Human activity recognition based on hidden Markov models

Liu, Xiao Hui

2006

Liu, X. H. (2006). Human activity recognition based on hidden Markov models. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/4747>

<https://doi.org/10.32657/10356/4747>

Nanyang Technological University

Downloaded on 26 Apr 2023 21:58:11 SGT

Human Activity Recognition Based on Hidden Markov Models

Liu Xiao Hui

School of Electrical & Electronic Engineering

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

2006

Acknowledgment

I would like to express my deepest appreciation to my supervisor, Dr. Chua Chin-Seng, for his guidance, patience and many invaluable comments and suggestions throughout the whole research project.

I also take this opportunity to thank Nanyang Technological University for the award of a research scholarship which enabled me to undertake this research.

I wish to acknowledge my family members who showed so much concern and care for me in the course of my studies. Their encouragement and unyielding support have contributed to the completion of this project.

Special thanks also go to all the people in the Intelligent Machine Research Laboratory, including Ren Ying, Wang Yingjie, Li Jiang, Zhang Wenbo, Li Xuchun and Chen Xujian, for their generous assistance provided throughout the entire project.

Contents

Acknowledgements	i
Contents	ii
Summary	vi
List of Figures	viii
List of Tables	xiii
List of Abbreviations	xv
1 Introduction	1
1.1 Motivations	3
1.2 Objectives of the Research	10
1.3 Major Contributions	11
1.4 Outline of the Thesis	13
2 Literature Review on Human Activity Recognition	15
2.1 Representation Approaches	17

2.1.1	Semantic Description	18
2.1.2	Motion-Based Approach	20
2.1.3	Appearance-Based Approach	24
2.2	Classification Approaches	30
2.3	Gait Recognition	34
2.3.1	Gait Period Detection	34
2.3.2	Spatio-temporal Analysis	36
2.3.3	Model-based Gait Recognition	39
2.4	Sign Language Recognition	40
2.5	Hidden Markov Model (HMM)	43
2.5.1	Likelihood Computation	46
2.5.2	HMM Training	48
2.5.3	Viterbi Algorithm	50
2.5.4	Using HMMs for HAR – An Example	51
3	Automatic Segmentation of Human Actions	63
3.1	Previous Work	65
3.2	The Background State of HMM	71
3.3	The Decoding Process	74
3.4	Validating the Segmentation Results	76
3.5	Experiments of Continuous Activity Segmentation	79
3.6	Summary	83

4	Rejection of Non-Meaningful Patterns Based on HMMs	86
4.1	Review of Related Works	88
4.2	Pairwise Likelihood Ratio Test	92
4.2.1	Definition of PLR	93
4.2.2	Grouping Multiple PLR Tests	97
4.3	Experiments with Simulated Data	100
4.3.1	LRT-Based Algorithms vs. PLRT	101
4.3.2	THMM vs. PLRT	104
4.4	Experiments with Real Data	109
4.4.1	Leave-One-Out Test	110
4.4.2	Group-based Test	114
4.4.3	Random Activity Test	116
4.5	Summary	118
5	Modeling Multi-Agent Activities	121
5.1	Related Work On Multi-Agent Activity Recognition	125
5.2	Observation Decomposed HMM (ODHMM)	127
5.2.1	ODHMM Structure	127
5.2.2	Parameter Estimation in ODHMM	130
5.2.3	Likelihood Computation	139
5.3	Experiments of ODHMM with Three-Person Interactions	143
5.3.1	Recognition Rate	148
5.3.2	Sensitivity Analysis	150

5.4	Recognition of Activities Involving Outliers	153
5.5	Experiment with Outliers	156
5.6	Summary	158
6	Conclusions and Recommendation	159
6.1	Conclusion	159
6.2	Recommendation for Further Research	163
	Author's Publication List	166
	Bibliography	168

Summary

Human activity recognition (HAR) is one of the most important areas of computer vision. The objective of HAR is to use computers to detect and recognize human activities automatically from a continuous image sequence. Such a technology is required by many application domains, such as visual surveillance, content-based retrieval and human computer interface.

