low level but important processes that the previously mentioned tasks need to build on top is "motion segmentation". Motion segmentation facilitates a system to focus on moving objects rather than spending valuable resources to interpret the stationary background. Similar to the classic study on human body motion perception by Johansson using moving light displays [8;9], the segmentation results should allow easy recognition of biological motion patterns when the segmented frames are displayed consecutively to human observers. There are many approaches

[2;10-16] to isolate and keep track of moving objects. Unfortunately, most approaches are either time consuming [12;14;16] or difficult to use that one needs to update the background information [2;11;13]. Due to these problems, it motivates us to propose a novel, effective and efficient framework to segment and track moving object outline employing only temporal difference.

For another activity of human behavior recognition, posture computation or recovery can play an important role. Posture recovery mainly focuses on estimating the configuration of an articulated human body and providing direct correspondence between labeled body parts and regions in the image. Generally speaking, the posture estimation related work can be reviewed and classified into two categories: approaches with explicit shape models and approaches without a prior shape model. Since the field of human posture estimation analysis is currently subject to intense investigation and yet still immature, we propose here a posture recovery method by employing convex shape modeling in our work.

In this thesis, on one hand, we put our most effort on human motion segmentation and human posture recovery, where quite a lot of work has been done. On the other hand, investigation of human behavior understanding is not presented due to time constraint. However, the proposed system is complete enough to show meaningful results and the human posture detection proposed in our work can provide valuable support for human behavior recognition.

1.2 Application

A capable system of recognizing human actions would be very useful to wide

spectrums of promising applications in many areas. For example, during the past several years, there are many smart surveillance systems being proposed, such as *Pfinder* [7], *W4* [2], *Sensor Forest* [6;17;18], and *VSAM* [56]. The systems have the abilities to automatically detect particular human activities which are considered harmful or dangerous from normal movements. They can be used in many security sensitive areas such as banks, stores, airports, parking lot, borders and so on. Generally speaking, all the applications are heavily relied on and built on top of human segmentation and posture estimation results. Another application is virtual reality system that can also be made more exciting using an action recognition system, where the users control navigation by moving their own body. The user can control the graphic character to run, jump, kick, and do other actions using the corresponding true body motion, which will give the user a true feeling of involvement in virtual world. Similarly, an athletic performance analysis system could be developed in this framework that observes the individuals' exercise movements and offers the feedback on the performance of exercises. Also, with the development of digital libraries, the ability to automatically describe video sequences will save much energy and effort in sorting and retrieving images or video sequences using content-based queries. Other applications include video conferencing, etc. All of the above applications require the ability to automatically and robustly recognize human actions.

1.3 The Proposed Approach

In this thesis, we outline a prototype body posture recovery system. The system flowchart is proposed in *Figure 1-1*. In our work, we mainly focus on three components:

motion based human body acquisition, motion pattern analysis and human posture computation.

Figure 1-1: Flowchart of proposed system.

These three components represent three of the main processes of human behavior recognition. Generally speaking, there are three compositions for human behavior recognition: human body segmentation, body posture recovery module and human activities recognition. Hence, in our work, we explore the first two. Human body segmentation and motion pattern analysis share some common procedure or activities such as they both use temporal difference based method to extract motion information. The human body segmentation results are then fed to the human posture computation process.

In our system, the motion based human body acquisition aims to segment the human body that moves. Human posture computation aims to reconstruct and configure a posture such that it accurately reflects the position and the configuration of the human body after the motion capture process. Moreover, we propose motion pattern analysis to exact region description for image data understanding and classification. In order to

have an overview of the new terms that will be discussed later in this section, a block diagram is presented in *Figure 1-2*.

Figure 1-2: Illustration of the terms used in our work.

This thesis presents a novel, effective and efficient framework to segment and track Moving object Outline (*MO*). Generally speaking, moving object outlines can be applied to human action perception. Outlines are by far the most commonly used information for meaningful image analysis. There is also much psychological evidence, such as our ability to understand cartoons, that outlines are sufficient for perception of human behaviors. In addition, outlines/edges are relatively insensitive to light changing and can be extracted in most images for representation. Cognitive psychological studies [20] indicated that human beings recognize line drawings as quickly and almost as accurately as gray-level images. These investigation results might imply that outlines of moving object could be employed for human behaviors recognition and to achieve similar accuracy as gray-level images. This is the reason why we extract *MO* as motion

segmentation result and further feed them to human posture estimation system to analyze human behaviors.

In our work, we segment *MO* rather than moving edge since we can apply more restrictions on the former to get genuine moving line segments. A stationary monochrome camera is assumed. We employ only temporal difference that takes difference of two consecutive frames and difference of the edge images followed by thresholding to extract outlines of moving regions. To the best of our knowledge, this is the first piece of work using only temporal difference for effective motion segmentation. Temporal difference has many advantages. Compared to background subtraction, it works without referring to the background image and hence it does not need to spend valuable resources to update the background. It is suitable and convenient to be set up at non-predetermined scenes (mobile surveillance) with complex background. It is faster than optical flow method and can be applied in real time. In addition, it has good adaptation to light changing and dim environment. It is found that the motion activities can be perceived by human observers easily when the segmentation results are displayed consecutively. Then, we generate the front/rear motion history images (*rMHI* and *fMHI*) and motion vectors along the moving object boundary to measure motion and distinguish which side of outline belongs to moving region. The results will be *MO* and motion vectors detected from each frame using only temporal difference. The motion vectors can be further applied to classify normal or fast human behaviors. In addition, Sedimentary Moving object Outline (*SMO*) for representing history mechanism is employed to record *MO* that stop moving. Hence, deposited objects can

be discovered and tracked. To track movement of overlapping body parts, inner *MHI* (*iMHI*) is proposed to isolate the mixed moving parts from each other and then track the movement of each isolated part. Then the *MHI* can be applied to generate motion feature for each tracked moving part.

Finally, we propose one approach for human posture computation, in which it can reconstruct and configure a posture such that it accurately reflects the position and the configuration of the human body after the motion capture process. For the human posture computation, there are two sequential processes: convex shape modeling and convex shape combination. Convex shape modeling aims to generate convex shapes that can be the candidatures for human head and torso. In the first place, convex shapes are obtained by employing dynamic programming. In addition, prediction and angle constraints of convex shapes' lines are introduced to reduce the complexity of the convex shape generation process. In the second place, during labeling the body part, the convex shapes are first determined as the candidates of body parts. Then the head and torso combination is located as a starting point by combining two convex shapes according to the topology of the estimated body posture under the assumption that head-torso model is a stable body part combination compared to only one body part such as head, torso or limbs. Lastly, when the head-torso location was determined, other primary body part such as limbs are mapped using prior knowledge about the topology of the estimated body posture.

1.4 Objectives

The major objective of this work is to design, investigate and develop a novel,

efficient and effective motion segmentation method and further apply these motion segmentation results to human pose estimation system for automatic acquisition of a human body model. Several objectives are to be fulfilled in our study.

1) Derivation of a novel, effective and robust motion segmentation method:

The proposed motion segmentation method should be able to extract a moving human body that performs different kinds of human activities such as walking, bowing, wresting, falling down, etc. We will test on 4 different types of inputs to observe and evaluate the performances. These 4 types inputs cover different kinds of normal and fast human behaviors under various scenarios such as dim lighting, light changing, inside elevator and low quality images from close-circuit cameras or TV. Such kinds of test environment could be complex enough to test the capability of the system,

2) Derivation of robust and reliable motion features:

The proposed human motion feature representation should be able to handle different motion pattern such as slow, fast and stop movements. In addition, the overlapping movements should also be detected. For example for the partial occlusion scenario, the proposed system should be capable of identifying and recognizing movement of overlapping body parts and generate motion feature for each tracked moving part.

3) Human body posture recovery:

With the motion segmentation results discussed previously, a scheme is sought to extract convex shape as the body part ingredients. An integration method

should then be derived to combine the body parts into a feasible human body posture through the analysis of a structural view of body poses.

1.5 Input Data

The proposed approach performance in this work is evaluated by different kinds of human behaviors. According to the environment, we separate the test sequences into four groups that are normal lab scenario, dim light, light changing, and low quality video from surveillance tape or TV. The normal lab scenario consists of 45 groups of image sequences and the activities include normal human behaviors such as walking, sit down, making a phone call, etc. In addition, there are fast behaviors that include running, crawling, fighting and knocking at the door, etc. For the dim and light changing groups, the human activities are similar with the previous group but the main difference is that all the actions happened in dim or light changing environment. For the last group, the environment is very complex and the image qualities are poor. The activities include robbery, running and attack. We observe frequent light changes caused by windows facing outside and people passing by just outside the window. In addition, the environment of dim light inside an elevator can also be found in this test set. Several examples of the test images are displayed in *Figure 1-3*.

1.6 Thesis Organization

In this thesis, there are totally 9 chapters including this chapter. A survey of previous work is reviewed in chapter 2. Then, the system overview is carried out in chapter 3. Chapter 4 introduces one new approach to segment moving object outline

using the technique of temporal difference. Several motion pattern representations to abstract the image data and the detailed description on motion vector generation are described in chapter 5. The generation of convex shapes using dynamic programming is illustrated in chapter 6. The combination of convex shapes and human body modeling are discussed in chapter 7. In chapter 8, we compare and discuss the experiment results for system evaluation. Finally, the thesis conclusion and future work are given in chapter 9.

Figure 1-3: Examples of the input images.

Chapter 2 Literature Review

Human motion capture and understanding are currently two of the most active research topics in computer vision that has two to three decades history and yet many fundamental problems remain unsolved. This chapter reviews the related work in the above two areas in which they are classified into two different categories:

- Human motion segmentation: An attempt is made to segment the foreground motion region out of the scene that is discussed in details in *section 2.1*.
- Human motion recognition: A system attempts to understand and analyze human body motion using models that is discussed in details in *section 2.2*.

The aim of human motion segmentation is to solve the problem of extracting human image as motion objects of interest from the image sequences. Generally speaking, the motion objects can be regarded as the foreground while the scene without motion objects can be considered as background. From this point of view, motion segmentation is the process of differentiating moving foreground objects from the background.

After human motion segmentation, human motion recognition mechanisms are discussed. Many human motion recognition methods have been proposed and can be classified based on different criteria. In this chapter, the human motion recognition approaches are categorized into two classes based on whether information of the object shape is employed in the motion analysis: approaches with explicit shape models and approaches without a prior shape model. A brief review of related works and techniques on the analysis of human motion will be given. More extensive surveys on the understanding of human motion can be found in [21-25]. Some related issues such as gait and gesture analysis for recognition and classification are also discussed in this chapter.

2.1 Human Motion Segmentation

Almost in all human behavior recognition systems, the first fundamental basis is motion segmentation that isolates moving objects from the rest of an image for further processing. Subsequent processes such as human tracking, pose estimation and behavior recognition are heavily dependent on it. Generally speaking, there are two methods for motion segmentation from image sequences, which are based on either

temporal [26-28] or spatial [29] information. For the spatial-based segmentation method, it commonly uses the characteristics of individual pixels or groups of pixels to obtain the objects from the background. For example, colors and edges are two typical characteristics. The motion can then be inferred. For the temporal-based segmentation method, it is assumed that the differences between images from a sequence should originate from movement of the object. In this case, the method is usually based on the assumption of a static background.

2.1.1 Temporal-Based Segmentation

The temporal-based segmentation method also falls into two subclasses: subtraction [3;13] and optical flow [47, 48, 49, 50, 51]. The subtraction method includes background subtraction and temporal difference techniques.

2.1.1.1 Background Subtraction

The use of background subtraction is very popular recently. When a scene is static, a background image without any objects can be recorded and used as a reference in the subtraction scheme that takes the difference between the current image and the reference background image in a pixel-by-pixel fashion. This section surveys techniques that are analyzed with respect to four important attributes: how the background is established initially (*preprocessing*); how the background is represented (*background representation*); how the object areas are distinguished from the background (*foreground detection*); and, how the background is maintained over time (*background maintenance*).

1. Preprocessing

Usually, reference background is initialized by estimating the model parameters during a short period under the assumption that no foreground objects occurred during the training period. However, in some real scenes, e.g., public areas, it is difficult or impossible to control the area being monitored. Recently some approaches have been proposed for learning reference background from frames even when moving objects are present in the scene.

Generally, motion detection can be utilized for reference background initialization. For example, Gutches *et al.* [35] proposed a method to initialize the background at each pixel in which it was independent of its neighbors. Meanwhile, the decision at each pixel was also relied on low-level motion information. In their work, the motion information was mainly used to choose candidate background intensities at each pixel, since such motion information could identify intensity transition from foreground to background and vice versa. Furthermore, Farin *et al.* [36] excluded pixels in moving objects from reference background by using block-based motion detection results.

Cohen [37] had shown capability of estimating reference background on each pixel level by regarding the problem as an exercise in optimal labeling. However, there were several assumptions for his method: (1) all of the required background is visible for some of the time; (2) the required background was more consistently stable than any other foreground pixel intensity; and (3) each background pixel was time-independent. Generally speaking, these prerequisites were not always satisfied

[38].

In the a mixture of Gaussian (*MoG*) representation moving objects can to some extend be accepted during initialization since each foreground object will be represented by its own distribution which is likely to have a low weight. For example, Stauffer and Grimson [6] introduced an approach to model each background pixel by using the ratio of a mixture of adaptive Gaussians. Eng *et al.* [39] reported on a ‘drowning-detection’ system consisting of multiple cameras by using a skin detector to find and remove humans from the training images. Furthermore, Kim *et al.* [40] represented each foreground pixel as a number of codewords during runtime. This codeword method used a temporal filter after the initialization phase to eliminate any codeword that had not recurred for a long period of time. A similar approach was used in [41;42] too.

2. Background Representation

Stauffer *et al.* [17] presented a theoretic framework for recovering and updating background images. It is based on a process where *MoG* model was used for each pixel value and the model was updated in run time. Then the *MoG* model was evaluated to decide which part was most likely from a background process. This system could track people and cars in outdoor environments by applying background subtraction. Currently this method has become the standard of background subtraction and *MoG* representation is the most widely used method by far, in which the work in [43;44] further optimized the *MoG* representation. In addition the representation of

MoG where the color and intensities were separated could be applied with many different color spaces [45], e.g., *RGB*, *YUV* [7], *HSV* [46], and normalized *RGB* [47]. We could benefit from these representation with different color spaces, such as the work in [48] used *HSV* color space for shadow detection. *MoG* representation could also be applied in a 3D color space in which it corresponded to ellipsoids or spheres of the Gaussian representations as presented in [17;47;49]. Truncated cylinders [40] and truncated cones [50] could also be used as geometric representations.

However the *MoG* representation could not work well for the scenarios with dynamic background and hence methods directly aimed at modeling dynamic background were applied in [51-53]. Monnet *et al.* [51] proposed a prediction-based online background model that explicitly tried to model dynamic textures. It seemed that their model was more suitable for repetitive patterns such as ocean waves and swaying trees. However, they required hundreds of images without moving objects to establish the background model, and the moving object could not be found if they had the same movement direction as the ocean waves. As in the approach proposed by Sheikh and Shah [52], the background distribution of a pixel was estimated by nonparametric density estimation over samples by extending the single pixel to a neighborhood of that pixel. However their detection result was compromised for the scenarios that waves almost appeared everywhere. Furthermore, this method had been extended in [54]. Approach in [53] aimed to model the dynamic texture by an Autoregressive Moving Average Model (*ARMA*). A robust Kalman filter algorithm was employed to iteratively estimate the intrinsic appearance of the dynamic texture

and then predict motion patterns in the scene.

3. Foreground Detection

The method proposed by Cutler and Davis [13] was somehow similar to that of Polana and Nelson [55]. They detected motion regions using background subtraction method. The regions were then tracked by their sizes and position history, or by the image template matching. The tracked object was aligned using its centroid based on the tracking results and normalized in scale using the object's position and a ground plane model. In the next step, Fourier analysis was applied to detect and characterize periodic movements.

Cucchiara *et al.* presented Sakbot (Statistical And Knowledge-Based Object detection), a system for moving object detection in image sequences in [17]. This system had the unique characteristic of explicitly addressing various troublesome situations such as cast shadows and ghosts. Cast shadows were detected and removed from the background update function. The main contribution of this system was the integration of knowledge of detected objects, shadows, and ghost in the segmentation process to enhance both object segmentation and background update.

VSAM [56] had developed a hybrid algorithm for motion segmentation by employing a layered adaptive background subtraction and three-frame differencing algorithm and it was effective for detecting moving objects in image sequences. In their system, they first extracted motion regions with three-frame differencing, and then filled out the holes within the bounding boxes of the regions using background

subtraction. The layered detection was composed of two processes: pixel analysis and region analysis. The purpose of pixel analysis was to determine whether a pixel was stationary or transient by observing a per pixel basis intensity value over time. Region analysis dealt with the agglomeration of groups of pixels into moving regions and stopped regions. Then a layer management process keeps track of the various background layers. *Figure 2-1* showed the flowchart of such processes.

Figure 2-1: Architecture of the detection process.

The output of the region-level layered detection algorithm was displayed in *Figure 2-2*. The detected regions were shown surrounded by bounding boxes, where all three overlapping objects were independently detected. Each stopped car was depicted as a temporary background layer, and the person was determined to be a moving foreground region overlaid on them. The pixels belonging to each car and to the person were well disambiguated.

Figure 2-2: The output of the region-level layered detection algorithm.

One segmentation result was shown in Figure 2-3. However, since a moving object might be separated into small regions in the three-frame difference, it would still fail to extract the whole body of the moving object in some cases.

Figure 2-3: Result of the detection algorithm; (a) Original image. (b) Detected motion regions.

Stringa [57] proposed a novel background subtraction algorithms based on mathematical morphology for detecting motion in scenes. This algorithm obtained stable segmentation results even under varying environmental conditions. In addition, comparison with some of the different background subtraction approaches are mentioned in [58;59]. Although some adaptive background subtraction methods have been developed with good results, but additional expensive process, i.e. adaptive

background maintenance, must be employed to make the background subtraction robust to image noise and gradual change of lighting conditions.

4. Background Maintenance

The method proposed by Haritaoglu *et al.* [2] learned and modeled background scenes statistically to detect foreground objects. Each pixel was presented with three values: its minimum and maximum intensity values and the maximum intensity difference between consecutive frames observed during the training period. These three values were used to update the background periodically. In their system, they used two different methods to update the background. One was a pixel-based update method that updated the background model periodically to adapt to illumination changes in the background scene. The other was an object-based update method that updated the background model to adapt to physical changes in the background scene. A deposited object could be added into the background scene if it did not move for a long period of time.

McKenna *et al.* [5] used an adaptive background model with color and gradient information to reduce the influences of shadows and unreliable color cues. There is another approach for resolving updating of background images which included parameter estimation based on pixel processes [18]. Ridder *et al.* [60] modeled each pixel value with a Kalman Filter to compensate for illumination variance. Yalcin *et al.* [61] obtained estimation and compensation that used sparse flow of corner features.

2.1.1.2 Temporal Difference

Apart from background subtraction, another intuitive approach to detect changes is to take the pixel-wise differences of two or three consecutive frames followed by thresholding in an image sequence. The change at each pixel can be detected if the difference in gray level exceeds a preset threshold, which is going to be adjusted dynamically in this work employing local structural information. Some other adaptive threshold selection methods can be found in [62;63].

As an example of this method, Lipton *et al.* [19] detected moving targets in real video streams using temporal difference. After the absolute difference between the current and the previous frame was obtained, a threshold function was used to determine changes. By using a connected component analysis, the extracted moving sections were clustered into motion regions. One general problem with temporal difference method is that it tends to include undesirable background regions on the periphery of the target where the object has “just been”. In their system, they could not tackle this problem. In addition, when multiple human targets were close together, they could be misclassified as a vehicle. Furthermore, another problem with temporal difference approach is that pixels from an object that does not move or are similar to their neighbors are difficult to detect. Therefore, one improved method to solve this problem was proposed by employing three consecutive images [56].

In Shao’s *et al.* work [64], they used a combination of frame difference and template matching to determine all the possible moving regions which belonged to potential target candidates.

Viola *et al.* employed a different type of temporal image difference mechanisms for faces [65] and pedestrians [66] detection. In [65], they employed machine learning to construct a detector from a lot of training examples and then found pattern of intensities by scanning all the input images. This system worked quite well for detection of faces. Later in [66], they described a pedestrian detection system that integrated image intensity information with motion information. In their system, firstly the direction of motion could be extracted from the difference between a rectangle of pixels in the current image and the corresponding rectangle in the previous image, in which it was generated by shifting the rectangle in the current image up, down, left and right. Motion filters operated on 5 images for four directions of shifting:

$$\Delta = \text{abs}(I_t - I_{t+1}); \quad U = \text{abs}(I_t - I_{t+1} \uparrow); \quad L = \text{abs}(I_t - I_{t+1} \leftarrow);$$

$$R = \text{abs}(I_t - I_{t+1} \rightarrow); \quad D = \text{abs}(I_t - I_{t+1} \downarrow)$$

where I_t and I_{t+1} were images in time, and $\{\uparrow, \downarrow, \leftarrow, \rightarrow\}$ were image shift operators ($I_t \uparrow$ was I_t shifted up by one pixel). Motion information of the direction of motion was extracted from the difference between shifted versions of the second image in time with the first image. One example of these images was shown in *Figure 2-4*.

Figure 2-4: An example of the various shifted difference images.

In this figure, the first two images were two typical frames with a low resolution pedestrian pictured. The following images showed the Δ, U, D, L and R results. Then the output with the lower energy had the higher probability that the human had actually moved along that direction. Finally the training process used *AdaBoost* to select a subset of features and construct the classifier.

2.1.1.3 Optical Flow

Optical-flow-based motion segmentation [67] used characteristics of flow vectors of moving objects over time to detect moving regions in an image sequence. However optical flow can be caused not only by object motions, but also camera movements or illumination condition changes.

Figure 2-5 illustrates two special cases. In the first example, a sphere with a uniform surface is rotating under a constant ambient light. Because every point on the sphere reflects the same color, the eye cannot observe any change in the color pattern of the imaged sphere and thus considers the sphere as being stationary. In the second example, the sphere is stationary, but is illuminated by a point light source that is rotating around the sphere. The motion of the light source causes the movement of the reflecting light spot on the sphere, which in turn can make the eye believe the sphere is rotating. The real motion is obviously zero because no 3-D motion is present, while the change in intensity induces optical flow and thereof apparent motion. The observed of apparent 2D motion is referred to as optical flow in computer vision literature.

Figure 2-5: The optical flow is not always the same as the true motion field.

The above examples reveal that the optical flow may not be the same as the true 2D motion. That's why there are two assumptions for optical flow computation: 1) the observed brightness of any object point is constant over time; and 2) Nearby points in the image plane move in a similar manner.

For a continuous image, let $f(x,y,t)$ denote the pixel intensity of (x,y) at time t . We can assume that the region at (x,y) is translated some small distance (dx, dy) during the interval dt . The location (x,y) will now be occupied by the pixel that occupied $(x-dx, y-dy)$ at time t . The rate of change of intensity at (x,y) will then be given by

$$f_t = -f_x \frac{dx}{dt} - f_y \frac{dy}{dt} \quad (2-1)$$

Denoting the flow, i.e. $(dx/dt, dy/dt)$ as (u,v) we have the brightness constraint which is an equation that the components of the flow satisfy at each pixel:

$$f_t + f_x u + f_y v = 0 \quad (2-2)$$

In Meyer *et al.*'s work [30], the initialization of the segmentation was done automatically using information on the optical flow. The results were used for gait analysis. Optical-flow-based methods could be used to detect independently moving

objects even in the presence of camera motion. The method was based on monotony operators which computed the displacement vector field to initialize a contour based tracking algorithm, called active rays, for the extraction of articulated objects. More detailed discussion of optical flow could be found in Barron's work [31].

Gibson and Spann proposed an approach to the problem of estimating a dense optical flow field [32]. In their work, there were two steps named motion trajectory estimation and dense flow estimation to generate dense optical flow field. The motion trajectory estimation stage aimed to compute a set of trajectories spatially distributed throughout the image describing the motion of a set of points over multiple frames. Then dense flow estimation process interpolated the sparsely computed trajectories to give a densely computed flow field describing the motion between two frames. These two steps were described in *Figure 2-6*.

Figure 2-6: Overview of the motion estimation process.

The proposed method offered a mid-way solution between pixel-based optical flow estimation and sparse feature point tracking methods. The generation of a dense optical flow field from an irregularly spaced motion trajectory set could be seen as a process of interpolation. Furthermore, an estimate of the motion at any point could be inferred from some or all of the surrounding motion trajectories. Logically if a motion

model, valid in the vicinity of some point (x_0, y_0) was required, then this should be based on the trajectories spatially close to (x_0, y_0) . This was shown in *Figure 2-7 (a)*.

Figure 2-7: An estimate of the motion at any point. (a) Motion model defined using information from spatially close trajectories and (b) effect of motion boundaries.

For any given point, to represent the motion at that point by the surrounding trajectories, the simplest approach was to consider the spatially closest trajectories. However basing a motion estimate on all of the spatially close trajectories would lead to problems near to motion boundaries, as could be seen in *Figure 2-7 (b)*. Such a simple approach resulted in the inevitable problem of over-smoothing of the motion field across the boundary. Hence, a separation of trajectories was then classified into distinct groups of similar motion, such that no two trajectories belonging to objects with different motion were used to estimate a given motion model. This process was illustrated in *Figure 2-8*.

Figure 2-8: The two motion models (shaded areas) are each estimated using trajectories from one region only.

The flow results for the trajectory-based method were shown in *Figure 2-9*.

Figure 2-9: The flow results for the trajectory-based method; (a) Reference frame of the Yosemite sequence, (b) true flow field, (c) estimated flow field-trajectory based method.

One of the main limitations of the method is the computational complexity. This is largely due to the use of the simulated annealing optimization framework used to compute both the trajectory information and the optical flow field.

Blostein *et al.* [33] proposed an approach for estimating the 3-D trajectory and structure of a moving object. In their approach, they used an extended Kalman filter that combined optical flow and feature point measurements. Another approach proposed by Memim and Perez [51] used a multi-resolution framework centered on the minimization of the optical flow constraint and a regularization term, with use of robust cost functions. This optimization is coupled with a segmentation mechanism that identified and preserved motion discontinuities.

Sidenbladh [68] calculated optical flow from a pair of consecutive images in a video shot of the scene. Image windows containing a walking human was then test against a support vector machine (*SVM*) to detect walking humans in video, in which the *SVM* was trained with human and non-human flow patterns. While optical flow could be noisy, image flow could be classified by employing higher level entities

instead. For example, Gonzalez *et al.* [69] proposed a method to generate flow vectors by feature based tracking of human motion. Sangi *et al.* [70] extracted flow vectors by applying particle filters.

Ju *et al.* [71] described a 2D approach where person's limbs were represented by planar patches connected to each other at joints. Then the optical flow parameters provided a rich and concise description of the activity that could be used for human activities recognition. Bregler and Malik [72] employed a mathematical result familiar to roboticists and used optical flow to track the human body. Both methods required manual initialization after which the optical flow based approach tracked the body.

Although the optical flow technique can detect independently moving subjects and estimate the shape of the moving object quite well and at the same time the application of optical flow have been well studied and shown good results in computer vision, most optical flow computation methods are very complex, sensitive to noise, and not applicable to video streams in real time system without high performance equipments.