There are several techniques that are important to an HAR system. These include video capture, human detection, feature extraction, activity segmentation, irrelevant pattern rejection and classification. The problems in these techniques must be carefully considered in order to improve the performance of an HAR system. This thesis focuses on three of these techniques — activity segmentation, irrelevant pattern rejection and classification.

The first contribution of this thesis is a background-state based activity segmentation framework. The focus is to find a way to label irrelevant observations in an activity sequence, without training some models that represent non-meaningful activities. This is realized by adding a background state to the HMM of each class of activities, whose associated density function is estimated from relevant observa-

tions instead of irrelevant ones, and the Viterbi decoding algorithm is applied using the new HMM. Since activity models are used in the segmentation process (the Viterbi decoding uses the activity model), the detected segments are consistent to our human understanding of activities.

The second contribution of this thesis is the rejection of irrelevant patterns. The variety of irrelevant activities makes it nearly impossible to define a generic model to represent all possible irrelevant patterns. The proposed rejection method is based on a modified version of likelihood ratio test (LRT), which is called a pairwise likelihood ratio test (PLRT). In PLRT, the distribution of the pairwise likelihood ratios is used as the rejection criterion so that the false alarm rate can be reduced. The advantage of PLRT is that models for irrelevant patterns are not required and the information required by the rejection process are obtained only from relevant patterns.

The third contribution of this thesis focus on another important HAR issue, multi-agent activity recognition. An observation decomposed hidden Markov model (ODHMM) is developed in this work. Different from the traditional HMM, ODHMM has multiple layers of observations, and a specific parameter representing each agent's role in the activity. ODHMM is suitable to deal with the changes in the number of the agents, and can provide a fast mapping of agents from the data set to the activity model. The detection of outliers in multi-agent activities is also presented, which uses the likelihood score of each agent for the hypothesized ODHMM as a reliability measurement.

List of Figures

1.1	The structure of a typical HAR system.	4
2.1	The MEI image (a) and MHI image (b) of a ‘lying down’ action. . . .	21
2.2	The ‘star’ feature used in [51].	25
2.3	The masks generated by the basis functions of ART on an 65 by 65 image.	27
2.4	A 4-state HMM with an ergodic state-transition matrix; white circles represent states and black circles represent observations.	44
2.5	Examples of tested activities - part I.	57
2.6	Examples of tested activities - part II.	58
2.7	Body shape sequences of activities used in experiments.	59
2.8	Three different person perform SitDown activity, their postures are different.	60
2.9	Examples of posture variations in activities that are performed by the same person.	60
2.10	State-transition structure of the left-right HMMs used in experiments.	61

2.11	The recognition rate of the trained HMMs versus the length of the sequences. The horizontal axis represents the percentage of frames that are used as input.	61
3.1	The HMM for Viterbi decoding in which alternative HMMs are embedded. $\{\lambda_i i = 1, 2, \dots, N\}$ is the set of meaningful HMMs, $\{\bar{\lambda}_j j = 1, 2, \dots, M\}$ is the set of alternative HMMs. S_0 and E_0 are pseudo beginning and end states respectively.	67
3.2	The structure of a HMM after adding the background state S_0	72
3.3	Viterbi decoding result of a sequence by using the JumpForward HMM with a background state; only a few key frames are selected (evenly extract from the original sequence) in order to reduce the number of shown frames.	76
3.4	Part of the detection result of an activity sequence containing two ‘punch activities’.	78
3.5	Parts of the used sequences. The black circles represent the beginning point of activities, while the white circles represent the end points . . .	84
3.6	ROC curves of the averaged true-detection rate (a), and the averaged accuracy versus the number of falsely detected activities (b).	85
4.1	Three simulated 2D Gaussian classes: 3D mesh of the probability densities of each class (a), and 2D view of the distribution of each class (b).	95