2.1.2 Spatial Approach

For the spatial-based segmentation method, it commonly uses the characteristics of individual pixels or groups of pixels to obtain the objects from the background. Many of the recent efforts extract object models based on such spatial features, other than temporal information. The properties that the features try to capture mainly include: 1) appearance, e.g. changing/noise model [10], color histograms compared with Bhattacharyya distance [73;74] and 2) shape, e.g. the part

templates in [75], the *SIFT* like orientation feature in [76;77], and the Histogram of Oriented Gradient (*HOG*) descriptor [78]. Hence we categorize these spatial approaches by whether information of the object appearance or shape is employed and review them in the following two sections: appearance-based segmentation and shape-based segmentation.

2.1.2.1 Appearance-based segmentation

Appearance-based techniques segment moving objects by employing the appearance model of the connected region, in which it is built on the ideas that: the appearance of human and background is different and the appearance of individuals are different.

Previously appearance-based methods segment object by comparing a 2D representation of object appearance against many prototype representations stored in a memory, and finding the closest match. For example, Poggio *et al.* [79] proposed a appearance based system by predicting different patterns of generalization to classify wire objects. Later Murase *et al.* [80] developed a system by addressing the problem of automatically learning object models not only for recognition but also for pose estimation. However, one problem for this approach is that we need to find the closest match of image representations of complete objects. To represent entire objects, it requires that the image is segmented into different regions. Unfortunately this kind of bottom-up high-level perceptual grouping seems to be invalid and impossible in the presence of clutter, occlusion and rotation of the object.

Utsumi *et al.* [81] proposed a method to describe and recognize the appearances

of objects based on geometrical structures. In their system the image was divided into a number of blocks and the mean and covariance matrix of the intensities were calculated for each block. Then this representation was based on a statistical analysis of Mahalanobis distances among each block of images. Finally objects with pixel value variety were recognized by employing a small number of appearance models. In addition, appearance-based methods could also be employed to detect humans in a still image [82] to detect humans entering a scene [83], to build appearance models by looking for coherence in the detections [84], or to index images in databases [85]. Generally speaking, these appearance-based methods have three significant defects. Firstly since human is usually modeled as a box, clutter in the background will therefore affect on the output segmentation results. Secondly, since these methods normally represent the human as one entity instead of a number of sub-entities, occlusion will strongly affect the segmentation outputs. Finally, since the models are general and do not adapt to the current scene, illumination changes will also affect the segmentation outputs.

2.1.2.2 Shape-based segmentation

Generally speaking, the shape of a human being is often very different from the shape of other kinds of objects in an image, in which it is a significant cue for shape-based segmentation. As opposed to the appearance-based models, the shapes of human beings are often very similar. Usually the shape-based segmentation methods allow human detection and tracking in unpredictable scenarios.

A lot of shape features can be employed for shape-based segmentation method,

such as contour fragments in [86;87], and edgelet in [88], were chosen by employing boosting algorithm to model the shape of the objects. They selected the features on the object boundary and their responses on the query image were used to describe the object. Later, based on these results, Wu *et al.* [89] proposed the base classifiers for detection and segmentation by analyzing the local shape features, and later generated cascade structured classifier by boosting the base classifiers. In addition, Wu [90] developed an approach based on a two-layer statistical field model that characterized the prior shape variations as a Boltzmann distribution in a Markov Field.

Zhao *et al* [91] developed a robust algorithm for extracting the silhouettes of individuals in cluttered scenes through stereo-based segmentation and neural network-based recognition. Later to make the algorithm more robust the gradients of the outline of a silhouette were further employed to represent the shape of the human being. Leibe *et al.* [92] combined local appearance features and stored templates from the outlines of walking humans. Zhao and Davis [93] proposed an algorithm of hierarchical template matching for human upper body detection. They estimated the similarity between a template and an image using the chamfer distance, in which the matched template provided shape constraints on segmentation. Furthermore, another approach used silhouette information in classification framework [94].

2.2 Human Pose Estimation

Human pose estimation analysis is currently one of the most active research topics in computer vision in the last two to three decades. Pose estimation refers to the process of estimating the configuration of an articulated body and providing direct

correspondence between labeled body parts and regions in the image.

Here we review and classify the work into two categories: approaches with explicit shape models and approaches without a priori shape model. Furthermore, the explicit shape models can be further analyzed as 2-D approaches and 3-D approaches while approaches without a priori model are generally based on 2-D analysis. In this chapter, a thorough review of related works and techniques on the analysis of human motion will be given. More extensive surveys on the understanding of human motion can be found in [22;24;25].

The basic theory of human motion analysis is initiated from a psychology point of view. There have been the classic studies on human perception by Johansson [8;9]. In his experiments, the moving light displays (*MLDs*) were attached to body parts of an actor dressed in black, and moving in front of a dark background. The experiments showed that human observers could almost instantly recognize biological motion patterns even when presented with only a few of these moving dots. There are two observations that one can conclude from the result of the research. Firstly, people can infer motion information from the *MLD* to recover 3D structure and subsequently use the structure for recognition. The second observation is that the movement patterns could be recognized directly from feature points without structure recovery [95].

2.2.1 2-D approaches with explicit shape models

Although the approaches without explicit shape models are necessary when no priori shape models are available, it is usually difficult to establish feature correspondence between consecutive frames. On the other hand, since human action

can be regarded as the motion of non-rigid or articulated objects, a prior knowledge of how the human appears in 2D can be exploited to segment, track, and label body parts. Thus most methods for the motion analysis of human body parts use the 2D models, such as stick figures, wrapped around with ribbons or “blobs”, to fit to the given image. The background subtraction is typically used to segment the human figure. Some of the methods are described in the following.

Guo *et al* [96;97] proposed an approach using model stick-figure to describe and recognize human motions. The human motion was recorded as sequence of the stick figure parameters, which could be used as input to a motion pattern analyzer. A model stick-figure was extracted by matching the stick model to the image-skeleton found from the silhouette in which a 2D stick-figure model is shown in *Figure 2-10*.

Figure 2-10: A 2D stick-figure model.

The method successfully classified the actions of walking, running, and others

with quite a high recognition rate. But only the movement of one person walking or running parallel to the image plane could be recognized. Nguyen *et al.* [98] also used this kind of representation of human body to build a novel hierarchical model of human dynamics encoded using Hidden Markov models.

Long and Yang [99] described a system called Log-Tracker for tracking the motion of different parts of the human body. The logs were superimposed on top of the original image, all background pixels of which were removed and marked with the value zero. Therefore, an accurate count of the number of human figure pixels represented by the log could be determined by counting the number of non-zero pixels under the superimposed log. It was also possible to measure how accurate a representation the log was for the body part in question by comparing the area of the log with the area found by the superimposition. Ribbon is another popular representation for 2D shape. Leung and Yang [3] proposed a general vision system for segmenting, tracking, and labeling of body parts from a silhouette of the human. In their work, the outline of the subject was estimated as edge regions described using 2D ribbons which were U-shaped edge segments in which a 2D stick-figure model fleshed out with ribbons was shown in *Figure 2-11*. This system demonstrated its ability to recognize poses of human performing gymnastic movements. Another ribbon-based approach for human motion analysis can also be found in [100].

Figure 2-11: A 2D stick-figure model fleshed out with ribbons.

In the work by Wren *et al.* [101;102] a set of “blobs” were used as tracking the human body. Their person finder system “*Pfinder*” [7] was a real-time system for tracking a person and interpreting his behavior in a complex environment with a fixed camera. The human body was tracked in 2D model using statistical models of the background and foreground to segment the image into blobs. In later work [102], they automatically built a behavior alphabet (a behavior was similar to a movement) and modeled each behavior using an *HMM*. They applied two *Pfinder* algorithms [7] to obtain 3D estimates of the hands and head. Therefore they used a combination of a kinematic model of human motion, a mixture of hidden Markov models and a high level classifier to investigate and understand purposeful human motion. Another example of labeling human body based on “blob” analysis is Ghost [103] developed as a module of W^4 [2;104;105] to estimate human body posture and people’s body parts from images in real time. W^4 was a real time visual surveillance system where the aim was to survey and recognize interactions between people and people or objects in an

outdoor environment. They used a standard predict–match–update scheme, where it matched predicted objects or persons with measured objects or persons in the image.

Another human detection system for visual surveillance was described by Kuno and Watanabe [106], the merit of which was to use simple shape parameters of silhouette patterns to classify humans from other moving objects such as butterflies and autonomous factory vehicles. Human walking was divided into three typical silhouette patterns: crossing, approaching and obliquely approaching. When a silhouette pattern was input, the values of the shape features were calculated and compared with the three basic pattern values to judge whether the silhouette pattern came from human or not.

There are other methods based on 2D appearance models of the human body, such as Kurakake and Nevatia [107], C. Barron and A. Kakadiaris [108;109], Kahn *et al.* [110] and Rogez *et al.* [111]. In general, 2D approaches are bottom-up approaches whose performance heavily depends on the accuracy of low-level processing. If there are missing regions or false appendages in the results of object segmentation caused by partial occlusion or motion detection errors, the 2D model driven interpretation of human body would fail. In addition, the existing 2D human model based approaches are applied mostly on single person and the background subtraction is typically used to segment the human figure. Hence usually the systems will show poor performance under image noise and gradual change of lighting scenario.

2.2.2 3-D approaches

There has been much attention in using 3D models for human motion tracking and action description for many years and several recent 3D models can be found in

[112-115]. The aim of the methods using 3D structure models is to recover 3D articulated structure of posture over time, where the postures are represented by joint angles with respect to an object-centered coordinate system. A 3D approach leads to a more accurate, compact representation of physical space which allows a better prediction and handling of occlusion and collision. It leads to meaningful features for action recognition, which are directly linked to body posture. 3D body models generally consist of two components: stick figures with a variety of degree of freedom (*DOF*) at the joints describing the skeletal structure and the geometrics, such as polygons or cylinders, describing the flesh surrounding the skeleton. The tracking of human motion with 3D models is done by predicting the parameter values for each degree of freedom, generating the perspective projection of the 3D pose model, and finally matching the projection to the moving object in real image. An important advantage of 3-D human model is the ability to handle occlusion and obtain more significant data for action analysis. However, it is restricted to impractical assumptions of simplicity regardless of the body kinematics constraints, and has high computational complexity as well.

The first articulated human model for tracking was presented by Hogg [116]. The human was modeled as an assembly of connected cylinders. For image comparison, the boundaries of the projected model limbs were compared to extracted edges in the image. The model was orthographically projected into the image. This means that the distance of the person to the image could not change very much during the sequence, since the orthographic projection assumed that the difference in camera human distance during

the tracking was much less than the distance from the human to the camera. (This made orthographic projection most suitable for surveillance purposes when the camera-human distance is large.) To address the problems of large self occlusion and a large parameters space, there were hard constraints on the motion of the model – using an analytically constructed temporal model of walking straight, the configuration of the human was basically determined by one phase parameter. Rohr [117] presented an approach similar to Hogg's work that used Kalman filters for estimating the pose between time steps. Several automatic 3D approaches developed recently are discussed in the following.

Pavlovic *et al.* [118] used a Dynamic Bayesian Networks (*DBNs*) based switching linear dynamic system (*SLDS*) model for figure motion analysis and tracking. Their kinematic model had 8 degrees of freedom (*DOF*), corresponding to rotations at the knees, hip and neck. They proposed a Viterbi approximation technique which overcame the exponential complexity of exact inference in mixed-state *DBNs*.

Gavrila and Davis [119;120] proposed an approach of 3D human model to track and recover unconstrained subject movement. The image sequences were acquired simultaneously from multiple calibrated cameras. The pose-recovery problem was formulated as a search problem for finding the pose parameters of the 3D human model. The perspective projections of the model were most similar to the appearances of the real human in the multi-view images. The 3D graphical model of a human body was represented by a 22-*DOF* (degree of freedom) model, 3 *DOF* for the human position, 3 *DOF* for the torso and 4 *DOF* for each arm and leg, as shown in *Figure 2-12*.

Figure 2-12: The recovered 3D models ELLEN and DARIU.

The kinematic skeleton was augmented with a skin composed of tapered super quadrics. A recognition cycle went as follows. Tracking commenced from a known initial configuration. In order to obtain stable edges, the people wore tight-fitting colored clothes, so the chains of tapered super quadrics were a good approximation to the external silhouette of the body. For each combination of the 22 body parameters the human model was synthesized from the cameras' point of view. Then the recovery of pose was based on a generate-and-test strategy. For each pose state, the synthesized appearance was matched with the edge image of the actual human. A robust variant of chamfer matching technique was used to generate the similarity measure between the synthesized and actual models. To search through the high dimensional pose parameter space, a decomposition approach and a best-first technique were used. When they found the best fit (highest similarity measure), the model was updated using these parameters. The recovery and tracking were initialized with a simple motion detection

and major axis determination, and then the states of torso, head, hands and feet were estimated. The system had been tested on actors in a variety of activities. Although the analysis by synthesis approach seemed to be the right one, it was still rather slow and computationally intensive.

Wachter and Nagel [121] proposed a 3D approach for quantitative geometric descriptions of the movements of a person. These descriptions were obtained by fitting the projection of a 3D person model to consecutive frames of an image sequence. The problem of fitting the person model to the image was formulated as a search problem in a high-dimensional parameter space to optimize a figure-of-merit function. In their work, both edges and regional information was applied in the matching. A person was then modeled by a set of elliptical cones with their axes as stick figures which were connected by joints. The model consisted of such body parts: head, neck, trunk, upper arms, forearms, hands, thighs, shanks, and feet. The kinematics of the model was represented by a transformation tree whose root was the person coordinate system and whose leaves were the coordinate system of the head, hands, and feet. The synthesized appearance of a posture was obtained by transforming the person coordinates into camera coordinates and perspective projection. Then the edge segments (straight lines) were extracted from the contour. An iterated extended Kalman filter (*IEKF*) was employed to estimate the covariance matrices of *DOFs*. For this purpose, the person model was augmented through a simple motion model of constant velocity for all *DOFs* that were used in the prediction step of the *IEKF*. The extended Kalman filter was a frequently used algorithm that was applied to predict the *DOFs* of joints in the next

frame. In the updated step, both region and edge information were used in the matching. The usage of Kalman filter not only yielded smooth estimation results, but also helped to predict the model more robustly. Edge matching was represented by the magnitude difference between the gray-value gradient and synthetic gradient, and the region correspondence was represented by directly matching the gray values of the points within the human body regions in consecutive images. The system had been applied on several walking persons, but the initialization of the person model was done interactively.

Delamarre and Faugeras [122] used silhouette contours to track the human body. The silhouette was the common feature space. Silhouettes were tracked using active contours that were sensitive to optical flow and image intensity. The 3-D body model (*Figure 2-13*) had 20 *DOF* which consists of spheres (neck, joints and head), parallelepipeds (hands, feet and body), and truncated cones (arms and legs).

Figure 2-13: 3D model of a person.

In order to compare the pose of the 3D model and the real Pose of the person, the 3D model was projected into the image plane and the model was adjusted to match the detected silhouette. Since active contours were used, it was possible that prior information could help drive the contour tracker. In their system, they attempted to solve the problems of depth ambiguity and occlusion by using multiple viewpoints. Constraints from all silhouettes were applied to the model, and this simultaneous set of constraints was solved by gradient decent. The claim was that occlusions were resolved by having multiple viewpoints. It was not clear how many non-occluded viewpoints were required for successful tracking. Feedback from dynamic constraints could allow this system to predict through simultaneous occlusion in all cameras.

Zheng *et al.* [123;124] build a standard model (*Figure 2-14*) with articulators and inflexible parts according to the anatomy size and proportions.

Figure 2-14: 3D model structure of a person.

Each articulator could be rotated in 3 *DOFs* either automatically or manually. The inflexible parts such as limbs and trunk between articulations were built at one of

the following levels: the simplest level consisted of boxes and cylinders; a little complicated level was composed of many ellipses describing bones and muscles; the complete one was obtained from a real human using a laser range finder. Surface shape and color were obtained and organized in parts. All of the body parts could be scaled in their size and proportions in an interactive way. Hence, the model not only had a moving function, but also had changeable shape, color and size. Another 3D model approach could be found in [125;126]. In their work, Sidenbladh *et al.* modeled the human body as configuration of 9 cylinders and 3 spheres with 25 *DOFs*, as shown in *Figure 2-15*. In addition, a 3D configuration of a moving object can be obtained by its 2D image projection. Chen *et al.* [127] recovered this 3D model according to the 2D image projection by modeling human with 17 line segments and 14 joints to denote the head, torso, hip, arms and legs.

*Figure 2-15: 3D model structure of human body. Each limb, i has a local coordinate system with the Z_i axis directed along the limb. Joints have up to 3 angular *DOF*,*

expressed as rotations $(\theta_x, \theta_y, \theta_z)$.

All cylinders were right circular, except for the torso that had an elliptical cross section. Each part was defined in a part centric coordinate frame with the origin at the base of the cylinder or sphere, and was connected to others at joints, the angels of which were represented as Euler angles. The generative model of human appearance had three main components: shape, appearance, and motion.

2.2.3 2-D approaches without explicit shape models

This approach describes human movement according to simple low-level, 2D features from a region of interest without the explicit recovery of shape, structure, and parts of human body. This means that these approaches use sub-features such as distinguishable points or lines on the object to investigate human behavior for bottom up estimation of whole-body 2D pose in individual frames or sequences. More and more real-time systems make use of the general features of motion, since the approach without explicit shape models does not require the recognition or tracking of specific parts of the human. It emphasizes the recognition rather than the modeling of temporal patterns. There are usually two steps for the 2-D approaches without explicit shape model: the first is feature extraction, and the second is feature matching over a sequence of images. The criteria for choosing a good feature are its robustness to noise, brightness, contrast, and size. Its benefit is that even in the presence of partial occlusion, some of the sub-features of the tracked objects remain visible. Some of the developed methods are described in the following.

Oren [128] performed object detection in static images. They segmented pedestrians in a number of images and generate a common template based on Haar

wavelets template, which could be used to encode an object as a set of regions and relationships among them. The template was obtained by applying a differential operator at various locations, scales, and orientations on the image grid of interest. In a training stage, all the example images were scaled and clipped to form a training set. A small subset of coefficients was selected to represent a desired object, based on considerations regarding relative strength and positional spread over the images of the training set. Once it had been established which wavelet coefficients to use as features, a support vector machine (*SVM*) classifier was applied to the training set. During the detection stage, one shifted windows of various sizes over the image, extracted the selected features, and applied the *SVM* classifier to verify whether the desired object was present or not. In runtime the template was compared to various parts of an image to find pedestrians. This technique could detect frontal and rear views of pedestrians in static of cluttered scenes, but the misdetection rate was also quite high.

Polana and Nelson [129-131] referred to “getting your man without finding his body parts”. Models for human action were described in statistical terms based on low level features or by simple heuristics. The approach had been especially popular for applications of repetitive motion activities, such as walking, running or climbing. In their work, people could be bounded by a rectangular box, and the centroid of the box was chosen as the feature point for tracking, which would be utilized to recognize human action. They used flow information and down sampling to represent image information and the repetitive motion activity could be accomplished on the basis of bottom up processing, which did not require the prior identification of specific parts.

First, the motion information was obtained by computing the normal flow between successive image frames using a simple differential technique, and each flow frame was decomposed into a spatial grid in both horizontal and vertical directions. The grid formed an element of the feature vector that described the state of movement at time t . Then, the spatial scale of the tracked object was normalized with a simple perspective analysis. Finally, the nearest-neighbor centroid algorithm was exploited to realize human action recognition. The classification and recognition of the activity was achieved by matching the resulting feature vector against reference motion templates of known classes. In addition, this approach could also deal with the more complex case of a moving camera and/or multiple (overlapping) objects, based on detecting and tracking independently moving objects.

In [55], Polana and Nelson further improved their method by extracting motion features from the reference curves that were lines parallel to the trajectory of the moving flow centroid. The spectral power was estimated for the image signals along these curves. In addition, this technique was view dependent, and since optical flow was used, it would be highly susceptible to image noise. Takahashi *et al.* [132] used an average edge vector for each tile, while Yamato *et al.* [133] used the number of foreground pixels in each tile.

The method proposed by Cutler and Davis [13;134;135] was somehow similar to that of Polana and Nelson. They described new techniques to detect and analyze periodic motion as seen from both a static and moving camera. By tracking a moving object of interest, they computed its self-similarity as it evolved over time. For periodic

motion, its self-similarity measure was also periodic. Therefore they applied time–frequency analysis to detect and characterize the periodic motion, and implemented tracking and classification moving objects using specific periodic patterns in the similarity plot, a 2D matrix of the absolute correlation between each pair images of the person in the sequence. In their work, there were two main parts, one was motion segmentation and tracking, and the other was periodicity detection and analysis. Motion regions were extracted by using background subtraction method. Then they were tracked by their size and position history, or by the image template matching. The tracked object was aligned using its centroid based on the tracking results and normalized in scale using the object’s position and a ground plane model. In the following step, Fourier analysis was applied to detect and characterize the periodicity. Then an image similarity metric between the aligned and normalized frames at time t_1 and t_2 was defined as

$$S(t_1, t_2) = \min_{|dx, dy| \leq r} \sum_{(x, y) \in B_{O_{t_1}}} |O_{t_1}(x + dx, y + dy) - O_{t_2}(x, y)| \quad (2-3)$$

where O represented foreground object and corresponding O_{t_1} denoted the object at time t_1 . In addition, $B_{O_{t_1}}$ was the bounding box of the object at t_1 . Then the 1D power spectrum of $S(t_1, t_2)$ for a fixed t_1 and all values of t_2 were calculated and the frequency of the periodicity was determined by extracting a peak in this spectra. The Fisher’s test was used to extract the fundamental frequency if the periodic data was stationary with white noise. For non-stationary periodicity, the matched peaks in the autocorrelation of the self-similarity $S(t_1, t_2)$ were used to detect the periodic motions.

Since the calculations of 2D self-similarity and the 2D autocorrelation were quite expensive, this technique was applied to small objects with low resolution to get a real-time performance.

Statistical shape model is another cue that can also be used to detect and track the contours of hands or persons. Tracking based on active contour models aims at directly extracting the shape of the subjects. This type of active contours is also called snakes. The idea is to have a representation of the bounding contour of the object and keep updating it over time dynamically. Active Shape Models (*ASM*) [136] and Active Appearance Models (*AAM*) [137], proposed by Cootes *et al.* were two popular models for the purpose of shape and appearance modeling and extraction. In *ASM*, Cootes *et al.* presented the shape models in terms of a training stage, in which shapes were described by the known feature point locations. Principal component Analysis was used to these feature points to obtain the representation of the training shapes using a reduced parameter set. According to this representation, some degree of generalization would be obtained to track the deformable shapes. While *AAM* was used to combine the constraints on both shape and texture in its characterization of face appearance. Based on the accurate modeling of the local features, *ASM* had better accuracy in shape localization and *AAM* had lower texture reconstruction error. The method proposed by Baumberg and Hogg [138] was another application of active shape model to the tracking of pedestrians. They tracked the outline of a moving body using a model-based flexible shape model that was based on B-splines. Then the shape model captured the considerable outline variations in the human silhouette during movement.

The image of front or rear views of a walking pedestrian were tested in their work.

Chomat *et al.* [139] employed principle component analysis (PCA) based spatial-temporal filters for analysis and recognition of human motion. The local appearance of the contents of spatio-temporal neighborhoods from a set of sequences was used for describing the appearance of human motion. Finally a Bayes classifier based on multidimensional histogram estimation was introduced to perform action selection. However the problem of recognizing the different actions in the feature space is not solved. Similarly Laptev *et al.* [140] extended the notion of spatial interest points into spatio-temporal domain and estimated a scale-invariant spatio-temporal descriptors for human action and events representation, in which they built on the idea of the Harris [141] interest point operators and then detected local structures in space-time. Finally, this representation was employed in detecting human walking in moving background. However the robustness of the approach was questionable. Later Laptev *et al.* [142] investigated local space-time descriptors for representing motion patterns. They compared motion representations in terms of spatio-temporal jets, position dependent histograms, position independent histograms, and principal component analysis computed for either spatio-temporal gradients or optic flow. Finally they concluded that in the combination of feature and measurement, position dependent histograms of optical flow with a Euclidean distance measure was the best for spatio-temporal action recognition, and the second best one was position dependent histogram of spatial-temporal gradients in combination with the normalized scalar product. Some other spatial scale invariant detectors includes:

Harris-Laplacian [143], *SIFT* [144].

Bobick and Davis proposed a real-time method for representing and recognizing simple human movements using temporal templates in [145-148]. They made use of the binary motion-energy-image (*MEI*) and motion-history-image (*MHI*) to interpret human movement in an image sequence. In their approach, *MEI*, a binary cumulative image, was used to represent spatial patterns of motion and *MHI*, a scalar-valued image, was used to describe a particular region that had just been swept out for the duration. First, motion images in a sequence were extracted by differencing, and these motion images were accumulated in time to form *MEI*. Since the *MEI* captures the spatial distribution of motion, its shape represented some important features of motion pattern. Then, the *MEI* was enhanced into *MHI*, which was a scalar-valued image. In the scalar-valued *MHI* image, more recently moving pixels were brighter. Taken together, the *MEI* and *MHI* could be considered as a two-component version of a temporal template, a vector-valued image in which each component of each pixel was some function of the motion at that pixel position. Finally, to recognize human actions, the seven Hu-moments were employed to describe the *MEI* and the *MHI* images. Hu-moments were a kind of region moment representation that interprets a normalized gray-level image function as a probability density of a 2D random variable. The training examples of each movement from a variety of view angles were collected. For each view of each movement, a statistical model was generated for shape moment parameters of both the *MEI* and the *MHI*. Then these view-specific representations were matched against the stored models of views of known actions. A Mahalanobis

distance could be used to classify the action of the object by comparing moments to the Hu-moments of prerecorded actions. There are also some extension applications of *MEI* and *MHI*. For example, *SMIs*, inspired by *MEHI* and *MHI*, are employed to the EyesWeb to open software platform [149;150]. The *SMIs* differed from *MEIs* that the silhouette in the last (more recent) frame is removed from the output image.

In addition, for 2D posture estimation approach, body part detectors (face, hands, arms, torso and legs) instead of whole 2D pose are analyzed. Such kinds of human body parts can be detected by employing 2D shape [151], *SVM* classifiers [152], AdaBoost [153], and locally initialized appearance models [154]. Mikolajczyk et al. [76] proposed a probabilistic method to model humans as flexible combinations of boosted face, torso and leg detectors. Parts are represented by a joint probabilistic body model. In [153] this approach is extended by Micilotta to detected human body parts into a full 2D human configuration, in which body configurations are assembled from the detected parts using *RANSAC*. Furthermore, the human body part are mainly used for Part Based Human Tracking [155].

There was also a lot of work to segment the images sequence into regions of consistent motions, such as Costeira and Kandade [156] proposed an approach for separating and recovering motion and shape of multiple independently moving objects without prior knowledge of the number of object. Irani et al. had done a serial of works [157-160] about motion factorization. In 2000, Irani et al. [157] analyzed factorization with uncertainty, which dealt with the correlated positional noise. Furthermore, Machline et al. [158] suggested a new definition of motion consistency

based on temporal consistency of behaviors across multiple frames and proposed a multi-body segmentation algorithm without requiring prior correspondence estimation. Zelnik-Manor *et al.* [160] further extended the method of spatial factorization to temporal factorization.

2.2.4 Discussion

From the above literature survey, it can be observed that 2D model based techniques may be more efficiently implemented in real-time than 3D model. The disadvantage of the 2D model is that the feature representation is difficult to be understood by people while the structure-based 3D methods can provide a natural representation of human motion, but the restoration and tracking of the 3D structure of human body and motion require extensive computational resources. In general, human activity can be broken down into a sequence of poses of a human subject. Hence, through recognizing a sequence of static configurations, human behaviors can be understood. Due to the high computational complexity of constructing 3D models, the proposed approach is based on 2D without explicit shape models approach.