4.2	The mesh of log-likelihood ratio between C_1 and C_2 , $P(x C_2)/P(x C_1)$ (a) and the equipotential curves of $P(x C_2)/P(x C_1)$ in the original 2D space (b).	96
4.3	The acceptance region of C_1 provided by PLRT, as the double arrow shows (a) and the distribution of PLR vectors of C_1 , using C_2 and C_3 as the support classes, in the PLR space (b).	98
4.4	The topology of five simulated HMMs; each circle represents a state and its entry $(\cdot, \cdot)$ represent the mean of its output probability function.	101
4.5	Log-likelihoods of sequences from HMM_3 , HMM_4 and HMM_5 ((a), (b) and (c)) and their log-likelihood ratios computed by cohort-based LRT using an average function ((d)).	102
4.6	The sub state-space constructed by LRT-Cohort and the source of non-meaningful classes.	104
4.7	Log-likelihoods and THMM thresholds for (a) the sequences from HMM_4 and (b) the sequences from HMM_5 .	105
4.8	Settings of HMMs in the second simulated experiment.	106
4.9	Likelihoods of 20 samples from HMM_3 in the second experiment for THMM and HMM_3	107
4.10	Histograms of the Mahalanobis distances from the 100 PLRs of each class to the PLR-distribution center of HMM_3 .	109
4.11	Average ROC curves for (a) LRT-Cohort, (b) LRT-PBM and (c) PLR test based on leave-one-out test.	111

4.12	ROC curves of different number of support classes.	113
4.13	The ROC curves of (a) LRT-cohort, (b) LRT-PBM and (c) PLR test, using Group 1 as meaningful set, Group 2 as non-meaningful set. . .	115
4.14	The ROC curves of (a) LRT-cohort, (b) LRT-PBM and (c) PLR test, using Group 2 as meaningful set, Group 1 as non-meaningful set. . .	116
4.15	Examples of the 'unseen' activities	117
4.16	The ROC curves of (a) LRT-cohort, (b) LRT-PBM and (c) PLR test, using all activities as the meaningful set and random activities as non-meaningful.	119
5.1	Key frames of an activity simulating a multi-agent 'Snatch Theft'. . .	122
5.2	Traditional HMM: S_t and O_i are the the state sequence and observa- tion sequence respectively, where $t = 1, 2, \dots, M$	128
5.3	The Observation Decomposed HMM: o_t is decomposed into multiple layers, and each layer is associated with a role parameter r_j ($j = 1, 2, 3$ in this example); the role paramters are combined as a vector h	128
5.4	Trajectories of each player in the seven tested three-agent activities. .	144
5.5	The image sequence of Activity 5 (a), Activity 6 (b) and Activity 7 (c).146	
5.6	Feature extraction. A_t and B_t refer to the positions of A and B at time t	147
5.7	The likelihoods versus the lengths of a sample from Activity 3. . . .	150
5.8	Sensitivity analysis of the proposed model.	152

5.9 Likelihoods of agents in a tested activity. Agents 1, 2 and 3 are real
agents, and agents 4, 5 and 6 are outliers. 154

List of Tables

2.1	Summary of literatures of activity representation. (arranged according to the order of citation)	30
2.2	The estimated state-transition matrix A of the activity classes (1) SitDown, (2) StandUp, (3) Point, (4) LieDown, (5) LieUp, (6) SquatDown, (7) SquatUp, (8) BendDown, (9) BendUp, (10) Punch, (11) Kick, (12) JumpForward, (13) JumpUp and (14) JumpFrog.	56
2.3	The estimated output probabilities of the activity classes (1) SitDown, (2) StandUp, (3) Point, (4) LieDown, (5) LieUp, (6) SquatDown, (7) SquatUp, (8) BendDown, (9) BendUp, (10) Punch, (11) Kick, (12) JumpForward, (13) JumpUp and (14) JumpFrog.	62
3.1	Activity segmentation results. N_a is the number of activities in each video, N_f is number of the falsely detected segments and N_c is the number of the correctly detected segments.	81
4.1	Detection and rejection results of THMM	112

4.2	Rejection Result of THMM using Group 1 as meaningful set, Group 2 as non-meaningful set.	114
4.3	Rejection Result of THMM using Group 2 as meaningful set, Group 1 as non-meaningful set.	118
4.4	Rejection Result of THMM using all activities as the meaningful set and random activities as non-meaningful.	120
5.1	Agent Assignment Table.	142
5.2	Recognition Rate.	149
5.3	Experimental results with proposed outlier removing method.	157
5.4	Experimental results of selecting the most reliable person	157