The background difference has become general method of motion segmentation, while temporal difference attracts few attentions on motion segmentation field. Compared with temporal difference method, background difference need more processing, such as background estimation and maintenance. Generally speaking, reference background is estimated with the model parameters during a short period under the assumption that no foreground objects occurred during the training period. Moreover, the background scene and the camera are required to be stationary. The

procedure of background difference become more complex or even impossible for environments like outdoor, or room with large window facing outdoor with frequent light changing, or mobile surveillance with camera mounted on a mobile robot since background difference need time to do initialization whenever the robot stops and does monitoring. In addition, as shown from our experimental results, our system needs only two frames to recover and perform correctly. Compared with other background subtraction systems, the proposed approach needs much less time to recover. For background subtraction approach, the restoration of background model from global changes depends on learning rate. For example, the system proposed by Li and Leung [4] can takes more than sixty frames to restore after lighting change. This illustrates one important advantage of the temporal difference.

For temporal difference based method, it is suitable and convenient to be set up at non-predetermined scenes (mobile surveillance) with complex background. It is faster than optical flow method and can be applied in real time. Although some demos of optical flow based methods show the ability to work in real time, there are still some native limitations for optical flow based method. According to the smoothness constraint, the movement of corresponding points in the two continuous frames should not be more than a few pixels. For some uncertain scenarios, this will limit the camera motion or background be slowly changed. Hence temporal difference is attractive to explore in our work.

Chapter 3 System Overview

In this chapter, we outline a prototype human body posture recovery system. The system flowchart is proposed in *Figure 3-1*. In our work, we define the human posture reconstruction problem to be a combination of two major components: motion based human body acquisition and human posture computation. Motion based human body acquisition aims to segment the human body that correspond to the movement. Human posture computation aims to reconstruct and configure a posture such that it accurately reflects the position and the configuration of the human body after the motion capture process. Moreover, we propose motion pattern analysis to exact region description for image data understanding and classification.

In the proposed system, human body acquisition consists of two sequential processes that are pre-processing and moving object outline, while human posture computation consists of two sequential processes that are convex shape modeling and convex shape combination as illustrated in *Figure 3-1*. In addition, human body segmentation and motion pattern analysis share some common procedure or activities such as they both use temporal difference based method to extract motion information. The human body segmentation results are then fed to the human posture computation process. Each of these five processes will be briefly described in the following.

Figure 3-1: Flowchart of human body modeling.

3.1 Pre-processing

Firstly, the system operates on raw input images and then extracts necessary information for human posture module. The pre-processing comprises two steps, i.e. edge detection and edge thinning. Each of these processes will be described in the following.

The edge detection problem is posed as detecting discontinuities in the image

intensity. The advantage of edge detection is that it can achieve data reduction and preserve the important structural properties of an image. This technique has been studied extensively in the work proposed by Heath et al. [161], in which they had investigated the performance of different edge detectors by comparing the edge detectors based on experimental psychology and statistics. One of their findings is that “No one single edge detector was best overall; for any given image it is difficult to predict which edge detector will be best”. Actually, in our work, the proposed approach does not rely on any specific edge detector. Any edge detector can be used and the implementation from Nevatia and Babu [162] is utilized to convert a 256 gray scale image to binary one. One example is illustrated in *Figure 3-2(b)*. The white pixels represent information, while the black represents the background.

After edge detection, a raw image can be reduced to the edge representation. In digital geometry, a thinning process can be employed to transform edges into a simplified, but topologically equivalent image. Based on this mechanism, thick edges can be reduced as chains of single pixel that can be easily traversed and represented (*Figure 3-2 (c)*).

Figure 3-2: An illustration of extracted edges; (a) original intensity image, (b) edge image, (c) thinned edges.

3.2 Moving Object Outline

Secondly, as a human moves and attains new postures, its apparent contour changes dynamically as the result of the motion of the human body. To represent motion and shape evolution in a compact and intuitive way, we introduce a new shape representation named moving object outline based on temporal difference approach which will be discussed in *chapter 4*. One example output is shown in *Figure 3-3 (b)*.

To make moving outline more computable for following processes, the moving object outline image can be reduced to its *Line Edge Map (LEM)* [163]. *LEM* is proposed here to integrate the structural information with spatial information of moving object outline image by grouping pixels of moving object edge map to line segments. The *LEM* representation, which records only the end points of line segments on curves, reduces the storage requirement. Efficient coding of moving object outline is one important aspect in dynamic programming based human body modeling system. As the basic units of *LEM* are the line segments grouped from pixels of edge map, we explore the information of *LEM* and investigate the feasibility and efficiency of convex shape modeling using *LEM*. The process to generate the *LEM* is described in *chapter 4*. One example of moving outline smoothing output is shown in *Figure 3-3 (c)*.

Figure 3-3: Human body modeling processing; (a) original frame, (b) MO result, (c) MO smoothing result, (d) human posture result.

3.3 Motion Pattern Analysis

Then, we generate the front/rear motion history images ($rMHI$ and $fMHI$) and motion vectors along the moving object boundary to measure motion and distinguish which side of outline belongs to moving region. The $rMHI$ is generated from a sequence of motion layers that indicates regions of motion, in which $rMHI$ region corresponds to a region where the moving object has just swept out at the most recent time period. The $fMHI$ is generated by shifting the motion layers to the front with respect to the motion direction and $fMHI$ can grab better motion cue from the rising front of the human body compared with $rMHI$. One group of examples of motion history images and motion vectors are shown in Figure 3-4.

Figure 3-4: Example outputs of motion history images with motion vector for a running action; (a) original frame, (b) three-layer fMHI, (c) three-layer rMHI, (d) motion vectors (with directions from black to white) overlaid on the original frame.

In addition, Sedimentary Moving object Outline (*SMO*) for representing history mechanism is employed to record *MO* that stop moving. Hence, deposited objects can be discovered and tracked. One example of *SMO* is shown in *Figure 3-5*. There are four images below, in which, image (a) illustrates the background environment and image (b) to (d) are the segmentation results with sedimentary *MO*. Current *MO* is shown in black while *SMO* is drawn with grey color.

Figure 3-5: SMO detection from deposited object; (a) original frame, (b)-(d) SMO segmentation results.

To track movement of overlapping body parts, inner *MHI* (*iMHI*) is proposed to isolate the mixed moving parts from each other and then track the movement of each isolated part. Then the *iMHI* can be applied to generate motion feature for each tracked moving part. One *iMHI* extraction example is shown in Figure 3-6.

Figure 3-6: Inner motion extraction; (a) original frame, (b) rMHI, (c) iMHI.

3.4 Convex Shape Modeling

Thirdly, convex shape modeling is one critical component of human body model

system. From previous two processes, we obtain the moving object outline based *LEM*. Then we apply the dynamic programming technique on *LEM* to obtain convex shapes that can be the candidatures for human head and torso with arbitrary pose and viewpoint. In addition, prediction and angle constraints of convex shapes' lines are introduced to reduce the complexity of the convex shape generation process. Extensive experiments demonstrate that the system can robustly and effectively generate convex shapes that can be employed in the process of human body modeling. Some example outputs of convex shapes are shown in *Figure 3-7* in black color.

3.5 Convex Shapes Combination

The final process of proposed system is convex shape combination. It can automatically acquire the human body posture by combining convex shapes from previous process. The system suggests a body part localization procedure to determine the convex shape positions according to principles and the topology of the estimated body posture under the assumption that head-torso model is a stable body part combination compared to only one body part such as head, torso or limbs. The details of this process will be discussed in *chapter 7*. One example output is shown in *Figure 3-3 (d)*, where the human body parts are differentiated by lines with different widths.

Figure 3-7: Examples of convex shapes detection; (a) original frame, (b)-(i) outputs of convex shapes.

Chapter 4 Moving Outline Generation

A common problem in motion analysis is “how to obtain and represent comprehensive information about moving objects in a scene”. Johansson [9] proposed that the actions of a person might be recognized using only information from motion in his famous classic *Moving Light Displays (MLD)* experiment on human perception. However, the requirement to attach lights or markers to a person is not practical. Based on this concept, the analysis of human behaviors should first tackle the problems of moving object segmentation and then motion perception. In this chapter, a novel, effective and efficient method for moving outlines generation is proposed to segment moving objects. The proposed is motivated by the psychological study [164] that human can recognize objects from their *LEM* almost as good as the original image. The movement vector of each outline will be derived in next chapter for motion representation.

The table of symbols and abbreviations, which will be used in this chapter, is shown in *Table 4-1*.

Table of symbols and abbreviations	
<i>EI</i>	<i>Edge Image</i>
<i>MO</i>	<i>Moving object Outline</i>
<i>frontMO</i>	<i>front Moving object Outline</i>
<i>rearMO</i>	<i>rear Moving object Outline</i>
<i>TDE</i>	<i>Temporal Difference Edge</i>
<i>TDO</i>	<i>Temporal Difference Outline</i>
<i>ROI</i>	<i>Region Of Interest</i>

Table 4-1: Symbols and abbreviations.

In this chapter, the implementation of the block "Moving Object Outline" shown in the flowchart of *Figure 3-1* will be described. As shown in *Figure 3-1*, the inputs are original images and the images after pre-processing, in which such examples can be found in *Figures 3-2 (a)* and *(b)*. The output of the block "Moving Object Outline" is *MO* segmentation as shown in *Figure 3-3 (b)*.

4.1 Process Overview

In this work, we propose segmentation of "Moving object Outline" (*MO*) rather than "edges of moving objects" since we can apply more restrictions on the former to get genuine moving line segments. An example of edge detection result of original image and *MO* overlaid on the real image is shown in *Figure 4-1*.

A stationary monochrome camera is employed. We compute the *temporal*

difference only. It takes difference of two consecutive frames and difference of the edge images followed by thresholding to extract outlines of moving regions. *Temporal difference* has many advantages. Compared to *background subtraction*, it works without referring to the background image and hence it does not need to spend valuable resources to update the background. Consequently, it is suitable and convenient to be set up at non-predetermined scenes with complex background. It is faster than *optical flow* method and can be applied in real time. In addition, it has good adaptation to lighting changes as compared to the background subtraction method.

Figure 4-1: MO segmentation result of walking activity: (a) original frame, (b) edge detection result overlaid on original frame, (c) MO overlaid on original frame, (d) edge detection result, (e) MO.

However, there are also disadvantages in employing *temporal difference*. First, the regions generated by temporal difference are just the approximate moving regions instead of the real moving objects. It generally does a poor job in extracting all the

relevant pixels, i.e. there may be holes left inside moving entities, and the detection results will either be fatter or slimmer than the true moving objects (see examples shown later in the chapter). These defects have little effect on our system since we segment moving outlines rather than moving regions.

The second problem is that it cannot distinguish which side of the outline is the moving region. To tackle this problem, we first use edge image (*EI*) and outline of moving object to generate primary moving outline (*MO*) to isolate current motion edges from pervious frame. Then, we generate the front/rear *motion history images* [1] and motion vectors along the moving object boundary to measure motion and distinguish which side of the outline that the moving regions are. This will be discussed in *chapter 5*. The results will be *MO* and motion vectors detected from each frame using only temporal difference. Finally, this scheme cannot detect non-moving or stop moving objects. However, we can classify them as non-interesting objects. Furthermore, we employ a history mechanism to record *MO* that stop moving. Hence, deposited objects can be discovered and tracked. It is found that the motion activities can be perceived by human observers easily when the segmentation results are displayed consecutively, similar to the results obtained from Johansson's work. The general framework for the process is shown in *Figure 4-2*. It is found that temporal difference can work as good as the background difference from the experimental results. It can be a good complement with respect to the background subtraction method. This is one of the major contributions of this work.

The moving outlines (*MO*) of objects in current frame are detected using 2

complementary processes along the left and right paths of *Figure 4-2*. The left process computes temporal difference edge (*TDE*) by taking temporal difference of the edge images while the right process takes temporal difference of the intensity images to generate temporal difference outline (*TDO*). Each of these processes will be described in detail in the following.

*Figure 4-2: Flowchart of the proposed system; **e**: edge operator, **+**: combination, **−**: temporal difference operation, **I**: input images, **EI**: edge image, **MO**: moving outline, **TDE**: temporal difference edge, **TDO**: temporal difference outline.*

4.2 Temporal Difference Edge

4.2.1 Image Loading

For our current work, the raw input images are in the *Tagged Image File Format* (*TIFF*). A publicly available library named "libtiff" [165] is used. This library provides the primitive functionalities for *TIFF* file loading and saving.

4.2.2 Temporal Difference Edge

Temporal difference edge (*TDE*) is first generated by subtracting the edge images

(EI) [162] of two consecutive frames from each other as $|EI(x, y, k) - EI(x, y, k-1)|$.

By doing subtraction (along the left path of *Figure 4-2*), most stationary background edges can be eliminated (see *Figure 4-3 (e)* and *Figure 4-4 (e)*). The remaining edges are mainly the moving edges from previous ($k-1$) and current (k) frames, plus some edges that belong to the stationary background which disappear and reappear due to object movements. To filter out the edges from previous frames, the subtraction result is further refined by intersecting it with $EI(k)$. The combined effect of subtraction and intersection gives *equation (4-1)*.

$$TDE_{k-1}^k(x, y) = \begin{cases} EI(x, y, k) & \text{if } |EI(x, y, k) - EI(x, y, k-1)| > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4-1)$$

where x and y are the coordinates, k is the time, and EI is computed in *section 3.1*. Two example results are shown in *Figure 4-3 (f)* and *Figure 4-4 (f)*. The result is the combination of foreground edges and background edges which reappear in current frame due to object movement.

4.3 Temporal Difference Outline

Since TDE can include stationary background edges, we will intersect it with the temporal difference outlines of objects to eliminate most if not all such edges. These outlines are generated here by computing the temporal difference of two consecutive raw images, $I(k)$ and $I(k-1)$, and then extracting the edges as (see *Figure 4-3 (g), (h)* and *Figure 4-4 (g), (h)*):

$$ROI_{k-1}^k(x, y) = \begin{cases} 1 & \text{if } |I(x, y, k) - I(x, y, k-1)| > T(k) \\ 0 & \text{otherwise} \end{cases} \quad (4-2)$$

$$TDO_{k-1}^k(x, y) = ROI_{k-1}^k(x, y) * e(x, y) \quad (4-3)$$

where TDO , ROI , T , e and $*$ are the temporal difference outline, region of interest, a dynamic threshold (to be discussed below), an edge operator and the convolution operation respectively. Two examples of outputs from the running and walking sequences are shown in *Figure 4-3* and *Figure 4-4*.

In our work, we consider region of interest as moving object. ROI segmentation simply divides an image into regions of potential moving objects and stationary background. In general, we have no idea of the sizes, positions and number of ROI s. However it is clear that the number of ROI s should be steady over a wide range of threshold values. This approach was originated from the work by O’Gorman [166] for intensity image threshold. The conclusion of their method is that if the number of regions is stable over a range of threshold values, it is reasonable to choose one value in this range to be a suitable threshold α . Some other adaptive threshold selection methods can be found in [62;63].

The approach we use here computes the numbers (y) of regions (eight connected components) generated from different threshold values (t), as shown in *Figure 4-5* which shows one example of the number of regions over different threshold values. Note that if the threshold t is set to be 5, the number of regions will be extremely large. It drops to one when t is decreased to 3. These lower-end threshold values are not needed to generate a suitable threshold here. Hence, we consider the curve to the right only starting from $t=7$ (see *Figure 4-5(b)*).

(a) $I(k-1)$	(b) $I(k)$	(c) $EI(k-1)$

(d) $EI(k)$	(e) $ EI(k) - EI(k-1) $	(f) TDE_{k-1}^k

(g) ROI_{k-1}^k	(h) TDO_{k-1}^k	(i) ROI_{k+1}^k

(j) TDO_{k+1}^k	(k) $frontMO_{k-1}^k$	(l) $rearMO_{k+1}^k$

(m) $frontMO_{k-1}^k \cup rearMO_{k+1}^k$	(n) $MO_k^2(x, y)$	(o) MO

Figure 4-3: Differencing outputs of a running sequence.

(a) $I(k-1)$	(b) $I(k)$	(c) $EI(k-1)$

(d) $EI(k)$	(e) $ EI(k) - EI(k-1) $	(f) TDE_{k-1}^k

(g) ROI_{k-1}^k	(h) TDO_{k-1}^k	(i) ROI_{k+1}^k

(j) TDO_{k+1}^k	(k) $frontMO_{k-1}^k$	(l) $rearMO_{k+1}^k$

(m) $frontMO_{k-1}^k \cup rearMO_{k+1}^k$	(n) $MO_k^2(x, y)$	(o) MO

Figure 4-4: Differencing outputs of a walking sequence with occlusion.

From *Figure 4-5*, it is observed that at low threshold values, there are many regions. However, most regions are the products of noise. In contrast, at high threshold values, a stable number of regions can be reached. These regions correspond to the moving objects. This observation suggests that a reasonable threshold may lie at the “corner” or “elbow” of the curve. Instead of detecting the corner, a simplified approach according to the points near the corner is adopted to find the first α (from left to right) which satisfies:

$$\alpha = \frac{y(t)}{y(t+1)} \leq \theta$$

where θ is empirically set as 1.2. The corresponding t will be the threshold $T(k)$ in *equation (4-2)*. ROIs are further refined by eliminating thresholded regions with area less than 20 pixels.

Figure 4-5: number of regions versus different threshold values; (a) original curve, (b) the curve starting from $t=7$ after magnification.

4.4 Moving Outline

TDE provides the potential moving edges in current frame while *TDO* provides the outlines of moving regions from previous and current frames. The intersection of

these two should then give MO from current frame. However, it is observed that only half of the required MO can usually be obtained from the intersection since ROI is a mixture of moving regions from previous and current frames merging together. Hence half of the outlines of moving objects in current frame are located within $ROIs$ as shown in *Figure 4-3 (k)*. The MO obtained can be called *frontMO* because they are at the front according to the movement direction. It is computed as:

$$frontMO_{k-1}^k(x, y) = TDO_{k-1}^k(x, y) \cap TDE_{k-1}^k(x, y) \quad (4-4)$$

The occluded MO can be called *rearMO* since they are at the rear according to the movement direction. They can be computed similarly if time can be reversed from $k+1$ to k as:

$$rearMO_{k+1}^k(x, y) = TDO_{k+1}^k(x, y) \cap TDE_{k+1}^k(x, y) \quad (4-5)$$

Finally, the first part of MO , MO^1 is defined as the union of these two as:

$$MO_k^1(x, y) = frontMO_{k-1}^k(x, y) \cup rearMO_{k+1}^k(x, y) \quad (4-6)$$

Two examples of the computed *frontMO* and *rearMO* are shown in *Figure 4-3 (k)* and *(l)*.

4.4.1 Complement

Since MO^1 depends heavily on edge detection results, it will fail at location when edge cannot be extracted though human can perceive ROI and MO from motion. One example is shown in *Figure 4-4 (f)* where some MO of the moving arm (on the right) cannot be extracted though the arm can be found from ROI and TDO (see *Figure 4-4 (g)*,

(*h*)) because the arm region is different from the background.

Based on *TDO*, one can complement the lost *MO* by *TDO*. From *equation (4-4)* and (4-5), we compute

$$frontMO_{k-1}^k(x, y) = TDO_{k-1}^k(x, y) \cap TDE_{k-1}^k(x, y)$$

$$rearMO_{k+1}^k(x, y) = TDO_{k+1}^k(x, y) \cap TDE_{k+1}^k(x, y)$$

These two equations can be complemented by substituting *TDE* by *TDO* as

$$frontMO_{k-1}^{k'}(x, y) = TDO_{k-1}^k(x, y) \cap TDO_{k-2}^k(x, y) \quad (4-7)$$

$$rearMO_{k+1}^{k'}(x, y) = TDO_{k+1}^k(x, y) \cap TDO_{k+2}^k(x, y) \quad (4-8)$$

Since $TDO_{k-1}^k(x, y)$ and $TDO_{k-2}^k(x, y)$ in (4-7) have the same front moving boundary and different back moving boundary, the intersection of these two can be used to retrieve the lost *frontMO*. Similar argument applies to *equation (4-8)* too. Hence, we can have a complement as

$$MO_k^2(x, y) = frontMO_{k-1}^{k'}(x, y) \cup rearMO_{k+1}^{k'}(x, y) \quad (4-9)$$

It is observed that $MO_k^2(x, y)$ can miss more outline information than $MO_k^1(x, y)$ since each of $frontMO_{k-1}^{k'}(x, y)$ and $rearMO_{k+1}^{k'}(x, y)$ has to depend on a region segmentation process (i.e. *TDO* is derived from *ROI*). Unfortunately, region segmentation is not reliable in general. However, the combination of $MO_k^1(x, y)$ and $MO_k^2(x, y)$ can complement each other as (see *Figure 4-4 (o)*). For another example, from *Figure 4-3 (m)*, (*n*) and (*o*), we observe that *Figure (m)* can capture human left arm and right leg, while *Figure (n)* detects the right arm, left leg and head quite well.

Hence *Figures (m)* and *(n)* are complement and the integration can provide more reliable *MO* results.

$$MO_k(x, y) = MO_k^1(x, y) \cup MO_k^2(x, y) \quad (4-10)$$

4.5 Moving Outline Representation

4.5.1 Contour Extraction

Contour is a compact way to represent the *MO* (moving outline). It records sequential lists of consecutive pixel locations which can be represented by integer coordinates. Given a sequence of consecutive coordinate points $p_1(x_1, y_1)$, $p_2(x_2, y_2)$, ..., $p_i(x_i, y_i)$, $p_{i+1}(x_{i+1}, y_{i+1})$, ..., $p_n(x_n, y_n)$, where p_i is connected to p_{i+1} , $1 \leq i < n$, we have $|x_i - x_{i+1}|$ and $|y_i - y_{i+1}|$ both less than or equal to one but not both equal to zero [167].

4.5.2 Dynamic Two-strip Algorithm (Dyn2S)

The polygonal approximation attempts to find polygons that can best fit a contour, based on certain estimate of appropriateness. Here, polygonal Approximation is accomplished by the Dynamic two-strip algorithm (Dyn 2S) [168].

Dyn2S is a curve representation method using feature points which this work is based on due to their flexibility in computing the support for a feature point. The whole procedure has two stages. In the first stage, important properties of the data points are derived. Two strips are fitted to the left and right sides of each point on the curve respectively and the points inside each strip are approximated as the left or right straight line. The best fitted strips are derived by adjusting the width of the strip and rotate it up

or down dynamically using the starting point as the pivot. Longer and narrower strips are preferred. The ratio, which is called the elongate value of a strip, is defined as $E = L/W$ of its length L over its width W to represent the merit of fit. The merit f for a point can be computed as

$$f = E^{left} \cdot S^{\theta} \cdot E^{right} \quad (4-11)$$

where E^{left} and E^{right} are the elongate values of the left and right strips, and $S^{\theta} = |180^{\circ} - \theta|$ is the acuteness of the angle θ between the two strips. In the second stage, a subset of the most representational points is selected based on the local maxima of point merits.

One example is illustrated in *Figure 4-6* where the minimum and maximum strip widths can be easily pre-selected as one and $0.1 \times D$ pixels, where 1 is considered small enough while D is the length of the image diagonal. Let P_0 be the current point to be processed, L_i ($i = 1, 2, \dots, 6$) be the set of points on P_0 's left side and R_j ($j = 1, 2, 3$) be points on P_0 's right side. An initial strip of minimal width would first extend from P_0 to encompass as many points as possible to its left side (i.e. the dashed-line box). Its width is then increased and its orientation is adjusted to include as many points as possible, until the maximum strip width is reached (i.e. the left solid-line box).

Figure 4-6: Illustration of the dynamic two-strip algorithm with the selection of minimum and maximum strip widths.

The measure of fit (i.e. elongate value) is recorded for each strip as a function of strip width but only the strip with the largest elongate value is marked and used. The same procedure is applied on the other side of the feature point.

There are advantages using strips to extract local features. First, digitization noise can be tolerated because the strip has width and it can enclose points that can be approximated as a straight line. Then the support region of one feature point can be reflected honestly by the strips.

In the Dyn2S algorithm, a strip is fitted to the left and right of each point on the curve as shown in Figure 4-7. The merit of a point can be computed as $f = E^{left} \cdot S^{\theta} \cdot E^{right}$, which is already described in equation 4-11. A group of significant points is selected using local maximum detection process based on the merits. A straight line is generated by connecting two significant points, which are the neighbor of each other on the curve. Such representation of line segments is named as the line edge map (*LEM*).

Figure 4-7: Illustration of the dynamic two-strip algorithm: An initial strip (the dashed line) of minimal width extends from P to encompass as many points as possible to one side.

After fitting all the contours by polygons, the gray scale image is reduced to its Line Edge Map, another example of the representation is shown in Figure 4-8.

Figure 4-8: An example of Line Edge Map. (a) original intensity image, (b) outline of moving object, (c) line edge map representation.

4.6 Experimental Results

The proposed approach has been implemented using the C++ language running on a Pentium IV system with a Meteor digitizer. The system is able to run at about 7 frames/sec with a resolution of 192×144 pixels, while the image sequences were grabbed at 15 frames per second. In this section, the method is evaluated.

Four different test scenes were used. The first test set (45 sequences) was used in [4]. It is a scene of an open area in a large laboratory with activities like walking,

making phone call, collecting printed documents, using a computer, moving chairs, etc. There are also fast activities like simulated fighting, knocking down a person, battering, kicking objects, falling down, etc. The second test (10 sequences) set captured activities in a large laboratory in dim light environments and the third test set (10 sequences) captured activities in a large laboratory in light changing. The last test set captured activities in two kinds of environment. One is inside an elevator (11 sequences) and the other is TV collection of robbery scenes captured by close-circuit cameras (5 sequences). The results are very encouraging and human observers can recognize most activities from just the segmented moving outlines. The experimental results on primitive extraction are illustrated from *Figure 4-9* to *Figure 4-18*.

Test set 1

In this test set, different types of actions have been performed with different views and distances from camera. The actions include multiple people walking together along the optical-axis, running from left to right, squatting down from up to down and object-throwing action from top right corner to lower left corner.

In this test set, the first example captures one guy squatting down. It starts from frame #3781 to frame #4030. Examples of the *MO* segmentation results of every five frames, starting from frame #3885, are shown in first two rows of *Figure 4-8*. The displayed *MOs* were generated from *equation (4-10)* prior we apply the thinning process. Four results of every 2 frames are displayed in each image of the third row and overlaid on top of each other. We use darker color to represent the more recent *MO* result. The overlaid results are able to display the tempo of the action. The numbers at

the top correspond to the image frame numbers.

Figure 4-9: MO segmentation results of person squatting down.

The second example shows an attempt to break door. It starts from frame #3730 to frame #3860. Examples of the *MO* segmentation results of every five frames, starting from frame #3739, are shown in Figure 4-10. Four results with every frame are displayed in each image of the third row and overlaid on top of each other.

The third example illustrates a running action in the scene from the left to right and then back to left. It starts from frame #1560 to frame #1609. Examples of the *MO* segmentation results of every four frames, starting from frame #1564, are shown in Figure 4-11. Four results with every frame are displayed in each image of the third row and overlaid on top of each other.

Figure 4-10: MO segmentation results of person breaking door with occlusion.

Figure 4-11: MO segmentation results of person running.

The fourth example captures one person walking along the optical-axis. It starts

from frame #3970 to frame #4099. Examples of the *MO* segmentation results of every three frames, starting from frame #3980, are shown in *Figure 4-12*. Four results with every 2 frames are displayed in each image of the third row and overlaid on top of each other.

Figure 4-12: MO segmentation results of person walking.