List of Abbreviations

ABBREVIATIONS	FULL EXPRESSIONS
CHMM	coupled hidden Markov model
EM	expectation maximization
HAR	human activity recognition
HCI	human computer interface
HMM	hidden Markov model
LRT	likelihood ratio test
ODHMM	observation decomposed hidden Markov model
PBM	perturbed background model
PLR	pairwise likelihood ratio
PLRT	pairwise likelihood ratio test

Chapter 1

Introduction

Computer vision technologies have been in development for decades and their contents are vast and diverse, but the main purpose of computer vision has never changed since it was first introduced — to program a computer to ‘understand’ a scene or an image. This desire results in the blossoming of various computer vision technologies, such as fingerprint recognition [99], face recognition [149], image segmentation [176], medical image processing [50], object recognition [135] and content-based image retrieval [151], among many others. These computer vision technologies are making great changes to our lives. For example, in today’s factories, computers have already superseded human operators in the inspection of products on pipelines. Identification by one’s fingerprint is no longer a job requiring expert skills; even a trainee can perform like a veteran with the help of computers. Computers can also remember and recognize a vast number of faces, which is probably beyond the capability of humans (in terms of the number of faces that can be recognized). These

technologies are significant, but there are still more things that computers need to understand. A major direction to advance computers' intelligence is the expansion from static scene (image) understanding to dynamic scene understanding.

The demand for dynamic scene understanding stems from the increasing volume of video data we have to process everyday. Visual surveillance is one of the application domains that require computers to have the ability to understand dynamic scenes. The current industry on security-surveillance relies on the use of closed circuit television (CCTV) and employs thousands of human operators in tedious jobs watching videos that are mostly uneventful. It is difficult for these workers to focus their attention on a wall of security monitors hour after hour. A system that can analyze videos and monitor human activities in the scene is highly necessary for these error-prone security tasks. Such a demand is increasing rapidly around the world as anti-terrorism becomes a global affair. Another area requiring dynamic scene understanding is video processing. Producing and distributing video data are no longer professional jobs that require well trained personnel and expensive facilities as they were required 20 years ago. With the aid of broadband multimedia communications, we have access to a tremendous source of video data which requires a lot of effort to analyze, and filter their contents. Automatic video indexing and retrieval frameworks can help us make more efficient use of these video data. Human-computer interaction (HCI) is also an important domain where dynamic scene understanding can contribute to. Natural and intelligent interfaces to replace mouse devices, keyboards and joysticks are always of great value. Applica-

tions that benefit from natural HCIs include the control of electrical and electronic systems, advanced computer games, virtual reality environment and control of robot assistances.

Technologies for dynamic scene understanding include video segmentation, content-based video retrieval, object tracking and human activity recognition (HAR). This thesis focuses on the last technology, which is one of the most important parts of dynamic scene understanding in our everyday lives.

1.1 Motivations

HAR is a very important technology for video-content analysis which attracts increasing attention in the computer vision community. The recognition of human activities using computer vision technologies can contribute to both civil and military applications. For civil applications, HAR can facilitate surveillance systems to detect abnormal activities, extract highlights from sports video or movies, and provide natural interfaces for entertainments. For military usage, HAR can be used to recognize enemies' actions, predict their strategies and help commanders respond quickly to the situation on the battle field. This project is motivated by the wide potential of applications of HAR. Before we go into the details of HAR, it is necessary to introduce briefly the structure of HAR systems.

A typical HAR system, which is shown in Figure 1.1, has six main components — video capture, human detection, feature extraction, segmentation, irrelevant pattern rejection and classification. The process to recognize a human activity is as follows.

Figure 1.1: The structure of a typical HAR system.