The fifth example displays a scene of a test scenario where four persons are walking together along the optical-axis. There are frequent mergings and separations of human body regions in this sequence with a non-moving person sitting in the lower left corner. This sequence starts from frame #5068 to frame #5196. Examples of the *MO* segmentation results of every seven frames, starting from frame #5070, are shown in *Figure 4-13*. Four results with every 2 frames are displayed in each image of the third row and overlaid on top of each other.

The sixth example illustrates an object-throwing action. The lower part of the actor was motionless. It starts from frame #3505 to frame #3640. Examples of the *MO* segmentation results of every two frames, starting from frame #3599, are shown in *Figure 4-14*. Four results with every frame are displayed in each image of the third row and overlaid on top of each other.

Figure 4-13: MO segmentation results of four persons walking.

Figure 4-14: MO segmentation results of object-throwing.

Test set 2

One example from test set 2 is displayed in *Figure 4-15*. In this test set, the activities were captured in dim light environments. Due to lack of light, many body parts cannot be seen and hence only the noticeable torso can be detected. This is acceptable since even human observer cannot do much better. This sequence takes from frame #3011 to frame #3093. *MO* segmentation results of every two frames which start from frame #3027 are shown in *Figure 4-15*.

Figure 4-15: MO segmentation results of person walking in dim light environments.

Test set 3

One example from test set 3 are displayed in *Figure 4-16*. In this figure, it shows example results of segmentation in light changing environment where the lights were being switched off. It starts from frame #1001 to frame #1249 of a walking person sequence. The results are displayed in *Figure 4-16(a), (b), (d), (f), (g), (h), (i), (k), (m), (n)* and *(o)*. *Figure 4-16 (c), (e), (j) and (l)* are the original intensity image of *Figure 4-16 (b), (d), (i) and (k)* respectively. Some front lights from the ceiling were turned off at frame #1090, *Figure 4-16(e)*. Consequently, the front object at the left margin turned dark, and the segmentation cannot be used from *Figure 4-16 (f) to Figure 4-16 (g)*. The system started to perform correctly at *Figure 4-16 (h), (i) and (k)*. Then some back lights from the ceiling were turned off at *Figure 4-16 (l)*. The following segmentation result at *Figure 4-16 (n)* cannot be used. The system started to perform correctly at *Figure 4-16 (o)*. Since *Figures (k), (m), (n) and (o)* display 4 continuous MO segmentation results, it shows that the system needs only two frames to recover and perform correctly. Compared with other background subtraction systems, the proposed

approach needs much less time to recover. For background subtraction approach, the restoration of background model from global changes depends on learning rate. For example, the system proposed by Li and Leung [4] can take more than sixty frames to restore after lighting change. This illustrates one important advantage of the temporal difference.

Test set 4

Figure 4-16 shows example results from a few sequences of test set 3 captured inside an elevator where the lower body part usually remains motionless. The results are good in general.

Figure 4-17 shows example results from 3 indoor sequences of robbery scenes. The video quality is low since it was captured from a closed circuit camera with uncontrolled lighting condition. For example, sequences *R1* and *R3* were affected by lighting disturbances from windows facing the street. The disturbances create many noisy regions in the segmentation results. Despite these, the segmentation results are good enough for human observers to distinguish the disturbances and perceive the activities based on motion.

Figure 4-16: Samples of MO segmentation results from test set in light changing environments; (c), (e), (j), (l) original frames, (a),(b),(d),(f),(g),(h),(i),(k),(m),(n),(o) MO segmentation results.

Figure 4-17: Samples of MO segmentation results from activities inside an elevator.

Figure 4-18: Samples of segmentation results from robbery scenes.

The robbery scenes, as mentioned previously, have low video quality, frequent lighting changes from windows facing outside, and relatively less / uncertain movement from those who were caught in the incidents.

The results here illustrate that the proposed system can produce segmentation results good enough for perception when there is enough motion occurring without

much noise. The activities include walking, battering, kicking, running, etc. with one or multiple persons. The system stops functioning properly when there is no or little motion. This is acceptable since “motionless” can be interpreted as non-interesting and non-urgent activity. On the other hand, though the results can still be recognized to certain extent when there is substantial noise, human can do better by looking at the original intensity video. This suggests that *MO* cannot be used alone. *MO* can be seen as a focusing mechanism to assist higher level human body motion recognizer that consults the original intensity video.

4.7 Summary and Discussion

In this chapter, we present a novel, effective and efficient framework to segment moving object outline using the technique of temporal difference. The overall segmentation results are good and robust with respect to the ease in perceiving the motion type in the scene. The results demonstrate that temporal difference can work as good as the background subtraction without background maintenance. Hence it is suitable and convenient to be set up at non-predetermined scenes (mobile surveillance) with complex background. It is faster than optical flow method and can work in real time. In addition, it has good adaptation to lighting changes and dim environment.

Chapter 5 Moving Pattern Representation and Motion Vector Generation

In [1], Bobick and Davis had developed a view based method to recognize specified human movements. They exploited *Motion Energy Image (MEI)* to describe spatial patterns of movements, and *Motion History Image (MHI)* to represent motion evolution. With this approach, motion features can be extracted from moving object silhouettes. This scheme works well with simple or separate object. In the following, we will exploit and adapt the *MHI* for moving pattern representation.

The table of symbols and abbreviations (*Table 5-1*), which will be used in the following description, is displayed below.

Table of symbols and abbreviations	
<i>MHI</i>	<i>Motion History Images</i>
<i>fMHI</i>	<i>front Motion History Images</i>
<i>rMHI</i>	<i>rear Motion History Images</i>
<i>iMHI</i>	<i>inner Motion History Images</i>
<i>SMO</i>	<i>Sedimentary Moving object Outline</i>
<i>MEI</i>	<i>Motion Energy Images</i>
<i>NOP</i>	<i>Non-Overlapping Parts</i>

Table 5-1: Symbols and abbreviations.

In this chapter, the descriptions of the block "Motion Pattern Analysis" shown in the flowchart of *Figure 3-1* will be presented. The inputs are original image and *MO* segmentation result, in which the input example is shown in *Figures 3-3 (a) and (b)*. The corresponding outputs include *fMHI*, *rMHI*, *iMHI*, *SMO* and motion vectors, which are shown in *Figures 3-4 (b), (c), Figure 3-6 (c), Figure 3-5 (b) and Figures 3-4 (d)* separately.

5.1 Motion History Image (MHI)

The motion history image [1] is a scalar-valued image where intensity is a function of recency of motion. The *MHI* is generated from a sequence of *ROIs* that indicates regions of motion. The original work used background subtraction but we use temporal difference in this work. Each *MHI* region corresponds to a region where

the moving object has just swept out at the most recent time period. To represent how motion is evolving, the *MHI* is generated with a simple replacement and decay operator. In the scalar-valued *MHI* image, the brightness of a pixel is determined by how recently there has been motion detected at that location.

One example to illustrate the process of *MHI* generation is shown in *Figure 5-1*. Let's assume the *ROI* is one square box and it starts moving from bottom left and stops at top right as displayed in *Figure 5-1 (a)*. The *MHI* results at time t_0 to t_4 are shown in *Figure 5-1 (b)* to *(e)*. The relationship of *MHI* at different time is shown in *Figure 5-1* and the generation of *MHI* can be regarded as a recursive process. For example, the *MHI* at time t_i consists of two parts which are the fading *MHI* at time t_{i-1} and the current *ROI* at time t_i . In an *MHI*, pixel intensity is a function of time at that location, where darker values indicate that motion has occurred recently while brighter values correspond to motion occurred a long time ago.

Figure 5-1: Example to illustrate the process of MHI generation.

As shown in Figure 5-2 (c), pixels currently in motion appear white, while pixels that appear in various shades of gray represent locations that have detected movement in the recent past. The *MHI* is calculated by combining temporal differencing information over time through the use of a temporal decay as:

$$MHI(x, y, k) = \begin{cases} t & \text{if } ROI_{k-1}^k(x, y) = 1 \\ \max(0, MHI(x, y, k-1) - 1) & \text{otherwise} \end{cases} \quad (5-1)$$

Figure 5-2: *MHI* result of walking activity (from left to right); (a) original frame, (b) temporal difference, (c) *rMHI*.

where k refers to frame k , t represents the maximum length of time or the maximum number of different layers in *MHI*. One example of the *MHI* construction result of a walking person is shown in Figure 5-2 (c). From the *MHI*, the direction of motion can be inferred from the gradual gradient transition across the layers. Since the layers occur at the rear with respect to motion direction, this type of *MHI* can be classified as the rear *MHI* or *rMHI*.

5.2 Front Motion History Image (fMHI)

Similarly, one can generate the *fMHI* to shift the motion layers to the front with respect to the motion direction as:

$$fMHI(x, y, k) = \begin{cases} t & \text{if } ROI_{k-1}^k(x, y) - ROI_{k-2}^{k-1}(x, y) > 0 \\ \max(0, fMHI(x, y, k-1) - 1) & \text{otherwise} \end{cases} \quad (5-2)$$

fMHI and *rMHI* can complement each other as shown in Figure 5-3 where the *rMHI* is poor due to occlusion.

Figure 5-3: *rMHI* and *fMHI* results of another walking activity; (a) original frame, (b) *rMHI*, (c) *fMHI*.

As another example, a person is standing up in Figure 5-4. There might not be enough motion cues from *rMHI*, but *fMHI* can grab better motion cue from the rising front of the human body. A problem related to *MHI* representation is that parts of the motion may be lost due to self-occlusion or overlapping of motion on the image plane. However, the extracted *rMHI* and *fMHI* may be informative enough for recognizing most actions, and they can serve as hints for further analysis.

Figure 5-4: *rMHI* and *fMHI* results of standing up activity; (a) original frame, (b) *rMHI*, (c) *fMHI*.

5.3 Inner MHI (iMHI)

The *MHI* approach is limited since it cannot track movement of overlapping body parts, for example, a person waving his arms in front of his body that is in motion too. We propose to isolate the mixed moving parts from each other and then track the

movement of each isolated part. The *MHI* can then be applied to generate motion feature for each tracked moving part. It is assumed that a region with similar intensity denote a single moving part. To segment the moving parts, the procedure can be illustrated as following:

In [1], *MEI* (Motion Energy Image) was proposed by Bobick and Davis to represent the swept region of a moving object within a fixed time interval in *MHI*. In our work, we will further analyze the *MEI* which is computed as:

$$MEI(x, y, k) = \begin{cases} t & \text{if } M(x, y, k) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (5-3)$$

where M can be $fMHI$, or $rMHI$.

The regions along the *TDE* inside the *MEI* are then collected and three intensity histograms are generated to derive thresholds for object segmentation as shown in Figure 5-5.

Figure 5-5: Histogram segmentation.

Histogram (a) is generated in 3 steps:

- The regions which are the 5×5 blocks along each *TDE* pixel are defined as R_{edge} which captures the pixels on the left and right of *TDE*. Hence, it captures at least half of the pixels belonging to the moving objects when the moving object is not too thin.
- Compute $R = MEI \cap R_{edge}$. R is likely to exclude more background pixels since the additional restriction of *MEI* has been applied.

- Histogram (a) is then generated according to each pixel intensity in R . An example of one histogram (a) is shown in *Figure 5-6 (II)*.

Figure 5-6: Histogram of pixel intensity of moving objects: (I) original frame, (II) histogram (a), (III) histogram (b), (VI) histogram (c).

Afterward, histogram (b) is generated by temporal smoothing. We accumulate the last 10 histograms of type (a) to generate histogram (b). Let the y-axis-value of the i th bin in histogram (a) of frame k be marked as $a_k(x_i)$. The histogram (b) of frame k is generated as

$$b_k(x_i) = \sum_{j=\max(k-10,0)}^k a_j(x_i) \quad (5-4)$$

This operation should not affect the intensity statistics and peaks of the collected moving pixels but tend to lower the peaks of the collected background pixels since they tend to have different intensity values. An example of histogram (b) is shown in *Figure 5-6 (III)*.

Finally, histogram (c) is generated by spatial smoothing. The X-axis of histogram (b) is partitioned into 50 bins. The average value in each bin is then computed and used as the new value for histogram (c) which has exactly 50 bins. An example of histogram (c) is shown in *Figure 5-6 (VI)*

We will select several peaks from the final histogram (c) as thresholds for segmentation. The number of peaks is empirically set as six. Let the y-axis-value be marked as $c(x_i)$. x_i will be selected as a threshold candidate if $x_i \in \mathcal{X}$ with $\mathcal{X} = \{x_i \mid c(x_{i-1}) \leq c(x_i) > c(x_{i+1})\}$. For each candidate threshold, the regions with pixels within the range of “threshold ± 15 ” gray levels are selected for inner region segmentation. Let the collection of the six thresholds in frame k be $IT_k = \{it_k(j) \mid j = 1, 2, \dots, 6\}$ where each $it_k(j)$ is one threshold to segment inner regions ($MEI.it_k(j)$) from current MEI . One $iMHI$ is obtained when at least three MEI s with the same threshold can be found, i.e. one from frame k , one from $k-1$ and one from $k-2$. An example of these six peaks, which are highlighted using thick lines, is shown in *Figure 5-6 (VI)*. The relation between MEI and $iMHI$ is illustrated in *Figure 5-7*, with $it_k(j) \approx it_{k-1}(m) \approx it_{k-2}(p)$. The procedure become

Figure 5-7: The relation between MEI and iMHI.

- 1 First, we generate the six MEI according to IT_k .

- 2 If $|it_k(j) - it_{k-1}(m)| < 10$, $MEI.it_k(j)$ will be added to the old *MEI* sequence related to $it_{k-1}(m)$, otherwise, the previous *MEI* sequence will be dropped and the first *MEI* of a new *MEI* sequence will be initialized according to the current $it_k(j)$. One example of 6 consecutive segmented results on *iMHI* is shown in *Figure 5-8*, in which the original and corresponding *iMHI* results are displayed in pair.
- 3 An *iMHI* can be generated when three layers of *MEI* from frames k , $k-1$ and $k-2$ can be found.
- 4 Once a *iMHI* is found, we can generate the *fMHI* and *rMHI* from it.

From previous discussion, there should be six regions corresponding to the six thresholds. In reality, we might not have exactly 6 different regions. In our system we only focus on the most significant inner moving regions. Small regions are ignored. Some other example results of the most significant *iMHI* are shown in *Figure 5-9*. From *Figure 5-9 (b)*, no cues of movement layers of legs and human body from the whole image can be detected by *rMHI*. However, layers can be detected quite well in *Figure 5-9 (c)* from *iMHI*. This novel approach of segmenting the moving objects can provide more information in analyzing sub-region motion.

Figure 5-8: One example of 6 consecutive segmented results on iMHI.

Figure 5-9: Inner motion extraction, (a) original frame, (b) rMHI from the whole image, (c) rMHI from iMHI.

5.4 Fast Motion Measurement

For faster motion, it is not possible to obtain continuous transition from one layer to the other, as in *Figure 5-10*, where the fast motion of a stick is observed. This has increased the difficulty to check whether the vector traverse through three layers of different colors, when we calculate the motion vector that will be discussed later in *section 5.6*. In our work, we employ morphological dilation to tackle this problem to certain extent and this brings additional advantage to fill gaps and holes inside the *MHIs* (*Figure 5-10 (c)*). On the other hand, it can merge regions incorrectly. However, if we are looking for consistent and monotonous transition, the merging should not

create serious problems. We employ the morphological operation here. The dilation of an image F is computed as:

$$G = F \oplus H \quad (5-5)$$

where H is a structuring element. In our experiment, H is defined as a 5×5 diamond pattern as shown below:

$$H = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Figure 5-10: Dilation of a swinging rod, (a) original frame, (b) rMHI, (c) frame after dilation.

In extreme situation, when fast motion occurs, it is difficult to extract consecutive layers from MHI since there are less overlaps of object regions from consecutive frames. One example of such cases is shown in Figure 5-11 (b). It is desirable to detect such situation and then label the moving region with the tag of fast motion.

Figure 5-11: nop MHI result of another example of a swinging rod, (a) original frame, (b) $rMHI (rMHI(x, y, k))$, (c) $fMHI (fMHI(x, y, k - 1))$, (d) nopMHI.

The observations of $rMHI$ and $fMHI$ imply that overlapping of two continuous layers decreases when motion speed increases. One example is shown in Figure 5-12, in which one object is moving from bottom left to top right. Figure 5-12 (a) and (b) are $rMHI$ and $fMHI$ of a normal speed moving object while (d) and (e) are $rMHI$ and $fMHI$ of a fast moving object. Empirically, overlapping of two continuous layers for normal speed motion is usually larger than 80 percents of the moving object size, while overlapping of two continuous layers for fast speed motion is less than 10 percents of the moving object.

Figure 5-12: Non-Overlapping Parts (NOP): (a) $rMHI$ of normal motion, (b) $fMHI$ of normal motion, (c) $nopMHI$ of normal motion, (d) $rMHI$ of fast motion, (e) $fMHI$ of fast motion, (f) $nopMHI$ of fast motion.

Hence the ratio of overlapping (or non-overlapping) area to the total area of $rMHI$ can be used to measure the speed of motion. In our work, the Non-Overlapping Parts (NOP) is obtained according to $fMHI$ and $rMHI$ as:

$$NOP(x, y, k) = \begin{cases} t & \text{if } rMHI(x, y, k) = fMHI(x, y, k - 1) = t - 1 \\ \max(0, NOP_t(x, y, k - 1) - 1) & \text{otherwise} \end{cases} \quad (5-6)$$

A measure of rapidity can be defined as $q(k) = \frac{Re(NOP(k))}{Re(rMHI(k))}$, where

$Re(NOP(k))$ is the region area of $NOP(k)$ which is defined as the number of pixels with intensity value larger than 0 in $NOP(x, y, k)$. $Re(rMHI(k))$ is defined similarly.

The range of $q(k)$ is $[0, 1]$.

Examples of non-overlapping MHI ($nopMHI$) with normal and fast motion are

shown in *Figure 5-12* (c) and (f). When the object moves slowly, the value of $q(k)$ will be small (close to zero). When the object has fast motion, the value of $q(k)$ will be large (close to one). Another example is shown in *Figure 5-11* where the large area of $nopMHI$ clearly indicates the fast motion.

5.5 Sedimentary MO

To track MO that stops moving, one can create a record of sedimentary MO (SMO) by computing the gradient magnitude (M) and orientation (O) at each new MO (or previous SMO) pixel from previous frame. If M and O have not changed from previous frame to current frame and no motion vector is found near such pixel, a SMO pixel is declared. The definitions for SMO and the changes are

$$\Delta M(x, y, k) = \begin{cases} 1 & \text{if } |M(k) - M(k-1)| < \rho \cdot M(k-1) \\ 0 & \text{otherwise} \end{cases} \quad (5-7)$$

$$\Delta O(x, y, k) = \begin{cases} 1 & \text{if } |O(k) - O(k-1)| < \beta \\ 0 & \text{otherwise} \end{cases} \quad (5-8)$$

$$SMO(x, y, k) = \Delta M(x, y, k) \cap \Delta O(x, y, k) \quad (5-9)$$

ρ is empirically set as 0.1 (tougher) for initial membership and 0.2 (easier) for subsequent membership. β is empirically set as $\pi/6$. Two examples of the experimental results are shown in *Figure 5-13*. In this figure, the black color edge is the current MO and the grey color edge is SMO .

Figure 5-13: The computation of SMO. (a), (c), (e), (g) are four original images for the kicking sequence; (b), (d), (f), (h) are the corresponding four SMO images; (i), (k), (m), (o) are four original images for the depositing object sequence; (j), (l), (n), (p) are the four corresponding SMO images.

5.6 Motion Vector Generation

Since the speed and orientation of motion provide strong clues for detecting the class of motion, it is desirable to generate motion vector with orientation and magnitude for motion description. A novel motion vector extraction approach is proposed in this section.

5.6.1 Motion Vector Extraction

To obtain the motion vectors of *MHIs*, two consecutive layers of different colors in *MHI* are theoretically needed at least. One example of two-layer *MHI* is shown in

Figure 5-14 (b) where one can compute the movement from the gray region to the neighboring bright region or vice versa by matching the smaller region to the neighboring larger region.

Figure 5-14: Two-layer and three-layer MHIs.

Since the temporal difference method is used to extract motion regions, we get two (instead of one) regions of the rod in the most recent layer (filled by the white color) in Figure 5-14 (b). One region corresponds to the covered background after the rod has moved while the other region corresponds to the rod in current frame. Hence, two-layer *MHI* cannot give reliable information to track and match the rod. We use 3 layers *MHIs* (Figure 5-14 (c)) in order to obtain robust and experimentally acceptable motion vectors. In general, more reliable motion vectors can be obtained if more layers are used. The drawback is that less motion vectors can be detected due to the stricter condition that must be satisfied. Another example of a three-layer *fMHI* is shown in Figure 5-15(b). Observation of Figure 5-15 (b) suggests that the distance between the two boundaries located in-between the three layers can denote the motion of the object.

Figure 5-15: Motion vectors extraction for a running action; (a) original frame, (b) three-layer fMHI, (c) three-layer rMHI, (d) thinned fMHI regions of the two brightest colors, (e) motion vectors (with directions from black to white) overlaid on the original frame.

Equivalently, we can use the skeletons of the moving regions labeled as color 1 and color 2 in Figure 5-15 (b) and (c) to compute the motion. Based on this suggestion, the motion vectors are generated by a two-step process. The first step is a region thinning process, i.e. thin the regions labeled as colors 1 and 2 in Figure 5-15 (b) and (c). The second step is motion vector computation. We label the thinned regions as *TR1* (Thinned Region 1) and *TR2* as shown in Figure 5-15 (d) and Figure 5-16. Chamfer distance [169] is then employed to generate the motion vectors.

Figure 5-16: Chamfer distances from $TR2$.

The chamfer distance from $TR2$ is employed to compute the motion vectors. Here, each point on the Chamfer distance map needs to store two pieces of information: the nearest $TR2$ point and the distance from such point. Each point (e.g. $P1(x,y)$ in Figure 5-16) on $TR1$ checks its stored chamfer distance from the nearest $TR2$ point (e.g. $P2(x,y)$ in Figure 5-16). The motion vector will be the vector that connects these two points from $TR2$ to $TR1$. In addition, the vector must be able to traverse through three layers of different colors (Figure 5-15 (b)) without gaps in-between. The distance between the two corresponding points can be defined as the magnitude of motion vector which is affected by motion speed and the number of frames grabbed in one second. In our experiment, the image sequences were grabbed at 15 frames per second.

The number of layers in *MHI* can be further extended to four, five and so on. The multiple-layer approach (more than 3) is similar to the three-layer approach. The thinned regions can be labeled as $TR1$, $TR2$, $TR3$ and so on. The motion vector can be the vector that connects two nearest chamfer distance points from $TR3$ to $TR1$.

Furthermore, the vector must traverse through the regions with different layers of colors. In general more layers can produce more reliable motion vectors. One drawback of generating vector using multiple-layer (more than 3 layers) is that less motion vectors can be detected due to the stricter condition that must be satisfied. Another drawback is that if the object moves back and forth or change the direction abruptly, the multiple layers requirement becomes a burden to the system. Hence we use only 3 layers in current work. Examples of Motion vector extraction result are shown in *Figure 5-17*.

5.6.2 Motion Vector Application

After we have generated the motion vectors, motion regions can be labeled with motion of uncertain, normal and fast as shown in *Table 5-2*.

Let SI stands for the sampling index and F_k be frame k . The next frame for processing is F_{k+SI} with $1 \leq SI \leq 9$, in which 9 is experimentally set to represent the maximum number of frames to skip. When half of the $TR2$ is classified as uncertain, SI is increased by 2 in order to facilitate computing motion vectors from slow motion. In another word, if we can obtain normal motion after increasing SI , the uncertain motion can be verified to be slow motion. Otherwise, the scene is classified as motionless. On the other hand, we decrease SI by 2 when half of the regions are classified as fast.

Motion Region	Description
Uncertain	Large motion area with few motion vectors: a scenario will be labeled as uncertain when the number of motion vectors is less than $Num_{TR2} / 2$, in which Num_{TR2} is the number of pixels in $TR2$.
Normal	Have motion vector detected for most $TR2$: a scenario will be labeled as normal when the number of motion vectors is not less than $Num_{TR2} / 2$. (Precondition: it is not a fast motion region)
Fast	Have substantial non-overlapping motion regions: a scenario will be labeled with fast motion when $q(k)$, computed from <i>section 5.4</i> , is not less than 0.5.

Table 5-2: Motion region classification.

Figure 5-17: Examples of Motion vector extraction where the vectors are generated from both the *fMHI* and *rMHI*; (a) original frames, (b) three-layer *rMHI*, (c) motion vectors (with directions from black to white) overlaid on the original frame. The vectors are generated from both the *fMHI* and *rMHI*.

5.7 Summary and Discussion

Motion pattern representation is an important step to understanding image data, and requires an exact region description in a form suitable for a classifier. In this chapter, the MHI is extended to front Motion History Images (*fMHI*), rear Motion History

Images (*rMHI*), inner Motion History Images (*iMHI*) and Sedimentary Moving object Outline (*SMO*) which represent how motion is evolved. Such kinds of descriptions can characterize different properties of the region. Furthermore, the representation is application independent. The combination of these motion pattern representations becomes powerful hints which can be passed to an event recognition system to perform high level interpretation of human behaviors. For instance, in *section 5.4*, this representation is utilized to segment and measure the motion.

Chapter 6 Convex shape modeling

Computer vision plays an increasingly important role in applications like surveillance systems, anthropometry, human factors design, ergonomics, teleconferencing, virtual reality and performance measurement of both athletes and patients with psychomotor disabilities. All of these areas require identification of the parts of a human body and estimation of their shape and motion parameters. Efficient and effective solutions would bring great advancement in these applications. In this chapter firstly we propose an effective and robust convex shape modeling approach using dynamic programming mechanism. Then we obtain several convex shapes that are the candidatures for human head and torso modeling. The full functionality of human body modeling will be discussed in next chapter.

6.1 Assumptions for Convex Shape Modeling

Human body modeling aims to reconstruct and configure a model. Before modeling human body, we have the following assumptions.

- Head can usually be inferred as convex circular shape. Head can be modeled by circle and is always circular no matter it is being captured from frontal or profile view.
- Torso can also be modeled as a convex ellipse shape in both frontal and profile views.
- Limbs are longer convex ribbons connected together. Hence they can be modeled as rectangles.
- Head and limbs are connected to torso, i.e. head location implies torso location, and vice versa.

In our work, the *LEM* generated according to *MO*, which is discussed in *chapter 4*, will be employed to recover the human body. In addition, any short line which length is two will be eliminated since its orientation can be greatly affected by noise. The angular information of each line is important in the convex shape extraction process.

In this chapter, the realization of block "Convex Shape Modeling" shown in Figure 3-1 will be carried out. The input is *LEM* of *MO* segmentation result and corresponding outputs are convex shape models, which are shown in *Figure 3-3 (c)* and *Figures 3-7 (b) to (i)*.

6.2 Overview of Convex Shape Modeling

From the above assumptions, a human body can be set up with three sub-models as circle, ellipse and rectangles, in which circle and ellipse are convex circular shape. The diagram of how to model a convex circular shape is shown in *Figure 6-1* and the details are discussed in next paragraph. Compared with convex circular shape modeling, rectangle modeling is easier and is not discussed in this chapter.

In our system, firstly we generate *Line Edge Map* of *MO* for each frame according to the approach discussed in *chapter 4*. Then we use a dynamic programming based method to generate 2 line chains of convex shapes. To obtain the line chains, firstly we separate all the lines in *MO* into two sets according to a selected initial line. One line chain will be selected on one side and another line chain on the other side of the initial line. Each line chain should satisfy the condition in composing a convex shape.