Videos are first captured by cameras and stored as image sequences. Human detection is then applied to each image to extract human subjects. The extracted human subjects are stored in a foreground image sequence and delivered to the feature extraction module. After that, the foreground image sequence is converted to

a sequence of feature vectors representing the human subjects in each image. The segmentation module is responsible for dividing the sequence of feature vectors into small segments, each of which is according to the boundaries of a certain human activity. Some non-meaningful activities may also be segmented and their presence can decrease both the efficiency and accuracy of the HAR system. Therefore, in the next step, every segment is validated according to their contents and the irrelevant ones are discarded. In reality, the segmentation module and the irrelevant pattern rejection module are often intertwined rather than separated. The validation result can be used to refine the segmentation result. This is shown in Figure 1.1 as a feed-back loop between the two modules. The cleaned activity segments are then passed to the classification module where each activity segment is classified into one of the known categories. The final report about the human activities in the video is generated based on the classification results. In real-life applications, the system structure may be more complicated than the one presented above. For example, if multiple persons exist in the scene, a tracking module is necessary and has to be added between the modules of detection and feature extraction.

Each component of the HAR system has its own problems. The problems in the video capture module, including capturing speed, image quality and transmission bandwidth, are rarely considered as computer vision problems, whose solutions depend on the development of hardware. It is the rest of the five modules that are targeted by computer vision technologies and are elaborated in the following parts.

Human detection is the basis of HAR, which extracts human subjects from

complex backgrounds. To detect human objects from a texture background is still a difficult task [55], and it usually requires extensive computations. Instead, in real applications, human detection is often simplified to the detection of moving targets in the scene. Moving targets can be detected by either background subtraction or motion analysis techniques [101, 145], which are much easier to be implemented in HAR systems. However, moving targets can also be vehicles and animals, a method to distinguish those targets from humans is necessary. Assumptions about the shape and size of the body are often used to decide whether a detected blob represents a human subject. The consequent drawback of this method is that it is application-dependent and its detection rules must be carefully adjusted in different application environments. Recently, researchers are interested in seeking more sophisticated human detection methods which have less or no dependency on application environments [35, 93, 152, 164]. Machine learning techniques are often used to assist the detection process. But those methods are still too complex to an HAR system, their accuracy and efficiency need to be improved.

Feature extraction has long been an important topic in HAR. The difficulty in feature extraction is caused by view variations of HAR systems. Images captured from a typical human activity are 2D projections of a series of 3D body postures. In real-life applications, the view directions of the cameras varies in different application environments and it is unreasonable to take the assumption that the observed 2D projections of human bodies are identical across these environments. So far, it is extremely difficult, if not impossible, to obtain highly discriminative 2D features

from human activities which are robust to variations in both body pose and projection. A 3D feature of human body can deal with the view variation problems. However, finding the correspondence between 3D features, such as joint angles and positions, from different frames is the key issue to be solved. There is still a long way to go before generic features can be defined for human activities under various viewpoints and environments. A detailed review of the representations of human activities is given in Section 2.1.

Automatic activity segmentation is another important issue in HAR systems. To recognize an activity, the beginning and end points of the activity must be located. The difficulty in finding these points lies in that human activities transit from one to another smoothly rather than abruptly. Some activity detection methods have been proposed based on analyzing the continuity of scene properties such as motion, color histogram or texture [4, 72, 178, 179]. These scene properties are not direct measurements of activities; hence, the detected activity boundaries may be not consistent with the concepts of human logic. The detected segments may contain more than one activity, or may contain only a part of an activity. The problem to be solved in activity segmentation is how to locate the boundary points accurately. Prior knowledge about the activities to be detected is necessary for this purpose. An activity model, such as an HMM, is a kind of prior knowledge of the activity. The way to combine such prior knowledge with a segmentation scheme is an open problem. In this thesis, a segmentation framework based on the Viterbi decoding of HMM is presented in Chapter 3.

Irrelevant pattern rejection plays an important role in HAR systems since most of the observed activities are irrelevant or non-meaningful. The early efforts on HAR research mainly focused on the classification issues [25, 26, 168]. Various representation and modeling methods are proposed and tested with selected databases. Usually, these tested databases are constructed only with ‘meaningful’ activity patterns, which means that every single pattern within the database belongs to one and only one known activity class. Therefore, there is only one type of classification error in these tests — the insertion error, indicating that a pattern is classified to a wrong class. Such tests are useful in evaluating the performance of the recognition algorithms, since they could show how good the tested algorithms can distinguish patterns from different classes. However, in real-life applications, such kinds of tests are not sufficient to measure the real performances of HAR systems. This is because the input patterns of an HAR system will never be composed purely of meaningful patterns. Irrelevant patterns always exist in company with meaningful ones. A useful activity recognition system must have the ability to distinguish meaningful patterns from non-meaningful ones as well as classify each meaningful pattern into the correct category. Some methods have been proposed to reject non-meaningful patterns (See Section 4.1 for references). But a prerequisite of these methods is that a model representing the entire non-meaningful pattern space must be obtained. Such a non-meaningful model is very difficult to build since we cannot predict all possible non-meaningful patterns and feed them to the model establishing process. A interesting problem of rejection is how to find a way to reject non-meaningful pat-

terns without requiring any prior knowledge of the characteristics of non-meaningful patterns. A new rejection algorithm is presented in Chapter 4, which uses only the information of known activities.

Classification of activities is the ultimate goal of HAR systems. In the past two decades, various classification methods have been used for individual activity recognition [13, 19, 39, 63, 154, 168]. Recently, the interactions between multiple persons are recognized as a significant source of information for the analysis of video content. The recognition of these interactions is called multi-agent activity recognition. A major problem in classification of human activities is the shortage of modeling methods for multi-agent interactions. Some models for multi-agent activities have been proposed but most of them concentrate mainly on the interactions between two persons and they are difficult to be extended to model activities containing more persons. The difficulties in modeling multi-agent activities stem from the fact that when the number of involved persons increases, the interactions between agents are much more complicated to model. This calls for new modeling methods that can give a concise and extendable representation of multi-agent human activities. Moreover, some persons in the scene may not play roles in an multi-agent activity — these persons may be considered as outliers — and they must be eliminated beforehand. It is difficult to distinguish these outliers from ‘true’ agents since the outliers are not significantly different from true agents. In order to detect outliers, each person’s activity and his/her interaction with others have to be analyzed. This is time consuming and slows down the processing speed. Efficient methods to locate

outliers quickly are needed. To overcome these problems, an ODHMM algorithm is presented in Chapter 5 for multi-agent activity recognition and outlier detection.

1.2 Objectives of the Research

The contents of this research include the last three modules of an HAR system described in Figure 1.1, namely segmentation, irrelevant pattern rejection and classification. The key objectives of this thesis can be summarized as follows:

- To develop an automatic segmentation framework that can divide continuous video into short segments according to the activity content. The segmentation method should be capable of using activity models to assure the accuracy of activity boundary detection.
- To investigate the rejection method of non-meaningful patterns which can be established using only meaningful patterns and requiring no explicit models for non-meaningful patterns.
- To design a multi-agent activity modeling method that can handle a large number of agents. An outlier detection process should be developed together with this modeling method.

1.3 Major Contributions

This thesis discusses the establishment of a framework for human activity recognition. Throughout this thesis, solutions for the different modules of the HAR system, including segmentation, irrelevant pattern rejection and classification, are presented. The proposed methods are based on hidden Markov models (HMMs) which are widely adopted by both speech and computer vision communities. The major contributions of this thesis are as follows:

- A *background-state* based method for automatic segmentation of activities, based on the Viterbi decoding. The main difficulty in activity segmentation is the labeling of irrelevant observations inside a sequence. A background state is added to the HMM of each activity of interest which gives labels to irrelevant observations in the input sequence. Differing from other Viterbi-decoding based spotting methods [100, 130], the parameter of the background state in our method is not trained by a set of selected irrelevant observations but trained by the meaningful observations only. Similar to our method of irrelevant-pattern rejection, irrelevant observations are not required in establishing the segmentation system which is desirable for most of HAR applications, since it is also very difficult to define a representative model for all possible irrelevant observations.
- A *pairwise likelihood ratio* (PLR) test to reject non-meaningful activity patterns. One open problem in non-meaningful activity rejection is the establish-

ment of a non-meaningful model [129, 161]. One advantage of the proposed test is that it does not require any explicit non-meaningful model. For each meaningful class, every other meaningful class is viewed as a support class which helps decide if a pattern should be accepted. For each support class, a novel test, named pairwise likelihood ratio test (PLRT), is used to measure the relevancy of a pattern to a model. Another contribution of PLRT lies in that, instead of using a truncating threshold as in the traditional likelihood ratio test (LRT) [61, 125, 131], PLRT makes its decision according to the distribution of the pairwise likelihood ratios. By combining multiple PLRTs, this measurement can reduce the acceptance region and enhance the rejection accuracy.