Figure 6-1: The flowchart of convex shape modeling.

To model a convex shape, firstly, we need to define and lay down the necessary conditions in detecting a convex shape. This is discussed in *section 6.3*. Then we need to provide a mechanism to obtain the optimal convex shape from large numbers of candidatures. Since dynamic programming is a technique which is suitable for solving optimal control problems, we employ it to generate convex shape. Two merit formulations are used in the dynamic programming process to integrate a new line into an existing line chain. The details of how to generate convex shape is described in details in *section 6.3.3*. Finally we combine similar convex shapes and generate feasible detection results which are discussed in *section 6.4*.

6.3 Convex Shape Generation

In Euclidean space, an object R is convex if and only if for any two points $x_1, x_2 \in R$, every point on the straight line segment x_1x_2 is also within the object. For example, a solid cube is convex, while anything that is hollow or has a dent in it is not convex. On the other hand, if a polygon is not convex, it is defined as concave. One example of convex and concave polygons is shown in *Figure 6-2*.

Figure 6-2: Two examples of convex and concave shapes. (a) a convex pentagon, (b) a concave hexagon.

For each line segment from the generated *LEM*, we can divide the image space into 2 parts using the line as the partition line. We can compute 2 best convex shapes by combining the line with other line segments from these 2 different parts respectively. This will be discussed in details in *section 6.3.1*. In our system, we use the mechanism of dynamic programming to generate 2 best convex shapes for each line segment. Suppose the total number of the line segments is n , we can have at most $2n$ optimal convex shapes. Later, a method is proposed to select an optimal pair of convex shapes to represent the human head and torso assuming that there is only one human subject in the scene.

The general idea of generating optimal convex shape for each line is that firstly we choose one line as the first line, and then we need to find repeatedly the next line to add to it to form a line chain of one convex shape. On the other hand, in dynamic programming viewpoint, we need to select a suitable line chain to be combined with the next line.

6.3.1 Image Space Partition

After we have selected an initial reference line, L_r , all the lines in *LEM* of *MO* can be divided into two parts employing L_r as a partition line. Lines on each side of L_r can be grouped together. Prior grouping, we need to look at way to determine if a line is on the same side of L_r .

Let $A = (x_1, y_1)$ and $B = (x_2, y_2)$ be two end points of L_r . Then we have the following equation.

$$0 = y - y_I - \frac{(y_2 - y_I)}{(x_2 - x_I)}(x - x_I) \quad (6-1)$$

With any arbitrary x and y , *equation (6-1)* will be zero only when (x, y) is on the line. If (x, y) is not on the line, the result will not be zero. In fact, every point with coordinates (x, y) on one side of the line will give negative value according to the formula or positive value when (x, y) is on the other side.

Hence a point with coordinates (x, y) can be defined to be on the negative side of a line when $F(x, y)$ of *equation (6-2)* is negative. Similarly the point with coordinates (x, y) is defined to be on the positive side of the line when $F(x, y)$ is positive. When the point is on the line, $F(x, y)$ will be equal to zero.

$$F(x, y) = y - y_I - \frac{(y_2 - y_I)}{(x_2 - x_I)}(x - x_I) \quad (6-2)$$

To determine whether a line L_k is on the negative or positive side of L_r , one can check whether L_k satisfies the following equations which are the negative (6-3) and positive (6-4) side conditions respectively.

$$\left\{ \begin{array}{l} ((F(x_e^k, y_e^k) < -r) \text{ AND } (F(x_s^k, y_s^k) < r)) \text{ OR} \\ ((F(x_s^k, y_s^k) < -r) \text{ AND } (F(x_e^k, y_e^k) < r)) \end{array} \right. \quad (6-3)$$

$$\left\{ \begin{array}{l} ((F(x_e^k, y_e^k) > -r) \text{ AND } (F(x_s^k, y_s^k) > r)) \text{ OR} \\ ((F(x_s^k, y_s^k) > -r) \text{ AND } (F(x_e^k, y_e^k) > r)) \end{array} \right. \quad (6-4)$$

where (x_s^k, y_s^k) and (x_e^k, y_e^k) are the starting and ending points of L_k . The value “ r ” is used to set up a small buffer zone. This means that the line may cross the reference line a bit. r is experimental set as 3. *Figure 6-3 (a) and (b)* illustrate portions

of the negative and positive sides of L_r . The regions are covered by the vertical parallel lines.

Figure 6-3: Examples of negative and positive regions. (a) negative region, (b) positive region.

Equations (6-3) and (6-4) can be applied repeatedly to partition lines into 3 groups, i.e. on the negative or positive side of L_r or none of the above. In this study, we are interested in only the first 2 groups.

After grouping, the image space can be partitioned into 2 parts, i.e. the *Negative Part (NP)* and the *Positive Part (PP)*. *NP* and *PP* contain the lines on the negative or positive side respectively. In the process, all lines with length larger than 3 can be set as L_r . One example of the representation for *PP* is shown in Figure 6-4 where the thicker lines in Figure 6-4 (b) and (c) are the initial reference lines.

Figure 6-4: An example of a PP from a real image. (a) original image, (b) line edge map, (c) the PP based on the initial line shown as a thicker line.

6.3.2 Convex Shape Composition

In our system, each convex shape is composed by recursively inserting a new line to an existing convex line chain. A merit is computed for each insertion and the best fitted line chain will be kept. The process in maintaining convexity can be illustrated in Figure 6-5 where L_r partitions the space into PP and NP . Furthermore, NP has been partitioned into 6 regions, A_1 , A_2 , A_3 , A_1' , A_2' and A_3' . Suppose we have line chain $L_r, L_{r+1} \dots L_k$ of a convex shape. For a new line segment L_{k+1} , if line segments L_r, L_k and L_{k+1} satisfy the condition of convexity, L_{k+1} can be added to line chain $L_r, L_{r+1} \dots L_k$. From the definition of convex shape in section 6.3, line L_{k+1} has to be inside region A_3 and the orientation of L_{k+1} cannot give rise to concave shape. This can be achieved by enforcing the following two constraints.

Figure 6-5: Convex shape definition.

Regional Constraint

Let β_1 be the larger turning angle of middle point of line L_r to L_k and β_2 the smallest turning angle of middle point of line L_r to L_{k+1} as shown in Figure 6-6.

L_{k+1} will be constrained to be in regions A_3 or A_3' if $\beta_2 \geq \beta_1$.

Figure 6-6: Illustration of the regional constraint.

Partition Constraint

For lines L_r, L_k, L_{k+1} , any two should be on the same side of the other line if they form a convex shape.

The first constraint restricts L_{k+1} to be in the lower part of *Figure 6-5* while the second constraint is the necessary condition for L_r, L_k, L_{k+1} to form a convex shape. Since the second constraint is more expensive to compute, we always check the first constraint first. Based on the first constraint, we sort the line segments according to their β_2 angular relationships with respect to L_r . Initially, when $k+1$ equals 1, the check can be omitted.

6.3.3 Convex Shape Optimization using Dynamic Programming

From previous discussion, our convex shape optimization problem can be mathematically expressed as computing new convex shape by inserting a new line to existing line chain recursively. While any recursive mathematical formula could be directly translated to a recursive algorithm, the recursive computation can bring inefficient programming result. To avoid this recursive explosion and reduce computational complexity, we can use dynamic programming to formulate the recursive algorithm as a non-recursive one that systematically records answers of the sub-problems in a table. Firstly the technical overview of dynamic programming is briefly illustrated as following.

In our work, we use a line chain map, $A[n][n]$, as shown in *Figure 6-7* to compute optimal convex shape, in which the y-axis labels the sorted line segment (with respect to β_2 in *section 6.3.2*) from L_1 to L_{n-1} , while the x-axis represents the iteration number

of the dynamic programming process.

L_{n-1}	$A_{1(n-1)}$						$A_{(n-1)(n-1)}$
L_{n-2}	$A_{1(n-2)}$						
...	...	...		...			...
L_p	A_{1p}						
L_{p-1}	$A_{1(p-1)}$			$A_{p(p-1)}$			
...	...						...
L_1	A_{11}						
	1	...	$p-1$	p	...	$n-2$	$n-1$

Figure 6-7: Line Chain Map Array.

Each element, A_{mp} , is a tuple $\langle C_{mp}, M_{mp} \rangle$ where C_{mp} records the line chain of an optimal convex shape from L_r to L_p in iteration m , and M_{mp} is the corresponding merit. The pseudo-code to generate $A[n][n]$ is shown below, in which the first two lines describe the initialization of the first column of $A[n][n]$.

- 1) FOR $k=1$ TO $n-1$
- 2) $A[1][k] = \langle \langle L_r, L_k \rangle, 0 \rangle$
- 3) FOR $i=2$ TO $n-1$ iteration
- 4) FOR $a=2$ TO $n-1$ // a : index of current line

$$M_{ia} = M_{(i-1)a}$$

$$C_{ia} = C_{(i-1)a}$$
- 5) FOR $b=1$ TO less than a // b : index of previous line

6) IF L_r, L_b and L_a satisfies convex condition

7) IF $\text{Merit}(C_{(i-1)b}, L_a) > M_{(i-1)a}$

8) $M_{ia} = \text{Merit}(C_{(i-1)b}, L_a)$

$$C_{ia} = \langle C_{(i-1)b}, L_a \rangle$$

where $\text{Merit}()$ computes the merit of a fit according to two measures (6-5) and (6-6).

They are the measures of completeness and roundness. The first measure compares the length of the line chain to that of the fitted circle. The second measure prefers a round convex shape. These 2 measures have a range from zero to one. They complement each other to find a suitable convex shape. The first measure is defined as:

$$M_1 = \begin{cases} \frac{TL_{mk}}{2\pi R} & \text{if } \frac{TL_{mk}}{2\pi R} < 1 \\ 1 & \text{otherwise} \end{cases} \quad (6-5)$$

where TL_{mk} stands for the total length of the line chain C_{mk} , and R is the radius of the fitted circle. The computation of R will be described later.

The second measure is defined as

$$M_2 = \begin{cases} \sin \theta & \text{if } 0 < \theta < \frac{\pi}{2} \\ 1 & \text{if } \theta \geq \frac{\pi}{2} \end{cases} \quad (6-6)$$

where θ is the measure of radius variation defined as $\theta = \frac{R}{\sigma} \bullet C$ with σ being the standard deviation of R . C is a constant and is set empirically as 0.2 in our system. From equation (6-6), we separate the merit calculation into two parts according to whether θ is larger than $\frac{\pi}{2}$. For the part $\theta \geq \frac{\pi}{2}$, we get $\sigma \leq \frac{2}{\pi} \bullet R \bullet C < 0.1R$ when we substitute

θ by $\frac{R}{\sigma} \bullet C$. Hence *equation (6-6)* can be explained as: if standard deviation σ is no larger than $0.1R$, M_2 is set as the maximum value. On the other hand, if standard deviation is larger than $0.1R$, M_2 is calculated as $\sin \theta$. The reason is that a smaller σ will be associated with a more circular shape. R is introduced to guarantee the merit measure is scale independent. Finally, the highest M_{mk} in A will be chosen and the corresponding C_{mk} will be the answer.

6.3.4 Computation of Radius R and deviation σ

Assume we have two line segments (L_r and L_m), we can use these two line segments to model a circle, in which we can calculate the radius according to the two line segments. Let the distance between the middle point of L_r and the middle point of L_1 be l and the angle from line segment L_r to L_1 be α as shown in *Figure 6-8*.

Figure 6-8: One example of a circle model.

Furthermore, let θ and β be the angles $\angle OAB$ and $\angle AOB$, and R the radius of a fitted circle. Hence if $0 < \alpha < \frac{\pi}{2}$, we have $\theta = \frac{\pi}{2} - \alpha$ and $\cos \theta = \frac{l}{2R}$. R can be computed as

$$R = \frac{l}{2 \cos \theta} \quad (6-7)$$

When $\frac{\pi}{2} \leq \alpha < \pi$, we have $\theta = \alpha - \frac{\pi}{2}$.

Suppose we have line chain $L_r, L_1, L_2, \dots, L_m$, then we can generate R_1 according to line segments L_r and L_1 and R_2 according to line segments L_r and L_2 . Similarly, we can generate $R_3, \dots, R_m$. Then the estimated radius of circle (R) is equal to average of radii $R_1, R_2, \dots, R_m$ as $R = \sum_{i=1}^m \frac{R_i |L_i|}{TL}$ where $|L_i|$ represents the length of line L_i and TL stands for the total length of the line chain C_{km} of current k^{th} iteration.

The computed R_i is modeled as a random variable of the normal distribution and the standard deviation becomes $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (R_i - \mu)^2}$ where the mean, μ , is set as R . A smaller σ will be associated with a more circular shape. To be scale independent, the measure of $\frac{R}{\sigma}$ is used as a merit in section 6.3.3.

6.4 Convex Shape Combination

After generating all the line chain map arrays (NA_{L_1} to NA_{L_n} for the negative part and PA_{L_1} to PA_{L_n} for the positive part) using dynamic programming, we will have many convex shape models that can be sorted by the merit. However, there are many same or similar shapes with common line segments. For example, if the line chain computed from NA_{L_k} are equal to, or subset of another line chain from NA_{L_j} , we can assume that all the line segments in these two line chains represent the same or similar convex shape. Hence combination of similar shapes is reasonable and necessary for robust convex shape modeling. One example of the redundant representations for two similar convex shapes based on two different initial lines is shown in Figure 6-9 where the thicker lines

in Figure 6-9 (b) and (c) are the reference initial lines.

Figure 6-9: An example of two similar convex shapes; (a) original image, (b) one convex shape shown in dark color, (c) another convex shape which is similar to (b).

For two convex shapes, if the overlapping region is larger than 90 percents of region of the larger shape, the line segments in these two shapes will be combined together to generate a new shape. For each shape, we can generate a rectangle to cover it. Suppose for two shapes, the rectangles can be defined as (see Figure 6-10):

$$\begin{cases} \text{Re}_1(\text{row1}_{\min}, \text{row1}_{\max}, \text{col1}_{\min}, \text{col1}_{\max}) \\ \text{Re}_2(\text{row2}_{\min}, \text{row2}_{\max}, \text{col2}_{\min}, \text{col2}_{\max}) \end{cases} \quad (6-8)$$

We further define:

$$\begin{cases} \text{row}_{\min} = \max(\text{row1}_{\min}, \text{row2}_{\min}) \\ \text{row}_{\max} = \min(\text{row1}_{\max}, \text{row2}_{\max}) \\ \text{col}_{\min} = \max(\text{col1}_{\min}, \text{col2}_{\min}) \\ \text{col}_{\max} = \min(\text{col1}_{\max}, \text{col2}_{\max}) \end{cases} \quad (6-9)$$

The intersection of two regions becomes:

$$\text{Re}_u(\text{row}_{\min}, \text{row}_{\max}, \text{column}_{\min}, \text{column}_{\max}) \quad (6-10)$$

An example of the intersection of two regions is illustrated in Figure 6-10, in which the intersection region is filled with parallel lines.

Figure 6-10: An example of the intersection of two regions Re_1 and Re_2 .

Two convex shapes will be combined together when the following boolean condition is true.

$$\frac{AreaOf Re_u}{\max(AreaOf Re_1, AreaOf Re_2)} > i \quad (6-11)$$

where the parameter i is empirically set as 0.9. The merit of the combined region is replaced by the larger merit of their members. Finally, we will have a series of convex shape models that are sorted by the merit. In our experiment, the output with the maximum merit will be regarded as head, since the head is comparatively more similar to a circle than other body part models such as the torso and limbs.

6.5 Summary and Discussion

The technique of generating convex shape using dynamic programming is described in this chapter. As a result, we can obtain several convex shapes that are the candidatures for human head and torso modeling with arbitrary pose and arbitrary

viewpoint. Extensive experiments, which will be shown in next chapter, demonstrate that the system can robustly and effectively generate convex shapes that can be employed for human body modeling.

Chapter 7 Human Body Modeling

This chapter proposes a system for automatic convex shape combination that represents a likely human body pose, provided that the convex shapes have been generated from previous chapter. The body model consists of three components: the head, torso and its limbs. The system suggests a body part localization procedure through the analysis of a structural view of body poses. The experimental results show that our method can be efficient and effective for reconstructing 2D human body model.

In this chapter, the implementation of the block "Convex Shapes Combination" shown in *Figure 3-1* will be described. The inputs are convex shape models and the output is body modeling result, which are shown in *Figures 3-7 (b) to (i)* and *Figure 3-3 (d)*.

The flowchart in constructing the human body is illustrated in *Figure 7-1*. Firstly, we obtain head and torso pairing by computing three compatible measures, which are convexity, link distance and size ratio, to choose one optimal head and torso combination. This is discussed in *section 7.1*. After having the optimal combination of head and torso, the limbs can be isolated by extracting the line segments that are not inside the head and torso regions. The details of limbs extraction is described in *section 7.2*. Then we refine the posture by combining head, torso and limbs in *section 7.3*. Finally experimental results for human body posture are illustrated in *section 7.4*.

Figure 7-1: The flowchart of human body modeling.

7.1 Head and Torso Combination

The head and torso can be modeled as convex shapes. Their combination process is treated here. It is assumed that head location implies torso location since they must be connected. A feasible solution can then start from this assumption. The details of the mechanism are described below.

7.1.1 Compatible Measures

In this work, we employ three measures, shape, size and link distance, to pair two compatible regions as head and torso. The rationales for choosing these measures are given below:

- Convexity: since we model head and torso as convex ellipse shapes in both frontal and profile views, we prefer shapes that give high measures of convexity.
- Link distance: it is desirable to have the detected head and torso near each other and they have no or less overlapping.
- Size ratio: we need to check whether the paired head and torso have sizes proportional to the ratio of an ideal head and torso.

These three measures can complement each other to compute pairs of compatible head-torso model. The details in computing each measure are described below.

7.1.1.1 Shape Measure Calculation

Firstly, the shape estimation can be calculated with the merit M which is described in previous chapter. From previous chapter, a larger merit M can represent a more convex shape. Hence, suppose we have merits M_1^k and M_2^k of two convex shapes

in the k^{th} pair, the convexity merit is then computed as the sum of them as $Convex_k$:

$$Convex_k = M_1^k + M_2^k \quad (7-1)$$

7.1.1.2 Link Distance Calculation

Secondly, the link distance is checked to determine the appropriateness in pairing two convex shapes. To measure the distance between two convex shapes, the following model pattern is proposed which is shown in *Figure 7-2* where we model two convex shapes with 2 circles and the central points O_1 and O_2 .

Figure 7-2: The link distance model.

The radii of the two convex shapes are R_1 and R_2 with $R_1 \leq R_2$. The distance between O_1 and O_2 is defined as Dis . Dis represents how near two convex shapes are. Three scenarios of Dis , in the range of $(0, +\infty)$, are shown in *Figure 7-3*.

Figure 7-3: Three scenarios of link distances. Dis is $R_2 - R_1$ in part (c).

Let R_1 correspond to the shorter radius and R_2 correspond to the longer radius. From Figure 7-3 (a) and Figure 7-3 (c), we have two scenarios of undesirable head-torso combinations from normal viewing angle of most human poses. According to these two cases, we can limit the range of Dis as:

$$R_2 - R_1 \leq Dis \leq 3R_1 + R_2, \quad \text{with } R_1 < R_2 \quad (7-2)$$

The upper limit of $3R_1 + R_2$ ensures that the head cannot be too far away from the torso while the lower limit of $R_2 - R_1$ indicates that the overlapping of head and torso has a limit. In normal viewing angle of most human poses, the setup of this range is reasonable, since even a human observer cannot accept two convex shapes as head and torso pair if Dis is out of the preset range. The ideal situation occurs when Dis is equal to $R_1 + R_2$ as shown in Figure 7-3 (b). Hence the proximity merit of two circles in the k^{th} pair can be defined as the following formulation.

$$Pro_k = \begin{cases} \sin\left(\frac{\pi}{2} * \frac{Dis^k}{R_1 + R_2}\right) & R_2 - R_1 < Dis^k \leq R_1 + R_2 \\ \cos\left(\frac{\pi}{2} * \frac{Dis^k - R_1 - R_2}{2R_1}\right) & R_1 + R_2 < Dis^k \leq 3R_1 + R_2 \end{cases} \quad (7-3)$$

The shape of *equation 7-3* is shown in *Figure 7-4*. From this figure, Pro_k will have the maximum value when Dis^k is equal to $R_1 + R_2$. Pro_k falls to zero when Dis^k is equal to $R_2 - R_1$ or $3R_1 + R_2$.

Figure 7-4: Shape of equation 7-3.

7.1.1.3 Size Ratio Calculation

We find that the area ratio of head and torso is approximately invariant. Hence it is reasonable to use the ratio of head and torso regions as a conditional parameter to generate head-torso model.

- ✧ Firstly, we need to find the invariant parameter, named reference ratio (RR) of size. We enclose the head and torso in two rectangles with area $AreaOfRec_h$ and $AreaOfRec_t$. RR is then defined as:

$$RR = AreaOfRec_t : AreaOfRec_h. \quad (7-4)$$

One example of such modeling with two rectangles is shown in *Figure 7-5*. Since RR will be different for frontal and lateral views, we generate RR by calculating the average value of RR for 30 sequences, in which the proportion of frontal and lateral views is 1:2 and the individuals include males and females from Asian and European. Empirically, RR is set as 4.5:1.

Figure 7-5: Example of rectangles enclosing head and torso.

- ✧ Similarly, we define a size ratio $SR_{th}^k = AreaOfRec_t^k : AreaOfRec_h^k$ with Rec_t^k and Rec_h^k being the rectangles enclosing the convex shapes from the k^{th} pair. SR_{th}^k is further restricted by $RR - \Delta < SR_{th}^k < RR + \Delta$ where Δ is the maximum difference between RR and SR_{th}^k . In this work, Δ is set as 3.5 such that SR_{th}^k is at least 1. Within this range, we can define a size ratio merit of the k^{th} pair as:

$$SRM_k = \cos\left(\frac{\pi}{2} * \frac{RR^k - RR}{\Delta}\right) \quad (7-5)$$

SRM_k will be zero if SR_{th}^k is out of range. The curve of equation 7-5 is shown in Figure 7-6, in which x-axis represents SR_{th}^k . From the figure, SRM_k will become maximum as 1 when SR_{th}^k is equal to RR . SRM_k will be zero when SR_{th}^k is either $RR - \Delta$ or $RR + \Delta$.

Figure 7-6: Shape of equation 7-5.

7.1.2 Pairing Head and Torso

We will select the optimal pair ($Pair(Re_1^i, Re_2^i)$), from n pairs of candidates as head and torso. The selection criterion is:

$$Pair = \{Re_1^k, Re_2^k; (\arg_k \max(Convex_k * Pro_k * SRM_k)(1 \leq k \leq n))\} \quad (7-6)$$

where $Convex_k$ is the merit sum of two convex shapes of the k^{th} pair, Pro_k is the proximity merit of the two convex shapes and SRM_k is the ratio merit of size of head

and torso.

From *equation (7-6)*, we can generate a workable human body model. However, in many situations, we find that the region of human torso is large and cannot be modeled using only one convex shape. One example is shown in *Figure 7-7 (b)*. The detected torso model is just the top part of a torso. To tackle this problem, we can expand current torso to include a connected convex shape. The details are discussed below.

Figure 7-7: Example in expanding the torso. (a) original image, (b) head-torso model before expanding, (c) head-torso model after expanding.

We recalculate a new merit $Convex'_k * Pro'_k * SRM'_k$ after expanding the torso (of current optimal head-torso combination) with one more candidate convex shape in which these two shapes satisfy the link distance constraint *equation (7-2)*. A new torso is formed according to the method described in *chapter 6*. Afterward, we recomputed all the merits. Particularly, $Convex'_k$ is computed as:

$$M_1^k + \max(M_2^k, M_3^k) \quad (7-7)$$

where M_3^k is the merit of the new shape added in. Finally, if the new

$Convex'_k * Pro'_k * SRM'_k$ is larger than the old one, we accept the new combination.

One example result after the torso expanding process is shown in *Figure 7-7 (c)*.

7.2 Limb Modeling

After head-torso pairing, we can generate limb according to the remaining *MO*.

First we assume that the limbs are around the torso. We can employ a rectangle

Re_{bottom}^k to cover all the *MO* that are below the torso. Similarly, we can generate Re_{left}^k ,

Re_{right}^k and Re_{top}^k to cover *MO* that are on the left, right and top of torso. In addition,

the rectangles $Re_{bottom}^k, Re_{left}^k, Re_{right}^k$ and Re_{top}^k should remove the region of head if

it is inside any of the rectangles. Finally, we select the rectangle with maximum region

(area) from $Re_{bottom}^k, Re_{left}^k, Re_{right}^k$ and Re_{top}^k as the major limb region which is

usually the leg part.

7.3 Head-torso-limb Modeling Refinement

With the limb part, one can further refine the selection by taking into consideration

the size ratio of limb to head. Empirically, the ratio of head to torso to limb is found as

1:4.5:5 (see *Figure 7-8*). It is obtained similarly as in previous section.

Figure 7-8: Example of the three rectangles models.

Let define the size ratio of limb to head as SR_{th}^k . The ideal ratio is 5. A new measure to pick head, torso and limb becomes:

$$Pair = \{Re_h^k, Re_t^k, Re_l^k; (\arg_k \max(Convex_k' * Pro_k' * (SRM_k + SRM_k'))(1 \leq k \leq n))\}$$

where

$$SRM_k' = \cos\left(\frac{\pi}{2} * \frac{SR_{th}^k - 5}{\Delta'}\right) \quad (7-8)$$

where Δ' is the maximum difference between two ratios and is experimentally set as 4. The meaning of Δ' is similar to Δ which is discussed in *section 7.1.1.3*. In addition, we need to use both SRM_k and SRM_k' , since SRM_k focuses on the ratio of head and torso while SRM_k' focuses on the ratio of limb and torso. One example of using this method to improve human body model is shown in *Figure 7-9 (c)*.

Figure 7-9: Head-torso-limb models. (a) original image, (b) head-torso model according to method in section 7.1, (c) head-torso-limb model according to method in section 7.3.

7.4 Experimental Results

In this section, to demonstrate the proposed system's robustness to human posture computation, three groups of human body reconstruction and configuration results are displayed. The tests take the convex shape input generated from previous

process described in *chapter 6*. The reconstructed human head-torso-limb labels are displayed from *Figure 7-10* to *Figure 7-12*. All the convex shapes have been sorted according to the likeness (*equations (6-5) and (6-6)*) of convex shape.

In *Figure 7-10* and *Figure 7-11*, two groups of the human body posture outputs are displayed. The first example captures one people walking along the optical-axis and the second one records one people falling down. The human body parts are differentiated by the thickness of the dark lines, in which the thickest lines inside a smaller rectangle represents head, the thicker ones inside a larger rectangle represents torso and the normal dark lines represent limbs.

The results illustrate that the proposed system can label human body effectively. However, our current human body posture computation is only focused on single person. To further verify the potential ability to extend to handle multiple people scenario, the third human head-torso labeling result is displayed in *Figure 7-12*, in which it captures two persons walking together along the optical-axis. From this result, only head-torso combination from one people is extracted according the method discussed in *section 7.1.2*. The human body parts are differentiated by the thickness of the dark lines, in which the thickest inside a smaller rectangle lines represents head, the thicker ones inside a larger rectangle represents torso. Actually the head and torso of another people can also be extracted. The example is shown in *Figure 7-13*. The displayed sample images are laid out in two groups in which each group include two rows and the first column in the first row of each group is the input *MO* segmentation result and the others are the most likely convex shape which are represented in darker

color. All the convex shape candidatures are sorted by the merit which is defined in *section 7.1* to identify the likelihood of convex shape. Hence the first convex shape has the highest merit. From this figure, we can find out that the head and torso of both two people can be extracted successfully. Further investigation can be made to model two human bodies based on our current convex shape extraction results.

Figure 7-10: Human pose estimation results of people walking.

Figure 7-11: Human pose estimation results of people falling down.

Figure 7-12: Human pose estimation results of two persons walking.

Figure 7-13: Convex shapes extraction for two people walking.

It is observed that the proposed system can produce the head-torso labels quite well for one person in a scenario of multiple persons and it demonstrates that the system has great potential to analyze multiple people scenario in the future. The scene where the system fails will be discussed in details in *chapter 8*.

7.5 Summary and Discussion

In this chapter, we have proposed a methodology for combining convex shapes to model complex dynamic human body. The approach was motivated by the simple observation of a feature view of body poses. By making the feature representation explicit in our approach, we obtained a straightforward means of convex shape combination. Empirical results have demonstrated the robustness and efficient utility of the method for human body modeling.

Chapter 8 Experimental Results

In this chapter, system performance evaluation is carried out. The proposed system has been implemented using the C++ language running on a Pentium IV system with a Meteor digitizer. The system is able to run at about 7 frames/sec with a resolution of 192×144 pixels, while the image sequences were grabbed at 15 frames per second.

Three tests were performed to assess the effectiveness of the proposed method. These include segmentation and perception experiment, motion pattern analysis, and human posture estimation.

1. Segmentation and perception experiment: it is employed to measure the effectiveness and robustness of motion segmentation results, in which the effectiveness is demonstrated by comparing with other people work and robustness is proved by investigating and applying the proposed method on various test scenes such as a large laboratory, an elevator and some TV collections of robbery scenes captured by close-circuit cameras. Moreover, some of the activities happened in dim or light changing environment. This has greatly increased the difficulty and challenge of the motion segmentation process. Nevertheless, impressive motion segmentation results have been obtained to demonstrate the capability of the proposed motion segmentation process.
2. Motion pattern analysis: in order to understand motion on issues, such as fast or slow human movement, novel motion pattern analyses are performed to generate descriptions of moving regions in a form suitable for a classifier. Such kinds of descriptions can characterize different properties of the region. Later they are applied to classify motion vector as fast or slow according to the motion vectors generated from $rMHI$ and $fMHI$. Hence by using the global speed, we can make a conclusion with the human activities as normal (*i.e.* walk) or fast (*i.e.* run). In addition, body part motion (represented by the proposed

iMHI) can be correctly detected in occlusion scenario, in which this motion information cannot be detected by *rMHI* or *fMHI*. Finally, one common problem of the temporal approach is that it cannot detect effective moving region if the object is motionless. The proposed *SMO* can potentially help us to record and remember the object when it stops moving. The experiment here is to measure the effectiveness of the proposed measures.

3. Human pose estimation: to estimate body posture, firstly we need to label body parts. In our work, we propose convex shape to represent human head and torso in which a novel technique of generating convex shape using dynamic programming is derived. The obtained convex shapes are then combined to suggest the likely posture. The experiment here measures the effectiveness of the posture estimation process.

These tests verify different features of the proposed system. Impressive results have been obtained.

8.1 Input Data

Four sets of test images were employed to assess the effectiveness of the proposed method. The first Test Set (45 sequences) was used in the work of [4]. It is a scene of an open area in a large laboratory with normal and fast activities like walking, moving chairs, fighting, crawling, etc. The second Test Set (10 sequences) captured activities in a large laboratory in dim light environment and the third test Set (10 sequences) captured activities in a large laboratory in light changing environment. The last Test Set is a collection of low quality video from closed-circuit cameras. It captured

activities in two kinds of environment. One is inside an elevator (11 sequences) and the other is TV collection of robbery scenes (5 sequences). Some examples of the images are displayed from *Figure 8-1* to *Figure 8-4*. The detail description and specification of each sequence can be found in *Table 8-1*. Examples of the input sequences can be found in *Appendix A*.

Set 1
	Two persons walking separately (#4791 and #4886)		One person running(#1747 and #1773)

	Falling down (#2712 and #2723)		Throwing box(#3555 and #3565)

	Two persons wrestling (#5282 and #5296)		Three persons walking together (#6782 and #6893)

	Two persons fighting (#6430 and #6460)		Four persons walking together (#5081 and #5189)	

Figure 8-1: Examples of test images; Set 1.

Figure 8-2: Examples of test images; Set 2.

Figure 8-3: Examples of test images; Set 3.

Set 4
	<i>Battering (#213 and #220)</i>		<i>Hugging (#375 and #380)</i>

	<i>Attacking (#500 and #525)</i>		<i>Three persons standing (#510 and #530)</i>

	<i>Two persons fighting (#720 and #735)</i>		<i>Break glass and rob (#7210 and #7235)</i>

	<i>Attacking (#7382 and #7402)</i>		<i>Threaten and run (#3942 and #3958)</i>

	<i>Rushing out (#2040 and #2048)</i>		<i>Dash a door to escape (#8352 and #8358)</i>	

Figure 8-4: Examples of test images; Set 4.

Test Set	Action Types	Test Image Set Specification			Test Environment
		Sequence Number	Human Actions	Description/Category	
Set I	Abnormal Action	I01	Battering	Vandalization, Robbery	large laboratory
		I02-I04	Breaking objects	Vandalization, Robbery	
		I05	Crawling	Sick, Escape	
		I06-I07	Eluding	Sick, Escape	
		I08-I09	Falling down	Beaten, Sick	
		I10	Deposit object		
		I11-I12	Fighting	Fight, Attack	
		I13-I14	Jumping	Escaping	
		I15-I16	Kicking objects	Vandalization	
		I17	Knocking down a person	Assault	
		I18-I19	Knocking door	Assault	
		I20-I23	Running	Escape, Assault	
		I24-I25	Throwing objects	Vandalization	
		I26	Tumbling	Beaten, Sick	
		I27-I28	Wrestling	Assault	
	Normal Action	I29-I30	Bowing		
		I31	Carrying a bag and walking away		
		I32	Making a phone call		
		I33-I34	Moving a chair and sitting down		
		I35	Pushing a trolley and walking around		
		I36-I37	Sitting down and standing up		
		I38	Squat		
		I39-I41	Walking separately	1, 2, 3 persons	
		I42-I44	Walking together	2, 3, 4 persons	
		I45	Working in front of a computer		
Set II	Abnormal Action	II01-II03	Breaking objects	Vandalization, Robbery	large laboratory dim light
		II04	Knocking door	Vandalization, Robbery	
		II05-II07	Running	Escape, Assault	
	Normal Action	II08-II09	Squatting down and standing up		
		II10	Walking	1 persons	
Set III	Abnormal Action	III01-III03	Battering	Vandalization, Robbery	large laboratory light changing
		III04-III05	Knocking door	Vandalization, Robbery	
	Normal Action	III06	Walking	1 persons	
		III07	Walking separately	2 persons	
		III08-III10	Waving arms		
Set IV	Abnormal Action	IV01	Attack	Fighting, attacking	TV collection robbery scenes dim light
		IV02	Break glass and rob	Fighting, attacking	
		IV03	Dash a door to escape	Dash, rush	
		IV04	Rush out	Running, fast moving	
		IV05	Threaten and run	Running, fast moving	
		IV06	Battering	Vandalization, Robbery	
		IV07-IV08	Fighting	Between two persons	
		IV09-IV10	Knocking down a person	Assault	elevator dim light
	Normal Action	IV11	Hugging	2 persons	
		IV12	Reading	3 persons	
		IV13-IV16	Standing	1, 2, 3, 4 persons	

Table 8-1: Description of Input Data.

8.2 Experimental Results

8.2.1 Segmentation and Perception Experiment

Motivated by the classic study on human body motion perception by Johansson using moving light displays [8;9], the segmentation results should allow easy recognition of biological motion patterns when the segmented frames are displayed consecutively to human observers. We generated 81 video scripts from the segmentation results and showed the video to 30 people. Each person was asked to identify the *number of people (NP)* and *action type (AT)* from each video script. We also recorded down the *number of viewings (NV)* for each person to confirm his / her finding, in which people gave up after at most two viewings. In addition, *number of frame (NF)* recorded the total number of frames in that sequence. It was found that 56 of the video scripts could be perceived perfectly with only one viewing. These scripts captured scenes with relatively much movement. The perception was good regardless of lighting change, dim light environment and being occluded in a crowd. The activities include walking, battering, kicking, running, etc. with one or multiple persons. The human observers were able to describe the activities as walking, fighting, throwing, running, etc.

In *Figure 8-5* and *Figure 8-6*, we have displayed two groups of 39 consecutive *MO* segmentation results of two scenarios, in which the first group starts from frame #6400 and the second one starts from #6080. In *Figure 8-5*, it is a scenario of two persons fighting, in which one original frame is shown as reference at the top left. Four results of every frame are displayed in *Figure 8-7* and overlaid on top of each other. We

use darker color to represent the more recent *MO* result. The overlaid results are able to display the tempo of the action. The numbers at the top correspond to the image frame numbers. *Figure 8-6* displays a scene of a test scenario where three persons were walking together along the optical-axis and later one person departed from the scene. Four results of every 2 frames are displayed in *Figure 8-8* and overlaid on top of each other. The darker color is used to represent the more recent *MO* result and the numbers at the top correspond to the image frame numbers.

From the previous two groups of *MO* segmentation results, we can find that the system stops functioning properly when there is no or little motion. For example, in *Figure 8-5*, sometimes the head and right leg of the right people cannot be segmented properly, when the corresponding parts of the actor are motionless. In addition, the attached shadow of the two persons is detected as foreground. Since our current system does not take shadow as the wanted result, this problem should be further investigated in future.

Example responses of the other less than perfect video scripts are tabulated in *Figure 8-9* which records the correctness of *AT* and *NP* in percentages. *AT* (*NP*) was computed by allocating one score to *AT* (*NP*) if the corresponding response from an observer on this video script was correct. Finally, *AT* and *NP* would be divided by 30 (people) to compute the average percentage of correctness. Since *AT* descriptions can be different from one person to the other, we accept the descriptions listed in *Table 8-2* as correct answers for each corresponding video script. We found that “*cannot determine*” was the most common errant response.

Figure 8-5: MO segmentation results of two persons fighting (I11).

Figure 8-6: MO segmentation results of three persons walking (I43).

Figure 8-7: Four MO segmentation results of two persons fighting and overlaid on top of each other (I11).

Figure 8-8: Four MO segmentation results of three persons walking and overlaid on top of each other (I43)

Scene number		Scene images	Segmentation result	Scene number		Scene images	Segmentation result
NF	NV			NF	NV		
AT	NP			AT	NP		
IV01				IV04
230	1.4			200	1.2		
90	90			93	90		
IV03				IV05
220	1.3			200	1.7		
90	93			90	87		
I05				I07
130	1.1			70	1.3		
97	100			90	93		
I27				IV14
70	1.2			170	1.2		
100	97			97	93		
IV15				IV10
150	1.4			40	1.1		
93	87			100	93		

Figure 8-9: Test scenes with less perfect perception rates. NF: number of frames in this sequence; AT(%): recognition rate of action type; NV: average number of viewings; NP(%): correct recognition rate of number of people.

<i>Sequence name</i>	<i>Major actions</i>	<i>Acceptable descriptions</i>
<i>Robbery 1</i>	<i>Break glass and rob</i>	<i>Fighting, attacking;</i>
<i>Robbery 2</i>	<i>Rush out</i>	<i>Running, fast moving</i>
<i>Robbery 3</i>	<i>Dash a door to escape</i>	<i>Dash, rush</i>
<i>Robbery 4</i>	<i>Threaten and run</i>	<i>Running, fast moving</i>
<i>Crawl 1</i>	<i>Crawl</i>	<i>Eluding, crawling, creeping</i>
<i>Crawl 2</i>	<i>Walk then crawl</i>	<i>Eluding, crawling, creeping</i>
<i>Assault</i>	<i>Attack</i>	<i>Fighting, attacking</i>
<i>Elevator1</i>	<i>Stand / motionless</i>	<i>No action</i>
<i>Elevator2</i>	<i>Stand / motionless</i>	<i>No action</i>
<i>Elevator3</i>	<i>Fight</i>	<i>Fighting, striking,</i>

Table 8-2: List of actions and acceptable descriptions for scenes difficult to interpret.

In addition, responses of all the Test Sets (from *Test Set 1* to *Test Set 4*) with respect to *NF*, *NV*, *AT* and *NP* are shown from Table 8-3 to Table 8-6.

Test Set I	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
NF	200	300	300	110	130	50	70	75	75	110	200	200	90	60	60
NV	1	1	1	1	1.1	1	1.3	1	1	1	1	1	1	1	1
AT(%)	100	100	100	100	97	100	90	100	100	100	100	100	100	100	100
NP(%)	100	100	100	100	100	100	93	100	100	100	100	100	100	100	100
Test Set I	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
NF	230	50	110	250	20	50	65	15	50	145	30	70	70	90	90
NV	1	1	1	1	1	1	1	1	1	1	1	1.2	1.1	1	1
AT(%)	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
NP(%)	100	100	100	100	100	100	100	100	100	100	100	97	97	100	100
Test Set I	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
NF	100	220	230	230	110	130	120	50	100	430	120	125	170	130	125
NV	1	1	1	1	1	1	1	1	1	1.1	1	1	1.2	1	1
AT(%)	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
NP(%)	100	100	100	100	100	100	100	100	100	97	93	97	100	97	100

Table 8-3: Responses of Test Set 1.

Test Set II	01	02	03	04	05	06	07	08	09	10
NF	100	120	260	115	75	30	115	50	40	260
NV	1.1	1	1.2	1	1	1	1	1	1	1
AT(%)	97	100	97	100	100	100	100	100	100	100
NP(%)	100	97	97	100	100	100	100	100	100	100

Table 8-4: Responses of Test Set 2.

Test Set III	01	02	03	04	05	06	07	08	09	10
NF	185	250	250	100	150	50	200	180	100	100
NV	1.2	1	1.1	1	1	1	1	1	1.2	1.1
AT(%)	97	100	100	100	100	100	100	100	97	93
NP(%)	100	100	97	100	100	100	100	100	97	100

Table 8-5: Responses of Test Set 3.

Test Set IV	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
NF	230	100	220	200	200	90	120	100	50	40	60	75	60	170	150	110
NV	1.4	1.5	1.3	1.2	1.7	1	1.2	1.1	1	1.1	1.1	1.2	1	1.2	1.4	1.3
AT(%)	90	93	90	93	90	100	93	100	100	100	93	93	100	97	93	90
NP(%)	90	90	93	90	87	100	93	93	97	93	87	90	100	93	87	87

Table 8-6: Responses of Test Set 4.

The robbery scenes, as mentioned previously, have low video quality, frequent lighting changes from windows facing outside, and relatively less/uncertain movement from those who were caught in the incidents. Hence, the perceptions were less than perfect and higher *NV* was needed. The crawling action, which is not common, was found a bit difficult for human observers to perceive. This was complicated further by the attached *MO* from the shadow. The particular assault action on the other hand had substantial and frequent occlusion between the 2 actors. Hence, it was not easy for human to count the number of people in the scene. For the elevator video scripts with multiple people, the lower accuracy of *NP* is due to motionless subjects which *MO*

cannot be detected using the temporal difference approach.

The results here illustrate that the proposed system can produce segmentation results good enough for perception when there is enough motion occurring without much noise. The system stops functioning properly when there is no or little motion. This is acceptable since motionless can be interpreted as non-interesting and non-urgent activity. On the other hand, though the results can still be recognized to certain extent when there is substantial noise, human can do better by looking at the original intensity video. This suggests that *MO* cannot be used alone. *MO* can be seen as focusing mechanism to assist higher level human body motion recognizer that consults the original intensity video.

Performance on Light Changing Environment

In *Figure 8-10*, it shows example results of segmentation in light changing environment where the lights were being switched off. The recorded sequence starts from frame #1001 to frame #1250 of a battering sequence. The results are displayed in *Figure 8-10 (a), (b), (d), (f), (g), (h), (j), (l), (m), (n)* and *(o)*. *Figure 8-10 (c), (e), (i)* and *(k)* are the original intensity image of *Figure 8-10 (b), (d), (j)* and *(l)* respectively. Some front lights from the ceiling were turned off at frame #1134, *Figure 8-10 (e)*. Consequently, some front objects at the left margin turned darker, and the segmentation cannot be used from *Figure 8-10 (f)* to *Figure 8-10 (g)*. The system started to perform correctly at *Figure 8-10 (h)*. Then some back lights from the ceiling were turned off at *Figure 8-10 (k)*. The following segmentation result at *Figure 8-10 (m)* cannot be used. The system started to perform correctly at *Figure 8-10 (n)* and *(o)*. Since *Figures (j) (l)*,

(*m*), (*n*) and (*o*) display 5 continuous *MO* segmentation results, it shows that the system needs only two frames to recover and perform correctly. Compared with other background subtraction systems, the proposed approach needs much less time to recover. For background subtraction approach, the restoration of background model from global changes depends on learning rate. For example, the system proposed by Li and Leung [4] can take more than sixty frames to restore after lighting change. This illustrates one important advantage of the temporal difference.

Figure 8-10: Samples of MO segmentation results from battering sequence in light changing environments (III02).

Comparison

It is difficult to comprehensively compare the proposed system with other methods, since different approaches have different strengths and weaknesses, and hence each experimental result has been displayed to highlight capability of each particular system. However, to the best of our knowledge, we have not found any system that can extract motion segmentation as good as in our experiment while only using temporal difference approach.

Compared with temporal difference method, background difference need more processing, such as background estimation and maintenance. Generally speaking, reference background is estimated with the model parameters during a short period under the assumption that no foreground objects occurred during the training period. Moreover, the background scene and the camera are required to be stationary. In addition, background difference method needs more time to recover from changed environment. For temporal difference based method, it is suitable and convenient to be set up at non-predetermined scenes (mobile surveillance) with complex background. It is faster than optical flow method and can be applied in real time. For optical flow based method, according to the smoothness constraint, the movement of corresponding points in the two continuous frames should not be more than a few pixels. For some uncertain scenarios, this will limit the camera motion or background be slowly changed. Hence temporal difference is attractive to explore in our work.

There are very limited works using temporal difference based method to segment moving object in recent years. In one recent work, Huang [170] developed a system to

use temporal difference based approach to segment and trace moving object under outdoor night environment. Many systems can segment human motion indoor and in controlled outdoor environments. However, moving object detection at night is difficult when the moving objects are far away, small and their representations have low contrast against the background. Traditional methods based on the analysis of the difference between successive frames or a background frame will not work [170]. In this section, we evaluate the performance of several relevant motion segmentation methods by comparing moving object detection results under outdoor night environments. Since the proposed method is not specially designed to detect small moving object under dim light environments, the comparison will not be biased to our favour.

In *Figure 8-11*, our moving object detection method is compared with two background subtraction based approaches and one temporal difference based approach here. The test sequence of “two persons at night” used comes from Huang's work [170], in which it is captured in night scene by standard *CCD* cameras (Panasonic *WV-CW860A*) with a resolution of 320×240 pixels. The cameras are part of the *NLPR* (National Laboratory of Pattern Recognition) surveillance system.

One of the background subtraction based approaches is Stauffer's work [17], which has become the standard of background subtraction today [25]. They introduced an approach to model each background pixel by using the ratio of a mixture of adaptive Gaussians (*MOG*). Another one is Elgammal's nonparametric background subtraction (*NPB*) method [171]. The temporal difference based method is from Huang's work [170], which is designed for detecting moving objects for outdoor

night surveillance. All the motion segmentation results are shown in *Figure 8-11*. In this figure, the original image is shown in (a), the motion segmentation of *MOG*, *NPB* and Huang's approach are shown in (b) (c) (d) separately. *Figure 8-11* (e) shows the *MO* segmentation result using the proposed method and *Figure 8-11* (f) shows the *MO* segmentation result overlaid on the original frame.

Figure 8-11: Motion segmentation comparison with relevant work. (a) original frame (#110), (b) segmentation result of MOG, (c) segmentation result of NPB, (d) segmentation result of Huang's method. (e) MO segmentation result of proposed method, (f) MO segmentation result of proposed method overlaid on the original frame.

From the segmentation results shown above, we can find out that the input image is under dim light environment and there are some slight light changing around the windows, door and the tree. For the segmentation result from *MOG* method, the two objects are detected but quite some regions around the windows and door are falsely extracted as moving object though the parameters in the *MOG* algorithm are carefully

chosen [170]. Similarly the *NPB* results also falsely include quite a lot of regions around the windows and door as moving object. This algorithm is affected by high gray level gradients and it is the reason that there is some false detection near the windows. For Huang's method, it performs the best. For our method, its performance is comparable to the *MOG* and *NPB*. To further compare the motion segmentation results with Huang's work, some more segmentation results can be found in *Figure 8-12*.

The first column of *Figure 8-12* is the moving objects detection results from Huang's work, in which they are detected by changing contrast saliency between the successive frames 1, 20, 40, 60, 80, 100, 120 [170]. The test sequence is the same as previous one. The rectangles in red indicate the regions where there is a large change in contrast between the two images. They show the locations of objects which have moved. The second column is the *MO* segmentation results from our method by using the same inputs of Huang's work. The third column is the *MO* segmentation result overlaid on original image.

Figure 8-12: Comparisons of two object detection results.

From the *MO* segmentation, we have some false detection such as the region around the windows and door. However, from Huang's results, there is also some false detection from the windows, door and trees. In addition, *Figure B-8* and *Figure B-9* in *Appendix B* shows another group of *MO* segmentation results overlaying on the original images, which are generated from frame 1 to 200 with every 3 frames as input, i.e. input frames are 1, 4, 7, 10, ... 196, 199. By changing the time interval between input images, our proposed method shows good adaptability and robustness for motion segmentation. In *Figure B-8* and *B-9*, there are totally 65 results, in which we can detect moving human bodies exactly for all the *MO* segmentation results. The only problem is that there are some false detection such as the region around the windows and door in which the problem is also present in Huang's work. In addition, compared with Huang's work, we can accurately extract the outline of moving object, while Huang just extracted several rectangles where there is a large change.

Though our method is not specially designed for outdoor dim light environment, our proposed *MO* segmentation results are comparable with Huang's work. Huang *et al* [170] however employed manually selected thresholds in order to tune the system to work well in the specific environment. There are totally 4 manually set parameters, which are the size of rectangular, the threshold on contrast measure, the threshold on the distance measure between two rectangles and the threshold on the differences between contrast scores. Another limitation of their work is that they have the assumption that the objects of interest occupy only a small percentage of pixels in the whole image, which it can only be applicable to environment like outdoor surveillance

with a wide field of view and in places where there are fewer moving objects. Our system is found more suitable and versatile since it uses no manually selected threshold value. Our work can also make use of tracking process to improve the result and achieve similar result as in their work. Hence, we can conclude that our work is comparable if not better to theirs.

8.2.2 Motion Pattern Analysis

In this section, the experiments tested the proposed system's robustness to motion pattern representation. The motion pattern analysis operates on the input images and extracts useful motion pattern with vector that is needed by higher level processing modules. The experiments were carried out on different types of behavior sequences to analyze the motion pattern, in which the test cases are from *Test Sets I, II, III and IV*. As shown in *section 8.1*, the test scene is a view of an open area in a large laboratory in normal, light changing and dim light environments where there are many activities happening in real life.

As discussed in *chapter 5*, we mainly propose three groups of motion pattern description for image data understanding and classification. Firstly, we generate the front/rear motion history images (*rMHI* and *fMHI*) and motion vectors along the moving object boundary to measure motion. The motion vectors can be further processed for classification of normal or fast human behaviors. Secondly to track movement of overlapping body parts, inner *MHI* (*iMHI*) is proposed to isolate the mixed moving parts from each other and then track the movement of each isolated part. Finally, Sedimentary Moving object Outline (*SMO*) for representing motion history is

employed to record *MO* that stop moving. Hence, deposited objects can be discovered and tracked. The details are described below from *section 8.2.2.1* to *8.2.2.3*.

8.2.2.1 Motion Vector Detection

As discussed in *chapter 5*, after we have generated the motion vectors and non-overlapping *MHI*, motion regions can be labeled with speed of normal or fast. Based on the status of motion region, human activities can be further classified as normal or fast behavior.

In order to measure the correctness of magnitude and direction of the motion vectors, we simply track the center of two consecutive frames and treat center movement as reference motion vectors. Suppose the centre of human body is calculated as the gravity of the rectangle covering the moving human body, in which the rectangle is manually drawn. The center of gravity of a rectangle is computed as the point where two diagonals meet. Two examples are shown in *Figure 8-13 (a)* and *(b)*. In this figure, we can get the location of center point with coordinates $(47, 73)$ in frame k and $(59, 76)$ in frame $k+1$. The vector is generated by connecting the center points of two consecutive frames. The magnitude and direction of the motion vector are calculated as 12.4 (i.e. $\sqrt{(59-47)^2 + (76-73)^2}$) and 0.24 (i.e. $\arctan \frac{76-73}{59-47}$). The generated motion vector upon the original image is shown in *Figure 8-13 (c)*. On the other hand, the average magnitude and direction generated by the proposed method are 14.3 and 0.21 . In our work, the average vector is computed by adding all the vectors generated from *section 5.6*. The resulting motion vector is drawn in *Figure 8-13 (e)*. Compared

with the ground truth, the difference of direction and magnitude are 1% (i.e. $\frac{0.24-0.21}{\pi}$) and 15% (i.e. $\frac{14.3-12.4}{12.4}$). The magnitude is larger than the reference one since the movement of arms and legs are more significant than human torso while the reference motion vector is calculated mainly from human torso movement. The overall motion vector generation is reasonable and good.

Figure 8-13: Reference motion vector generation for human running sequence. (a) original frame k with the rectangle covering the moving human body, (b) frame $k+1$, (c) reference motion vector, (d) motion vectors generated from our method, (e) average of the motion vector from our method.

Due to scaling, the motion vector alone is not reliable for human movement status measure. The reason is that the moving object, which is closer to the camera, may have larger magnitude while the movement may not be significant. In our work, we propose the non-overlapping *MHI* combining with motion vectors to measure human movement type. The non-overlapping *MHI*, which is discussed in details in section 5.4, is mainly used to differentiate fast motion from normal one. A motion region is

classified as uncertain if we cannot generate “enough” reasonable motion vector, in which the “enough” is defined in *Table 5-2*. From this table, a scenario will be labeled with fast motion when $q(k)$, which is computed from *section 5.4*, is not less than 0.5. At the same time, the number of motion vectors is not less than $Num_{TR2} / 2$. One such example can be found in *Table 8-7*. A scenario will be labeled as normal when $q(k)$ is less than 0.5 and the number of motion vectors is not less than $Num_{TR2} / 2$. One example is displayed in *Table 8-9*. A scenario will be labeled as uncertain when the number of motion vectors is less than $Num_{TR2} / 2$.

In *Figure 8-14*, one experimental result with non-overlapping *MHI* of fast motion pattern is shown. *Figure 8-14 (1)*, (3) and (5) show the motion vectors overlaid on the original frame, (2), (4) and (6) display corresponding non-overlapping *MHI*.

From *Figure 8-15* to *Figure 8-19*, the displayed sample images are laid out in five columns which are the original frame, temporal difference frame, *rMHI*, *fMHI*, and motion vectors overlaid on the original frame. In the motion vectors frame, the motion direction is denoted from the dark to bright and the length of the dark-bright line denotes the magnitude of corresponding motion vector.