- An *observation decomposed hidden Markov model* (ODHMM) for the modeling of multi-agent activities. An important issue in multi-agent activity recognition is the recognition of the role of each agent. ODHMM has multiple layers of observations, where each agent has a particular observation layer, and a role parameter is added to the model which represents the alignment between an agent in the data set and an agent in the activity model. Compared with traditional HMM, ODHMM requires less human effort in designing the state transitions, has fewer parameters to estimate, provides a comparable recognition performance and is not sensitive to the loss of tracking of the agents. Based on ODHMMs, an outlier detection method is also proposed which uses the likelihood score of each agent for an ODHMM to measure its relevancy to

the model.

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

1.4 Outline of the Thesis

Chapter 2 presents a literature survey of HAR technologies. Different representation approaches of human activities are first presented. After that, the classification approaches of HAR are reviewed. Then two specific application domains of HAR — gait recognition and sign language recognition — are introduced. A short introduction of hidden Markov models (HMMs) is also given in the final part of Chapter 2, since our work is based on HMMs. An example of using HMM for human activity recognition is provided to help readers understand how HMM works in human activity recognition.

Chapter 3 presents an automatic video segmentation framework based on a background state of HMMs. A brief review of related works in both speech and activity recognition is first presented. The proposed segmentation framework based on the background state is then presented. The decoding of the HMM having a background state is discussed in detail. In order to reduce the false alarm rates in segmentation, a validation process of the segmentation results is incorporated into the segmentation framework. The proposed segmentation framework is tested with an individual activity database.

In the validation process of the segmentation results, a pairwise likelihood ratio

(PLR) test is used. Chapter 4 presents the details of this algorithm. In this chapter, existing algorithms for non-meaningful pattern rejection are reviewed and drawbacks of these algorithms are discussed. After that, the definition of PLR and the criteria for rejection are presented. In order to enhance the rejection accuracy, a grouping method of different PLRTs is also proposed. The performance of the proposed algorithm is compared with those of other algorithms, for both simulated and real data.

Chapter 5 discusses the modeling of multi-agent activities. The existing multi-agent activity recognition methods are first discussed in this chapter. The proposed observation decomposed hidden Markov model (ODHMM) is introduced in both discrete and continuous forms. Issues such as parameter estimation and likelihood computation of ODHMM are discussed in details. Experiments with seven street activities are conducted to show the performance of ODHMM for multi-agent activity recognition. The sensitivity of our method in case of incomplete data is also presented. An automatic outlier detection method based on ODHMMs is also discussed since some agents in the scene may not play any role in the activity.

Conclusions and Future research directions of this work are presented in Chapter 6.

Chapter 2

Literature Review on Human Activity Recognition

Human activities are one of the most important sources of information in video data. The recognition of human activities, such as gesture, gait and action has a wide range of applications. Surveys related to this topic can be found in [3, 55, 66, 103]. Typical application domains that benefit from HAR technologies include visual surveillance, content-based video retrieval, and natural human-computer interface (HCI).

Many promising projects have already been done for visual surveillance systems such as [31, 59, 110]. Analysis of human activities can be used to detect abnormal or dangerous events, interpret meanings of behaviors and identify persons. With the technology of HAR, security guards can be ‘liberated’ from the tedious job of keeping their eyes upon the surveillance video all the time. The guards only need to respond to the alarms that are generated by computers; hence they have more time

*CHAPTER 2. LITERATURE REVIEW ON HUMAN ACTIVITY
RECOGNITION*

16

to solve security issues perfectly. Automatic video indexing and retrieval also benefit from HAR significantly. Compared with earlier video processing technologies such as shot detection [58, 67, 117], shot clustering [54, 147] and video type classification [37, 150], HAR-based video indexing methods can provide users with more detailed information about the contents of interest. For example, in a tennis game, the event detection signifies when a player performs a 'baseline play' while the activity recognition can further reveal whether a fore-hand or back-hand stroke is performed by the player. HAR is also important for the development of advanced HCIs. Many novel approaches to establish natural HCIs have already been seen, such as eye gaze [112], face expression [29, 172], body posture [16] and hand gesture [90, 118]. As the HAR technology develops, people will have more and more convenient and natural ways to communicate with computers, machines and robots.