Figure 8-14: Fast motion pattern for the battering sequence (I03); (1), (3), (5) the motion vectors overlaid on the original frame, (2), (4), (6) corresponding non-overlapping MHI.

In Figure 8-15, it shows two example results of the detected motion patterns, in which the first four rows capture one people running while the last four rows capture one people dashing a door to escape from a robbery scene. Both activities are classified as fast behavior. The table of classification based on the measure to classify these two sequences is shown in Table 8-7. In this table, Num_{TR2} is the number of pixels in $TR2$ and $TR2$ is the thinned region discussed in section 5.6. The row with label “Number of motion vectors from $TR2$ ” records the percentage of Num_{TR2} in current image. The $q(k)$ row represents the $q(k)$ value and the row with label “Motion region classification” denotes the motion region type.

<i>Image</i>	<i>Row 1</i>	<i>Row 2</i>	<i>Row 3</i>	<i>Row 4</i>	<i>Row 5</i>	<i>Row 6</i>	<i>Row 7</i>	<i>Row 8</i>
<i>Number of motion vectors from TR2</i>	0.6	0.6	0.7	0.7	0.6	0.5	0.5	0.6
<i>q(k)</i>	0.7	0.7	0.8	0.8	0.6	0.7	0.7	0.8
<i>Motion region classification</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>

Table 8-7: Motion region classification for Figure 8-15.

In Figure 8-16, the first four rows capture one people threatening and running while the last four rows capture two people wrestling with each other in elevator. The first activity is classified as fast behavior while the second one is classified as normal behavior. The reason why the second is classified as normal is that the motion region of wrestling is not fast and hence is classified as normal, in which the classification table is shown in Table 8-8. In Figure 8-17, the first four rows capture one people waving his arm while another people just walk away and the last four rows capture one people rushing out from a robbery scene. In addition, the first activity is classified as normal behavior and the second one is classified as fast behavior. The classification table is shown in Table 8-9.

<i>Image</i>	<i>Row 1</i>	<i>Row 2</i>	<i>Row 3</i>	<i>Row 4</i>	<i>Row 5</i>	<i>Row 6</i>	<i>Row 7</i>	<i>Row 8</i>
<i>Number of motion vectors from TR2</i>	0.6	0.7	0.6	0.5	0.5	0.6	0.7	0.6
<i>q(k)</i>	0.7	0.7	0.6	0.6	0.2	0.3	0.4	0.3
<i>Motion region classification</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Normal</i>	<i>Normal</i>	<i>Normal</i>	<i>Normal</i>

Table 8-8: Motion region classification for Figure 8-16.

<i>Image</i>	<i>Row 1</i>	<i>Row 2</i>	<i>Row 3</i>	<i>Row 4</i>	<i>Row 5</i>	<i>Row 6</i>	<i>Row 7</i>	<i>Row 8</i>
<i>Number of motion vectors from TR2</i>	0.7	0.7	0.6	0.8	0.6	0.7	0.7	0.7
$q(k)$	0.3	0.4	0.4	0.4	0.5	0.6	0.7	0.7
<i>Motion region classification</i>	<i>Normal</i>	<i>Normal</i>	<i>Normal</i>	<i>Normal</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>

Table 8-9: Motion region classification for Figure 8-17.

In Figure 8-18, motion patterns for seven different sequences are shown; the 1st row shows walking (normal behavior), 2nd row shows knocking at the door (fast behavior), 3rd row shows eluding (fast behavior), 4th row shows crawling (fast behavior), 5th row shows stick swinging (fast behavior), 6th row shows throwing (fast behavior), 7th row shows attacking (fast behavior), 8th row shows walking under light changing (normal behavior). The classification table is shown in Table 8-10.

<i>Image</i>	<i>Row 1</i>	<i>Row 2</i>	<i>Row 3</i>	<i>Row 4</i>	<i>Row 5</i>	<i>Row 6</i>	<i>Row 7</i>	<i>Row 8</i>
<i>Number of motion vectors from TR2</i>	0.7	0.8	0.7	0.7	0.6	0.8	0.5	0.7
$q(k)$	0.2	0.6	0.6	0.6	0.6	0.8	0.6	0.4
<i>Motion region classification</i>	<i>Normal</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Normal</i>

Table 8-10: Motion region classification for Figure 8-18.

In Figure 8-19, more motion patterns for other seven different sequences are shown; the 1st row shows bowing (normal behavior), 2nd row shows squatting down (normal behavior), 3rd row shows waving arms (normal behavior), 4th row shows attacking in low light environment (fast behavior), 5th row shows breaking glass and robbing (fast behavior), 6th row shows kicking (fast behavior), 7th row shows another kicking (fast behavior), 8th row shows pushing a trolley (normal behavior). The

classification table is shown in *Table 8-11*.

<i>Image</i>	<i>Row 1</i>	<i>Row 2</i>	<i>Row 3</i>	<i>Row 4</i>	<i>Row 5</i>	<i>Row 6</i>	<i>Row 7</i>	<i>Row 8</i>
<i>Number of motion vectors from TR2</i>	<i>0.6</i>	<i>0.5</i>	<i>0.8</i>	<i>0.6</i>	<i>0.5</i>	<i>0.6</i>	<i>0.6</i>	<i>0.7</i>
<i>q(k)</i>	<i>0.4</i>	<i>0.2</i>	<i>0.3</i>	<i>0.5</i>	<i>0.5</i>	<i>0.6</i>	<i>0.7</i>	<i>0.2</i>
<i>Motion region classification</i>	<i>Normal</i>	<i>Normal</i>	<i>Normal</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Fast</i>	<i>Normal</i>

Table 8-11: Motion region classification for Figure 8-19.

The results of motion vector extraction are shown from *Table 8-12* to *Table 8-17*.

There are two kinds of errors:

- The false positive number: the number of frames that motion region is detected when there is none;
- The false negative number: the number of frames that motion region is not detected when there are movements.

The definitions of the abbreviation used in the following tables are:

- NF: *number of frames in this sequence*
- FP: *false positive*
- FN: *false negative*
- SR: *success rate*

$$\text{where } SR = \frac{NF - FP - FN}{NF}$$

Figure 8-15: Motion patterns with vector results of people running (I20: images #1558, #1561, #1564 and #1567) and dashing a door to escape from a robbery scene (IV03: images #8340, #8346, #8352 and #8358).

Figure 8-16: Motion patterns with vector results of people threatening and running (IV05: #3948, #3954, #3960 and # 3966) and striking in elevator (IV07: #507, #514, #521 and #528).

Figure 8-17: Motion patterns with vector results of people waving arm (III08: #5780, #5781, #5782 and #5783) and rushing out from a robbery scene (IV04: #2047, #2048, #2049 and #2050).

Figure 8-18: Motion patterns with vector results of eight different human activities: 1st row shows walking (I39), 2nd row shows knocking at the door (I18), 3rd row shows eluding (I06), 4th row shows crawling (I05), 5th row shows stick swinging (I02), 6th row shows throwing (I24), 7th row shows attacking (I01), 8th row shows walking under light changing (III04).

Figure 8-19: Motion patterns with vector results of eight different human activities: 1st row shows bowing (I29), 2nd row shows squatting down (II08), 3rd row shows waving arms (III09), 4th row shows attacking in low light environment (III02), 5th row shows breaking glass and robbing (IV02), 6th row shows kicking (I15), 7th row shows another kicking (I16), 8th row shows pushing a trolley (I35).

Table 8-12: Test result of motion pattern detection for Test Set I (sequences 01-15).

Table 8-13: Test result of motion pattern detection for Test Set I (sequences 16-30).

Table 8-14: Test result of motion pattern detection for Test Set I (sequences 31-45).

Table 8-15: Test result of motion pattern detection for test set II (sequences 01-10).

Table 8-16: Test result of motion pattern detection for Test Set III (sequences 01-10).

Table 8-17: Test result of motion pattern detection for Test Set IV.

To further illustrate the false positives and negatives, Figure 8-20 shows

examples for false positives and negatives.

Figure 8-20: Images for false positive and negative: 1st row shows an example of false positive (I09) (the shadow is detected as motion and actually there is no motion), 2nd row shows an example of false negative (I27) (motion of waving arm is not detected while there is movement happening. The reason is due to the waving arm is overlapped with human torso and hence it is difficult to detect).

The overall detection results are good and robust even in complex environment such as lighting change, dim light environment, being occluded in a crowd, low image quality and frequent lighting changes from windows facing outside.

Generally speaking, the results from *Test Set I* are the best compared with other three test sets, while results from *Test Set IV* are the worst. These observations are in line with our expectation, since the scenario in *Test Set I* is in normal lighting laboratory while the images of *Test Set IV* are from complex environment with frequent lighting changes from windows facing outside. Hence the success rate for sequence *IV1*, *IV2* and *IV5* in *Test Set IV* is only around 85%. However, the overall detection rate in *Test Set IV* is also acceptable.

For the low light environment and light changing sets *II9* and *III6*, the detection rate is also acceptable. For the light changing condition, two frames are needed to

recover from light changing environment and start to produce the correct detection. In Li's work [4], more than 10 frames are needed. For the result of light changing case, the false positive is much higher than false negative since more motion regions are extracted due to the light changing instead of actual movement. The high incorrect detection rates, such as sequence *I5*, *I6* and *I20* in *Test Set 1*, are due to the very fast motion and the embedded shadow of a crawling person respectively. One such kind of error detection results are shown in *Figure 8-21*, in which the shadow is detected as motion and corresponding motion vectors are extracted. Movement along the optical-axis in test set *I41* also causes poor recognition rate. The reason is that in this sequence, there are three people walking along the optical-axis and the movement along the optical-axis is not significant. Hence, even for a human observer, it is a little bit difficult to figure out the direction of the movement sometimes. In short, the result is encouraging with high detection rate.

Figure 8-21: Motion patterns with vector results of people running (I20: #1570, #1578, #1579 and #1580).

8.2.2.2 Inner MHI (*iMHI*)

To demonstrate the robust and efficient utility of the method for *iMHI* generation, several groups of experimental results are displayed. From these results, we can observe that human body part motion can be correctly detected. The *iMHI* segmentation results become powerful hints which can be employed to track occluded human body for multiple persons' activities.

From Figure 8-22 to Figure 8-24, the result images are laid out in six columns which are the original frame, *rMHI*, *fMHI*, motion vectors overlaid on the original frame, the most significant *iMHI* and motion vectors obtained from *iMHI*.

In Figure 8-22, motion patterns with *iMHI* for eight different sequences are

shown; 1st row shows attacking, 2nd row shows walking, 3rd row shows kicking, 4th row shows another walking, 5th row shows falling down, 6th row shows depositing bag, 7th row shows fighting, 8th row shows another fighting.

In *Figure 8-23*, motion patterns with *iMHI* for eight different sequences are shown; 1st row shows kicking, 2nd row shows attacking, 3rd row shows running, 4th row shows walking, 5th row shows throwing, 6th row shows wresting, 7th row shows bowing, 8th row shows two persons walking.

In *Figure 8-24*, motion patterns with *iMHI* for eight different sequences are shown; 1st row shows walking, 2nd row shows two persons walking, 3rd row shows three persons walking, 4th row shows four persons walking, 5th row shows another walking, 6th row shows striking inside an elevator, 7th row shows wresting inside an elevator, 8th row shows attacking inside an elevator.

Figure 8-22: Motion patterns with iMHI results of eight different human activities; 1st row shows attacking (I04), 2nd row shows walking (I36), 3rd row shows kicking (I28), 4th row shows another walking (I37), 5th row shows falling down (I08), 6th row shows depositing bag (I10), 7th row shows fighting (I11), 8th row shows another fighting (I11).

Figure 8-23: Motion patterns with iMHI results of eight different human activities; 1st row shows kicking (I16), 2nd row shows attacking (I01), 3rd row shows running (I16), 4th row shows walking (I33), 5th row shows throwing (I25), 6th row shows wrestling (I28), 7th row shows bowing (I30), 8th row shows two persons walking (I40).

Figure 8-24: Motion patterns with iMHI results of eight different human activities; 1st row shows walking (I34), 2nd row shows two persons walking (I42), 3rd row shows three persons walking (I43), 4th row shows four persons walking (I44), 5th row shows another walking (I32), 6th row shows striking inside an elevator (I08), 7th row shows wrestling inside an elevator (I06), 8th row shows attacking inside an elevator (I09).

8.2.2.3 Sedimentary Moving object Outline (SMO)

The example in *Figure 8-25* captures one person depositing an object. It starts from frame #4000 to frame #4110. In this figure, the top left image shows one example of the original scene while the other images are the segmentation results with *SMO* of every two frames, starting from frame #4016. Current *MO* is shown in black while *SMO* is drawn with grey color. Another example is shown in *Figure 8-26*, in which it captures one person knocking down another person under light changing environment. It starts from frame #3000 to frame #3250. The top left image shows one example of the original scene while the other images are the segmentation results with *SMO* of every two frames, starting from frame #3060. Current *MO* is shown in black while *SMO* is drawn with grey color.

These two sets of results here illustrate that the proposed system can produce sedimentary moving object segmentation results good enough for perception when some parts of human body stops moving or an object is deposited. However, the system stops functioning properly when lighting is changed. In this sequence when light changes, some background information is misclassified as *SMO*. One such kind of error detection result can be found in *Figure 8-26*. Since *SMO* is generated by computing the gradient magnitude and orientation at each new *MO* and previous *SMO* pixel, *SMO* may be faded when pixels in image are changed due to slight light changing.

Figure 8-25: The computation of SMO for the depositing object sequence.

Figure 8-26: The computation of SMO for the knocking down people sequence.

8.2.3 Human Pose Estimation

The third experiment tested the proposed system's capability to recover pose of a single human object. Two sub-processes were involved. The first process checked the effectiveness of convex shape computation. Convex shapes are needed in the second process, the body part labeling module.

Convex shape modeling

The convex shape modeling module operates on the *MO* segmentation results from *Test Sets* and then extracts convex shape models. In this section, *MO* segmentation results mainly from *Test Set 1* are employed to verify the effectiveness of convex shape extraction method. The *MO* results from *Test set 1* are similar with the ones in other three test sets. In addition, the activities in *Test set 1* are richer than the others. *Test set 1* captures different types of behavior with arbitrary poses and viewpoint in real life such as: walking, running, crawling, falling down, bowing, jumping etc. However, compared with *MO* results from *Test Set 1*, the other three test sets have different properties. For the *MO* results from *Test Set 2*, some parts of human body are not detected due to dim light. The *MO* results from *Test Set 3* include some background when light changes. For the *MO* results of *Test Set 4*, a lot of noise can be found from the robbery sequence.

Some example outputs of these detected convex shapes are shown from *Figure 8-27* to *Figure 8-31*. Each figure records four results. Each result is displayed across two rows, starting from the *MO* segmentation result and then the most likely convex shapes sorted in descending order and shown in darker color. The sorting is done according to the merit defined in *chapter 7* to identify the likelihood of convex shape. The computed merit is displayed at the lower right corner or top right corner. The first

convex shape has the highest merit.

Figure 8-27: Convex shapes extraction for people walking. Four images are #2560, #2564, #2568 and #2572.

Figure 8-28: Convex shapes extraction for people bowing. Four images are #4033, #4035, #4038 and #4040.

Figure 8-29: Convex shapes extraction for people falling down. Four images are #6446, #6452, #6458 and #6464.

Figure 8-30: Convex shapes extraction from the robbery sequence. Four images are #3940, #3942, #3958 and #3960.

Figure 8-31: Convex shapes extraction for two people walking. Four images are #5730, #5734, #5738 and #5742.

From these outputs, it is observed that, as a preliminary process of human pose estimation, useful convex shapes are successfully extracted, in which human head and torso are included. However even though the head should be the most optimal convex shape for a human body, actual results show that the most optimal convex shape may not always be head. It is possible since there may happen to be some other more optimal convex shape compared with head inside human body outlines. That's the reason that we cannot simply regard the most optimal one as head and we need to generate the head-torso combination to locate the head.

Body Part Labeling

The body part labeling process aims to check the effectiveness of human body reconstruction and configuration. All experiments in this section received inputs from the previous convex shapes extraction process. The output would be the estimated human head-torso-limb labels.

Though all the convex shapes have already been sorted according to the likeness of convex shape (in *chapter 7*), the first several convex shapes may not be human part candidates. However, we assume that human head and torso should be among the input convex shapes. Otherwise, the system will output the most likely head-torso model, even if it is not the true human body outlines. To overcome this limitation, more work (such as face detection) is needed to be done in future.

In *Figure 8-32* and *Figure 8-33*, they display examples of the human body posture outputs of every two frames after combining convex shapes. The human body parts are differentiated by the thickness of the dark lines, in which the thickest lines

inside a smaller rectangle represent head, the thicker ones inside a larger rectangle represent torso and the normal dark line represent limbs. *Table 8-18* and *Table 8-19* show the result of human pose estimation. To measure the effectiveness of convex shape combination, two kinds of errors are proposed:

- The false head or torso convex shape detection number: the number of frames at which the head or torso convex shape is not properly discovered.
- The false head-torso-limb combination detection number: the number of frames at which the head-torso-limb combination is not properly reconstructed.

The definitions of the abbreviation used in the following tables are:

- NF: *number of frames in this sequence*
- FV: *false head or torso convex shape detection*
- FC: *false head-torso-limb combination detection*
- SR: *success rate*

where $SR = \frac{NF - FC}{NF}$ and FV is used to identify the effectiveness of human head-torso label.

Figure 8-32: Human pose estimation results of people bowing and knocking at the door. (a) people bowing: starting from frame #4030, (b) knocking at the door: starting from frame #1557.

Figure 8-33: Human pose estimation results of people falling down and jumping. (a) people falling down: starting from frame #2505, (b) people jumping: starting from frame #4120.

Table 8-18: Human pose estimation Success Rate test results for Test Set I (Group 1).

Table 8-19: Human pose estimation Success Rate test results for Test Set I (Group 2).

It is observed that, as the final process, our human pose estimation method is

relatively effective at working with *MO* segmentation results with different types of human activities. Furthermore, from previous two groups of human pose estimation results and statistical data, we can find that the system stops functioning properly when the input human body segmentation has missed body part. That's the reason why crawling and squatting test in *Table 8-18* and *Table 8-19* have poor results.

Two failure examples are shown in *Figure 8-34* and *Figure 8-35*. The first failure example is the robbery sequence, in which there are one people running in front of other three persons. Examples of the pose estimation results of every two frames, starting from frame #3940, are shown in *Figure 8-34*. In this figure, the human pose estimation results are poor, since there are totally four persons activities and furthermore the *MO* segmentation inputs are occluded and partial human body is not detected. The failure case show how this human pose estimation system works when there are noisy and complex inputs without completely human body detection. However the proposed system can still estimate human pose, such as the second image in the first row (#3940), first image in the third row (#3958) and third image in the third row (#3962).

The second failure example is the sequence with one person walking and waving the arms, in which many movements occur in this sequence. Examples of the pose estimation results of every two frames, starting from frame #8025, are shown in *Figure 8-35*. In this figure, example of *FV* is shown in fourth image in the second row (#8039), in which the human torso cannot be extracted properly since the appearance of waved arm and human torso *MO* segmentation result is broken. One example of *FC* is shown in the fourth image in the fourth row (#8059). One false head-torso combination is

generated since the arm-torso has higher merit compared with the real head-torso combination. This kind of problem is difficult to overcome when false head-torso happens to have higher merit compared with the real head-torso. In order to solve this problem, further investigation employing history of body parts can be investigated in future.

Overall, the system can capture the head and torso combination successfully most of the time even human limbs cannot be found from the input. This highlights the great potential to perform multiple people and occluded-human body pose estimation using the proposed head-torso model. Examples of the detection results of multiple people scenario have been shown in *chapter 7*.

Figure 8-34: Human pose estimation results of robbery sequence.

Figure 8-35: Human pose estimation results of people waving arms.

8.3 Summary

A series of experiments have been conducted to test the system performance using three sets of test sequences. Encouraging experimental results have been obtained especially on *MO* segmentation results. At the same time, the findings provide valuable directions and hints for system development to overcome the current system's disadvantage and improve its performance in future work.

Chapter 9 Conclusion and Future Work

9.1 Work summary

The central theme of this thesis is to propose and build a human pose interpretation system, particularly in employing moving object outline extracted from temporal difference based segmentation approach. To demonstrate and investigate the functions and performance in different shapes and activities of a human pose recognition system, a serial of modules are implemented to achieve the target, which is described as below.

Chapter 4 proposes a novel, effective and efficient framework to segment moving object outline using the technique of temporal difference. It applies the temporal difference method with adaptive threshold to extract the moving object outline even in light changing and dim environment. The moving object outlines with *LEM* refined are later fed to the human posture estimation module of the system.

Chapter 5 addresses several motion pattern representations to understand image data, and provides motion description in a form suitable for a classifier. In this chapter the *MHI* is extended to front Motion History Images, rear Motion History Images, inner Motion History Images and Sedimentary Moving object Outline. The combination of all these descriptions can represent how motion is evolved. Such kinds of descriptions can characterize different properties of the region and the representation is application

independent. Due to shortage of time, we have not explored building higher level applications that can make use of these motion representations. The representations, however, form a strong foundation for future work.

In *chapter 6*, we propose a technique to generate convex shape using dynamic programming. We define two merits to generate the optimal convex shape and finally we can obtain several convex shapes that are the candidatures for human head and torso modeling with arbitrary pose and arbitrary viewpoint. Later, we further address a methodology for posture estimation from combining simple dynamical models with convex shape representations. The robustness and efficient utility of the method for human body labeling have been demonstrated.

9.2 Contributions

The main contribution of the work is its capability to extract moving object outlines in complex environment such as lighting changes and dim environment. Furthermore, several proposed novel motion pattern representations can provide powerful hints that can be passed to an event recognition system to perform high level interpretation of human behaviors. Finally, one robust and efficient utility of the method for human body labeling system is presented. The experimental results are very encouraging. The main contributions made in this thesis are listed as following:

➤ **Temporal difference based moving object outline segmentation:**

We present a novel, effective and efficient framework to segment moving object outline only using the technique of temporal difference. The overall segmentation results are good and robust with respect to the ease in perceiving

the motion type in the scene. For example, from analyzing moving object outline segmentation results, it is found that the motion activities can be perceived by human observers easily when the segmentation results are displayed consecutively. The contribution can be further described as following:

- An efficient threshold algorithm is applied to differentiate moving object from background. The process rate can reach up to 18 frames per second by employing this threshold algorithm to segment temporal difference result.
- We employ only temporal difference to detect and analyze moving objects. This indicates that a background maintenance free approach can be used to compute image differencing. The results demonstrate that temporal difference can work as good as the background subtraction without background maintenance. Compared to background subtraction, temporal difference can work without referring to the background image and hence it is suitable and convenient to be set up at non-predetermined scenes (mobile surveillance) with complex background.
- It is faster than optical flow method and can work in real time.
- This approach is particularly suitable to low light condition (when color cannot be detected) and it has very good adaptation to light changing situations.
- It has tolerance and ability to overcome very noisy conditions by

comparing with Stauffer's work which has already been discussed in *section 8.2.1*.

All the experimental results have shown robust and encouraging detection with great potential for future development to recognize human behaviors.

➤ **Motion pattern representation:**

It is an important step to understand image data, and obtain region description in a form suitable for a classifier. In this thesis, the *MHI* is extended to *fMHI*, *rMHI*, *iMHI* and *SMO* which represent how motion is evolved. Such kinds of descriptions can characterize different properties of the region which are discussed as following:

- rear Motion History Images: It can represent how motion is evolving and the direction of motion can be further inferred from the gradual gradient transition across the layers.
- front Motion History Images: It can grab better motion cue from the rising front of the human body.
- inner Motion History Images: From good performance results of *iMHI*, occluded human body part motion can be correctly separated under occlusion scenario, in which this motion information cannot be extracted by *rMHI* or *fMHI*.
- Sedimentary Moving object Outline: One common problem of the temporal approach is that it cannot detect effective moving region if the object is motionless. The *SMO* can potentially help us to record

and remember the object when it stops moving.

The most significant contribution for the motion pattern representation is that the overlapping movements are able to be detected. For example for the partial occlusion scenario, the proposed system has the capability to identify and recognize movement of overlapping body parts and generate motion feature for each tracked moving part.

Furthermore, the representation is application independent. The combination of these motion pattern representations becomes powerful hints which can be passed to an event recognition system to perform high level interpretation of human behaviors.

➤ **Motion Vector Generation:**

We have derived a novel frame work for motion extraction with stationary camera, which combines *MHIs* and chamfer distance together to generate motion vectors. The motion vector extraction result can be passed to an event recognition system to understand image data. In our current work, the speed (based on motion vector result) can be employed to classify as normal or fast behavior. Furthermore, human motion is more complicated than motion of other objects like cars or faces. For humans, large accelerations or sudden changes in motion are common and human articulation further aggravates the problem. In our system, it can extract the motion vector with the input image in light changing, dim light environments and low image quality and frequent lighting changes from windows facing outside. Therefore it can be extended to

apply to public traffic surveillance system to monitor the vehicle activities.

➤ **Convex shape extraction:**

A novel technique of generating convex shape using dynamic programming is described in this thesis in which dynamic programming based technique is employed to cut down *CPU* cost. As a result, we can obtain several convex shapes that are the candidatures for human head and torso modeling with arbitrary pose and arbitrary viewpoint. In addition, we can easily locate the human head by analyzing human head and torso modeling. Hence, the robust and effective human head extraction method can be applied to human tracking since several tracking approaches mainly rely on human head.

➤ **Human body estimation:**

In our work, we have proposed a methodology for combining simple dynamical models with convex shape representations as a way of modeling complex dynamical human body. The approach is motivated by the simple observation of a feature view of body poses. By making the feature representation explicit in our approach, we obtain a straightforward means of mediating between simpler models as a means of capturing more complex behavior. Afterward we make some improvements about the head-torso-limb modeling method. Empirical results demonstrate the robust and efficient utility of the method for human body modeling.

9.3 Future Work

Future research should be directed toward implementing a human behaviors

recognition system which will mainly focus on multiple moving objects in scene. The model with the motion language script like structures of intermediate patterns for an action should be derived and the multi-person analysis model can base on *Moving Light Displays (MLD)* concept, i.e. observation of pattern judged by human. Moreover, the multiple human body pose estimation can be further investigated, since we have already obtained some fair results in *chapter 8* which establish the head and torso model for one person even though with multiple people activities. Multiple human body pose estimation can be analyzed based on current outcome in the future.

9.4 Thesis conclusion

In this thesis a substantial contribution has been made in the design of a general human pose estimation system. Accordingly, a novel temporal difference based moving object outline extraction method and a novel convex shape generation mechanism are proposed. A thorough study of the system is put into practice and an in-depth demonstration has been carried into execution on the techniques. Suggestions for future research are presented based on the current work. The effect of the technique can be far-reaching with unforeseen potential.

References

- [1] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257-267, 2001.
- [2] I. Haritaoglu, D. Harwood, and Davis.L.S., "W: Real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809-830, 2000.
- [3] M. K. H. Leung and Y. H. Yang, "First Sight: A Human Body Outline Labeling System," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 359-377, 1995.
- [4] L. Li and M. K. H. Leung, "Suspicious Human Action Detection for Video Surveillance," *Advanced Concepts for Intelligent Vision Systems*, pp. 60-65, 2001.
- [5] S. J. McKenna, S. Jabri, and Z. Duric, "Tracking groups of people," *Computer Vision and Image Understanding*, vol. 80, no. 1, pp. 42-56, 2000.
- [6] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747-757, 2000.
- [7] C. Wren, A. Azarbaygaui, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, 1997.
- [8] G. Johansson, "Visual Motion Perception," *Scientific American*, pp. 76-88, 1975.
- [9] G. Johansson, "Visual Perception of Biological Motion and a Model for its Analysis," *Perception and Psychophysics*, vol. 14, no. 2, pp. 210-211, 1973.
- [10] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and*

- Machine Intelligence*, vol. 25, no. 10, pp. 1296-1311, 2003.
- [11] L. Li and M. K. H. Leung, "Integrating Intensity and Texture Differences for Robust Change Detection," *IEEE Transactions on Image Processing*, vol. 11, no. 2, pp. 105-112, 2002.
 - [12] T. Meier and K. N. Ngan, "Automatic segmentation of moving objects for video object plane generation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 525-538, 2003.
 - [13] R. Cutler and L. Davis, "Robust real-time periodic motion detection, analysis, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 781-796, 2000.
 - [14] P. Smith, T. Drummond, and R. Cipolla, "Layered motion segmentation and Depth Ordering by Tracking Edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 4, pp. 479-494, 2004.
 - [15] L. Wang, T. N. Tan, W. M. Hu, and H. Z. Ning, "Automatic gait recognition based on statistical shape analysis," *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1120-1131, 2003.
 - [16] C. Orrite, J. M. Rincon, J. E. Herrero-Jaraba, and G. Rogez, "2D silhouette and 3D skeletal models for human detection and tracking," *IEEE Proceedings of the 17th International Conference on Pattern Recognition*, vol. 4, pp. 244-247, 2004.
 - [17] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 246-252, 1999.
 - [18] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using adaptive tracking to classify and monitor activities in a site," *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, pp. 22-31, 1998.
 - [19] A. Lipton, H. Fujiyoshi, and R. Patil, "Moving target detection and tracking from real-time video," *IEEE Workshop on Application of Computer Vision*, pp. 8-14, 1998.
 - [20] V. Bruce, E. Hanna, and N. Dench, "The importance of mass in line drawings of faces," *Applied Cognitive Psychology*, vol. 6, pp. 619-628, 1992.
 - [21] J. Aggarwal and Q. Cai, "Human motion analysis: a review," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428-440, 1999.
 - [22] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585-601, 2003.

- [23] C. Cedras and M. Shah, "Motion-base recognition: a survey," *Image and Vision Computing*, vol. 13, no. 2, pp. 129-155, 1995.
- [24] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231-268, 2001.
- [25] T. B. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 90-126, 2006.
- [26] R. Jain and H. H. Nagel, "On the analysis of accumulative difference pictures from image sequences of real-world scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 206-214, 1979.
- [27] K. Irena and C. Sergio, "Temporal video segmentation: A survey," *Image Communication*, vol. 16, pp. 477-500, 2001.
- [28] B. Gunse, A. M. Tekalp, and P. J. L. Van Beek, "Content-based access to video objects: Temporal segmentation, feature extraction and visual summarization," *IEEE Transactions on Signal Processing*, vol. 66, no. 2, pp. 261-280, 1998.
- [29] R. Fablet and P. Bouthemy, "Spatio-temporal segmentation and general motion characterization for video indexing and retrieval," *10th DELOS Workshop on Audio-Visual Digital Libraries*, 1999.
- [30] D. Meyer, J. Denzler, and Niemann.H., "Model based extraction of articulated objects in image sequences for gait analysis," *Proceedings IEEE International Conference Image Processing*, pp. 78-81, 1998.
- [31] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal Computer Vision*, vol. 12, no. 1, pp. 42-77, 1994.
- [32] D. Gibson and M. Spann, "Robust optical flow estimation based on a sparse motion trajectory set," *IEEE Transactions on Image Processing*, vol. 12, no. 4, pp. 431-445, 2003.
- [33] S. D. Blostein, L. Zhao, and R. M. Chan, "Three-dimensional trajectory estimation from image position and velocity," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 4, pp. 1075-1089, 2000.
- [34] M. Memim and P. Perez, "Dense estimation and object-based segmentation of the optical flow with robust techniques," *IEEE Transactions on Image Processing*, vol. 7, no. 5, pp. 703-719, 1998.
- [35] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. Lyons, and A. K. Jain, "A

- background model initialization algorithm for video surveillance," *International Conference on Computer Vision*, pp. 733-740, 2001.
- [36] D. Farin, P. H. de With, and W. Effelsberg, "Robust background estimation for complex video sequences," *International Conference on Image Processing*, vol. 1, pp. 145-148, 2003.
- [37] S. Cohen, "Background estimation as a labeling problem," *IEEE International Conference on Computer Vision*, vol. 2, pp. 1034-1041, 2005.
- [38] D. Russell and S. Gong, "Minimum cuts of a time-varying background," *Proceeding British Machine Vision Conference*, vol. 2, pp. 809-818, 2006.
- [39] H. L. Eng, K. A. Toh, A. H. Kam, J. Wang, and W. Y. Yau, "An automatic drowning detection surveillance system for challenging outdoor pool environments," *International Conference on Computer Vision*, pp. 532-539, 2003.
- [40] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172-185, 2005.
- [41] M. Heikkila, M. Pietikainen, and J. Heikkila, "A texture-based method for detecting moving objects," *British Machine Vision Conference*, 2004.
- [42] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657-662, 2006.
- [43] D. S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 827-832, 2005.
- [44] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," *International Conference on Pattern Recognition*, vol. 2, pp. 28-31, 2004.
- [45] F. Kristensen, P. Nilsson, and V. Owall, "Background segmentation beyond RGB," *Asian Conference on Computer Vision*, pp. 13-16, 2006.
- [46] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337-1342, 2003.
- [47] S. J. McKenna, S. Jabri, Z. Duric, and H. Wechsler, "Tracking Interacting People," *International Conference on Automatic Face and Gesture Recognition*, pp. 348-353, 2000.

- [48] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara, "Detecting moving shadows: algorithms and evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 918-923, 2003.
- [49] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environments," *Computer Vision and Pattern Recognition*, vol. 2, pp. 406-413, 2004.
- [50] P. Fihl, R. Corlin, S. Park, T. B. Moeslund, and M. M. Trivedi, "Tracking of individuals in very long video sequences," *International Symposium on Visual Computing*, pp. 60-69, 2006.
- [51] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," *International Conference on Computer Vision*, vol. 2, pp. 1305-1312, 2003.
- [52] Y. Sheikh and M. Shah, "Bayesian modelling of dynamic scenes for object detection," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1778-1792, 2005.
- [53] J. Zhong and S. Sclaroff, "Segmentating foreground objects from a dynamic textured background via robust kalman filter," *International Conference on Computer Vision*, pp. 44-50, 2003.
- [54] S. Mahamud, "Comparing Belief Propagation and Graph Cuts for Novelty Detection," *Computer Vision and Pattern Recognition*, vol. 1, pp. 1154-1159, 2006.
- [55] R. Polana and R. Nelson, "Detection and recognition of periodic, non-rigid motion," *International Journal Computer Vision*, vol. 23, no. 3, pp. 261-282, 1997.
- [56] R. T. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, D. Tolliver, and N. E. O. Hasegawa, "A system for video surveillance and monitoring: Vsam final report," *Technical report CMU-RI-TR-00-12, Carnegie Mellon University*, 2000.
- [57] E. Stringa, "Morphological change detection algorithms for surveillance applications," *Proceedings British Machine Vision Conference*, pp. 402-412, 2000.
- [58] H. Wang and D. Suter, "A novel robust statistical method for background initialization and visual surveillance," *Asian Conference on Computer Vision*, pp. 13-16, 2006.
- [59] H. Wang and D. Suter, "Background initialization with a new robust statistical approach," *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 15-16, 2005.

- [60] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive background estimation and foreground detection using Kalman-filtering," *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, pp. 193-199, 1995.
- [61] H. Yalcin, R. Collins, and M. Hebert, "Background estimation under rapid gain change in thermal imagery," *Computer Vision and Image Understanding*, vol. 106, no. 2, pp. 148-161, 2007.
- [62] P. K. Shoo, S. Soltani, A. K. C. Wong, and Y. C. Chen, "Survey of thresholding techniques," *Computer Vision, Graphics and Image Processing*, vol. 41, no. 2, pp. 233-260, 1988.
- [63] A. Pikaz and A. Averbuch, "Digital image thresholding based on topological stable state," *Pattern Recognition*, vol. 29, pp. 829-843, 1996.
- [64] J. Shao, S. K. Zhou, and Q. Zheng, "Robust appearance-based tracking of moving object from moving platform," *Proceedings of the 17th International Conference on Pattern Recognition*, pp. 215-218, 2004.
- [65] P. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," *Computer Vision and Pattern Recognition*, vol. 1, pp. 511-518, 2001.
- [66] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153-161, 2005.
- [67] A. Verri and T. Poggio, "Motion field and optical flow: Qualitative properties," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 490-498, 1989.
- [68] H. Sidenbladh, "Detecting human motion with support vector machines," *International Conference on Pattern Recognition*, vol. 2, pp. 188-191, 2004.
- [69] J. J. Gonzalez, I. S. Lim, P. Fua, and D. Thalmann, "Robust tracking and segmentation of human motion in an image sequence," *International Conference on Acoustics, Speech, and Signal Processing*, pp. 29-32, 2003.
- [70] P. Sangi, J. Heikkila, and O. Silven, "Extracting motion components from image sequences using particle filters," *The 12th Scandinavian Conference on Image Analysis*, pp. 508-514, 2001.
- [71] S. X. Ju, M. J. Black, and Y. Yacoob, "Cardboard people: A parameterized model of articulated image motion," *International Conference on Automatic Face and Gesture Recognition*, pp. 38-44, 1996.
- [72] C. Bregler and J. Malik, "Tracking people with twists and exponential maps,"

- IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8-15, 1998.
- [73] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-575, 2003.
 - [74] L. Q. Xu and P. Puig, "A hybrid blob- and appearance-based framework for multi-object tracking through complex," *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 73-80, 2005.
 - [75] E. Borenstein and J. Malik, "Shape Guided Object Segmentation," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 969-976, 2006.
 - [76] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors," *European Conference on Computer Vision*, vol. 1, pp. 69-82, 2004.
 - [77] A. Kapoor and J. Winn, "Located Hidden Random Fields: Learning Discriminative Parts for Object Detection," *European Conference on Computer Vision*, pp. 302-315, 2006.
 - [78] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proceeding IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 886-893, 2005.
 - [79] T. Poggio and S. Edelman, "A network that learns to recognize three-dimensional objects," *Nature*, vol. 343, no. 1, pp. 263-266, 1990.
 - [80] H. Murase and K. Nayar, "Visual learning and recognition of 3-d objects from appearance," *International Journal of Computer Vision*, vol. 14, no. 1, pp. 5-24, 1995.
 - [81] A. Utsumi and N. Tetsutani, "Human Detection Using Geometrical Pixel Value Structures," *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 39-44, 2002.
 - [82] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 349-361, 2001.
 - [83] K. Okuma, A. Taleghani, N. D. Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: multitarget detection and tracking," *European Conference on Computer Vision*, vol. 1, pp. 28-39, 2004.
 - [84] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Tracking People by Learning

- Their Appearance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 65-81, 2007.
- [85] I. B. Ozer and W. H. Wolf, "A hierarchical human detection system in (Un) compressed domains," *IEEE Transactions on Multimedia*, vol. 4, no. 2, pp. 283-300, 2002.
- [86] A. Opelt, A. Pinz, and A. Zisserman, "A Boundary-Fragment-Model for Object Detection," *European Conference on Computer Vision*, vol. 2, pp. 575-588, 2006.
- [87] J. Shotton, A. Blake, and R. Cipolla, "Contour-Based Learning for Object Detection," *IEEE International Conference on Computer Vision*, vol. 1, pp. 503-510, 2005.
- [88] B. Wu and R. Nevatia, "Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors," *IEEE International Conference on Computer Vision*, vol. 1, pp. 90-97, 2005.
- [89] B. Wu and R. Nevatia, "Simultaneous Object Detection and Segmentation by Boosting Local Shape Feature based Classifier," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.
- [90] Y. Wu and T. Yu, "A field model for human detection and tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 753-765, 2006.
- [91] L. Zhao and C. E. Thorpe, "Stereo- and neural network-based pedestrian detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 3, pp. 148-154, 2000.
- [92] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 878-885, 2005.
- [93] L. Zhao and L. S. Davis, "Closely coupled object detection and segmentation," *International Conference on Computer Vision*, vol. 1, pp. 454-461, 2005.
- [94] A. Opelt, A. Pinz, and A. Zisserman, "Incremental learning of object detectors using a visual shape alphabet," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 3-10, 2006.
- [95] C. Cedras and M. Shah, "A survey of motion analysis from moving light display," *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, pp. 214-221, 1994.
- [96] Y. Guo, G. Xu, and S. Tsuji, "Tracking human body motion based on a stick

- figure model," *Journal of Visual Communication and Image Representation*, vol. 5, no. 1, pp. 1-9, 1994.
- [97] Y. Guo, G. Xu, and S. Tsuji, "Understanding human motion patterns," *Proceedings 12th IEEE International Conference Pattern Recognition*, pp. 325-329, 1994.
- [98] H. Nguyen, M. Worring, and R. Boomgaard, "Occlusion robust adaptive template tracking," *IEEE International Conference on Computer Vision*, vol. 1, pp. 678-683, 2001.
- [99] W. Long and Y. Yang, "Log-tracker, an attribute-based approach to tracking human body motion," *International Journal Pattern Recognition and Artificial Intelligence*, vol. 5, no. 3, pp. 439-458, 1991.
- [100] I. Chang and C. Huang, "Ribbon-based motion analysis of human body movements," *Proceedings IEEE International Conference Pattern Recognition*, pp. 436-440, 1996.
- [101] C. Wren and A. Pentland, "Dynamic models of human motion," *International Conference on Automatic Face and Gesture Recognition*, pp. 22-27, 1998.
- [102] C. Wren, B. Clarkson, and A. Pentland, "Understanding purposeful human motion," *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 19-25, 2000.
- [103] I. Haritaoglu, D. Harwood, and Davis.L.S., "Ghost: A human body part labeling system using silhouettes," *Proceedings IEEE International Conference Pattern Recognition*, pp. 77-82, 1998.
- [104] I. Haritaoglu, D. Harwood, and Davis.L.S., "W: Who? When? Where? What? ?A real time system for detecting and tracking people," *International Conference on Automatic Face and Gesture Recognition*, pp. 222-227, 1998.
- [105] I. Haritaoglu, R. Culter, D. Harwood, and Davis.L.S., "Backpack: Detection of people carrying objects using silhouettes," *Proceedings IEEE Interantional Conference Computer Vision*, pp. 102-107, 1999.
- [106] Y. Kuno and T. Watanabe, "Automatic detection of human for visual surveillance system," *International Conference on Pattern Recognition*, vol. 3, pp. 865-869, 1996.
- [107] S. Kurakake and R. Nevatia, "Description and tracking of moving articulated objects," *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, pp. 491-495, 1992.
- [108] J. Barron and I. A. Kakadiaris, "Estimating anthropometry and pose from a

- single uncalibrated image," *Computer Vision and Image Understanding*, pp. 268-284, 2001.
- [109] J. Barron and I. A. Kakadiaris, "On the improvement of anthropometry and pose estimation from a single uncalibrated image," *IEEE Workshop on Human Motion*, pp. 53-60, 2000.
- [110] R. Kahn, M. Swain, P. Prokopowicz, and J. Firby, "Gesture recognition using the perseus architecture," *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, pp. 734-471, 1996.
- [111] G. Rogez, C. Orrite-Urunuela, and J. Martinez-del-Rincon, "A spatio-temporal 2D-models framework for human pose recovery in monocular sequences," *Pattern Recognition*, vol. 41, no. 9, pp. 2926-2944, 2008.
- [112] X. Zhao and Y. Liu, "Generative tracking of 3D human motion by hierarchical annealed genetic algorithm," *Pattern Recognition*, vol. 41, no. 8, pp. 2470-2483, 2008.
- [113] R. Rosales and S. Sclaroff, "Specialized mappings and the estimation of human body pose from a single image," *IEEE Workshop on Human Motion*, pp. 19-24, 2000.
- [114] R. Rosales and S. Sclaroff, "Algorithms for inference in Specialized Maps for recovering 3D hand pose," *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 136-141, 2002.
- [115] R. Rosales, V. Athitsos, and S. Sclaroff, "3D hand pose reconstruction using specialized mappings," *IEEE international Conference on Computer Vision*, vol. 1, pp. 378-385, 2001.
- [116] D. C. Hogg, "Model-based vision: A program to see a walking person," *Image and Vision Computing*, vol. 1, no. 1, pp. 5-20, 1983.
- [117] K. Rohr, "Towards model-based recognition of human movements in image sequences," *Computer vision, graphics, and image processing - Image Understanding*, vol. 59, no. 1, pp. 94-115, 1994.
- [118] V. Pavlovic, J. M. Rehg, T. J. Cham, and K. Murphy, "A dynamic bayesian network approach to figure tracking using learned dynamic models," *Proceeding International Conference on Computer Vision*, pp. 94-101, 1999.
- [119] D. M. Gavrila and L. Davis, "3-D model-based tracking of humans in action: A multi-view approach," *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, pp. 73-80, 1996.
- [120] D. M. Gavrila and L. Davis, "Towards 3D model-based tracking and

- recognition of human movement: a multi-view approach," *Proceedings of IEEE International Symposium on Computer Vision*, pp. 253-258, 1995.
- [121] S. Wachter and H. H. Nagel, "Tracking persons in monocular image sequences," *Computer Vision and Image Understanding*, vol. 74, no. 3, pp. 174-192, 1999.
- [122] Q. Delamarre and O. Faugeras, "3D articulated models and multi-view tracking with silhouettes," *Proceedings of the Seventh International Conference on Computer Vision*, vol. 2, pp. 716-721, 1999.
- [123] J. Zheng and S. Suezaki, "A model-based approach in extracting and generating human motion," *IEEE international Conference on Pattern Recognition*, vol. 2, pp. 1201-1205, 1998.
- [124] J. Zheng and D. Takagi, "Interactive human motion acquisition from video sequences," *Computer Graphics International*, pp. 209-217, 2000.
- [125] H. Sidenbladh, M. Black, and D. J. Fleet, "Stochastic tracking of 3D human figure using 2D image motion," *European Conference on Computer Vision*, pp. 702-718, 2000.
- [126] H. Sidenbladh, M. Black, and L. Sigal, "Implicit probabilistic models of human motion for synthesis and tracking," *European Conference on Computer Vision*, vol. 1, pp. 784-800, 2002.
- [127] Z. Chen and H. J. Lee, "Knowledge-guided visual perception of 3d human gait from a single image sequence," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, pp. 336-343, 1992.
- [128] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 193-199, 1997.
- [129] R. Polana and R. Nelson, "Detecting activities," *Proceedings IEEE Computer Vision Pattern Recognition*, pp. 2-7, 1993.
- [130] R. Polana and R. Nelson, "Recognizing Activities," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815-818, 1994.
- [131] R. Polana and R. Nelson, "Low level recognition of human motion," *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 77-82, 1994.
- [132] K. Takahashi and S. Seki, "Recognition of dexterous manipulations from time-varying images," *Proceeding IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 23-28, 1994.
- [133] J. Yamato and J. Ohya, "Recognizing human action in time-sequential images

- using hidden markov models," *Proceeding IEEE Conference on Computer Vision and Pattern Recognition*, pp. 376-385, 1992.
- [134] R. Cutler and L. Davis, "View-based detection and analysis of periodic motion," *IEEE International Conference on Pattern Recognition*, pp. 495-499, 1998.
- [135] R. Cutler and L. Davis, "Robust periodic motion and motion symmetry detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 615-622, 2000.
- [136] T. Cootes, C. Taylor, D. Cooper, and J. raham, "Active shape models: their training and applications," *Computer Vision and Image Understanding*, vol. 61, pp. 38-59, 1995.
- [137] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *Proceedings European Conference on Computer Vision*, vol. 2, no. 484, p. 498, 1998.
- [138] A. Baumberg and D. Hogg, "An efficient method for contour tracking using active shape models," *Proceedings IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 194-199, 1994.
- [139] O. Chomat and J. L. Crowley, "Recognizing motion using local appearance," *Proceeding International Symposium on Intelligent Robotic Systems*, pp. 271-279, 1998.
- [140] I. Laptev and D. Lindeberg, "Space-time interest points," *Proceeding International Conference on Computer Vision*, pp. 432-439, 2003.
- [141] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of the Fourth Alvey Vision Conference*, pp. 147-151, 1988.
- [142] I. Laptev and D. Lindeberg, "Local descriptors for spatio-temporal recognition," *Proceeding W4 Workshop on Spatial Coherence for Visual Motion Analysis*, 2004.
- [143] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," *Proceeding International Conference on Computer Vision*, pp. 525-531, 2001.
- [144] G. D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Jorunal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
- [145] A. Bobick and J. Davis, "An appearance-based representation of actions," *Proceedings IEEE International Conference Pattern Recognition*, vol. 1, pp. 307-312, 1996.
- [146] A. Bobick and J. Davis, "Real-time recognition of activity using temporal

- templates," *Proceedings IEEE Workshop Applications of Computer Vision*, pp. 39-42, 1996.
- [147] J. Davis and A. Bobick, "The representation and recognition of human movement using temporal templates," *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, pp. 928-934, 1997.
- [148] J. Davis and A. Bobick, "Virtual PAT: A virtual personal aerobics trainer," *Workshop on Perceptual User Interfaces*, 1998.
- [149] A. Camurri, R. Trocca, and G. Volpe, "Interactive Systems Design: A KANSEI based Approach," *Proceedings of the NIME-02 Conference on New Interfaces for Musical Expression*, pp. 155-162, 2002.
- [150] A. Camurri, I. Lagerlof, and G. Volpe, "Recognizing emotion from dance movement: comparison of spectator recognition and automated techniques," *International Journal of Human-Computer Studies*, vol. 59, pp. 213-225, 2003.
- [151] T. J. Roberts, S. J. McKenna, and I. W. Ricketts, "Human pose estimation using learnt probabilistic region similarities and partial configurations," *European Conference on Computer Vision*, vol. 3024, pp. 291-303, 2004.
- [152] R. Ronfard, C. Schmid, and B. Triggs, "Learning to parse pictures of people," *European Conference on Computer Vision*, pp. 700-714, 2002.
- [153] A. Micilotta, E. Ong, and R. Bowden, "Detection and tracking of humans by probabilistic body part assembly," *British Machine Vision Conference*, vol. 1, pp. 429-438, 2005.
- [154] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Strike a pose: tracking people by finding stylized poses," *Computer Vision and Pattern Recognition*, vol. 1, pp. 271-278, 2005.
- [155] Q. Zhao, J. Kang, H. Tao, and W. Hua, "Part Based Human Tracking In A Multiple Cues Fusion Framework," *International Conference on Pattern Recognition*, vol. 1, pp. 450-455, 2006.
- [156] J. Costeira and T. Kanade, "A multi-body factorization method for motion analysis," *Proceeding International Conference on Computer Vision*, pp. 1071-1076, 1995.
- [157] M. Irani and P. Anandan, "Factorization with uncertainty," *Proceeding European Conference on Computer Vision*, pp. 539-553, 2000.
- [158] M. Machline, L. Zelnik-Manor, and M. Irani, "Multi-body segmentation: Revisiting motion consistency," *Proceedings Workshop on Vision and Modeling of Dynamic Scenes*, 2002.

- [159] L. Zelnik-Manor and M. Irani, "Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations," *Proceeding IEEE Conference on Computer Vision and Pattern Recognition*, pp. 287-293, 2003.
- [160] L. Zelnik-Manor and M. Irani, "Conditional feature sensitivity: A unifying view on active recognition and feature selection," *Proceeding European Conference on Computer Vision*, pp. 434-445, 2004.
- [161] M. D. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer, "Comparison of edge detectors: a methodology and initial study," *Computer Vision and Image Understanding*, vol. 69, pp. 38-54, 1998.
- [162] R. Nevatia and K. R. Babu, "Linear feature extraction and description," *Computer Graphics and Image Processing*, vol. 13, no. 3, pp. 257-269, 1980.
- [163] Y. Gao and M. K. H. Leung, "Face recognition using line edge map," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 24, pp. 764-779, 2002.
- [164] I. Biederman and J. Gu, "Surface versus Edge-Based Determinants of Visual Recognition," *Cognitive Psychology*, vol. 20, pp. 38-64, 1988.
- [165] *Libtiff TIFF library version 3. 0. Copyright Sam Leffler and Silicon Graphics Inc. Can be found at <ftp://ftp.sgi.com/graphics/tiff>., 1999.*
- [166] L. O. Gorman, "Binarization and multi-thresholding of document images using connectivity," *Symposium on Document Analysis and Information Retrieval*, pp. 237-252, 1994.
- [167] J. Chen, "Logo Recognition using Line Segment Hausdorff Distance," *School of Computer Engineering. Singapore: Nanyang Technological University*, 2002.
- [168] M. K. H. Leung and Y. H. Yang, "Dynamic two-strip algorithm in curve fitting," *Pattern Recognition*, vol. 23, pp. 69-79, 1990.
- [169] A. Rosenfeld and J. L. Pfalz, "Distance function on digital pictures," *Pattern Recognition*, vol. 1, no. 1, pp. 33-62, 1968.
- [170] K. Huang, L. Wang, T. Tan, and S. Maybank, "A real-time object detecting and tracking system for outdoor night surveillance," *Pattern Recognition*, vol. 41, no. 1, pp. 432-444, 2008.
- [171] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using non-parametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151-1163, 2002.

Appendix A Sequences Used in Our Experiment

In the following pages, five key frames from each test sequences of four sets of test images are displayed.

1. Sequences of the Lab Scene with Normal Lighting

The first test set was captured in a scene of an open area in a large laboratory with normal and fast activities like walking, moving chairs, fighting, crawling, etc.

2. Sequences of the Lab Scene with Dim Lighting

The second test set captured activities in a large laboratory in dim light environment.

3. Sequences of the Lab Scene with Light Changing

The third test set captured activities in a large laboratory in light changing environment.

4. Sequences of Robbery and Elevator Scenes with Dim Lighting

The last test set is a collection of low quality video from close-circuit cameras. It captured activities in two kinds of environment. One is inside an elevator and the other is TV collection of robbery scenes.

Appendix B Motion Segmentation and Motion Patterns Results

In the following pages, the results of motion segmentation and motion patterns are shown.

1. Motion Segmentation Results

Four groups of *MO* segmentation results of every 2 frames are displayed in the next four pages, in which one original frame is shown as reference at the top left.

Figure B-1: MO segmentation results of one person kicking (I16).

Figure B-2: MO segmentation results of one person sitting down (I33).

Figure B-3: MO segmentation results of three persons walking (I41).

Figure B-4: MO segmentation results of object-throwing (I24).

Figure B-5: MO segmentation results of one person kicking (I16).

Figure B-6: MO segmentation results of one person sitting down (I33).

Figure B-7: MO segmentation results of three persons walking (I41).

Figure B-8: MO segmentation results of two persons at night: the MO segmentation results are overlaid on the original images.

Figure B-9: MO segmentation results of two persons at night: the MO segmentation results are overlaid on the original images.

2. Motion Patterns with Vector Results

The motion pattern results are displayed from next page onwards, in which the images are laid out in five columns that are the original frame, temporal difference frame, $rMHI$, $fMHI$, and motion vectors overlaid on the original frame.

Figure B-10: Motion patterns with vector results of people falling down.

Figure B-11: Motion patterns with vector results of people kicking.

Figure B-12: Motion patterns with vector results of people waving arms.

Figure B-13: Motion patterns with vector results of people bowing.

Figure B-14: Motion patterns with vector results of people pushing a trolley.