This chapter presents a literature survey of HAR technologies. Section 2.1 presents different representation approaches of human activities. The classification approaches of HAR are reviewed in Section 2.2. Two specific application domains of individual human activity recognition — gait recognition and sign language recognition — are introduced in Section 2.3 and Section 2.4, respectively. A short introduction to hidden Markov models (HMMs) is given in Section 2.5, on which this thesis is based. An example of using HMMs in human activity recognition is also presented in this section.

2.1 Representation Approaches

The definition of human activity is very broad. Hand gesture, body gesture, walking style, movement trajectory and interaction between people can all be considered as human activities. Human activities are characterized by patterns of motion and dynamics of posture changes, based on which three main approaches have been explored in representing human activities. The first approach is semantic behavior description, which relies on the interacted events between humans and target objects to interpret the meaning of human activities. Semantics are very effective in dealing with activities that have no unique patterns in motion or poses. For example, the activity ‘making a phone call’ can simply involve two events: ‘the object approaches the phone’ and ‘the object is attached to the phone for a while’. It is impossible to use motion or dynamics of body poses to recognize such an activity, since too many possible movements may result from a person when making a call. The limitation of this approach is that it requires knowledge about the locations and the states of the objects that interact with people. It is also incapable of recognizing activities in which people do not interact with other objects. The second approach is based on motion. In this approach, activities are described by trajectories or motion patterns of body parts. This approach is particularly effective when the meaning of activities depend more on moving parts of the body, such as hand gestures. Since only the body parts having significant motion are considered for representation, the redundancy in the original data can be discarded which makes it easier to store and retrieve the data. The disadvantage of motion-based representation is that the

computational cost for motion, such as optical flow, is very high. Even though the extraction of motion trajectories does not necessarily require the computation of optical flow, it is a prerequisite to locate different body parts to derive their motion trajectories, which is difficult in applications when body parts occlude each other. To avoid these complexities, the third approach uses appearances — the shapes of full bodies — as the representation of activities. Using the appearance for representation reduces the complexity of tracking algorithms and saves processing time. Nevertheless, the disadvantage of appearance-based approach is that the appearances are highly view-dependent whereas the motion-based approach is less vulnerable to view variations. These representation approaches are discussed in the following sections.

2.1.1 Semantic Description

In semantic description approach, it is assumed that the activities to be recognized are composed of one or more simple events. These events are normally detected by analyzing either the status of people and objects, or the interaction between people and objects. For example, the event ‘reach’ can be detected by observing the decrease of the relative distance between a person and an object, and the event ‘touch’ is detected if an object is occluded by a close person. The recognition of certain classes of activities is realized by modeling the relationships among these simple events. Semantic description can help convert a continuous video into a narrative report, such as ‘A man went into the room, stayed for 5 minutes, then

left’.

Ayers et al. [6] established a system which detected persons and position changes of the objects that could interact with people. Simple events were further induced semantically. For example, the ‘use terminal’ event was detected by observing ‘person near the monitor’ and ‘scene change in mouse’. A conditional state-transition model was used to convert long activity sequences into textual expressions. Izumi et al. [76] proposed a natural language description of human behaviors. Production rules were used to connect these events into consecutive expressions. Moore et al. [104] proposed an architecture named ‘ObjectSpace’ which was composed of three layers — extraction layer, object layer and scene layer. Human hands and objects were detected through the first two layers. Behaviors were inferred by the top layer using the Bayesian classification method. Pinhanez et al. [120, 121] proposed a PNF (past, now and future) network, which is converted from a manually codified *interval algebra constraint network*, to model the temporal relations among events. In the PNF network, activities were represented hierarchically, where the nodes at the top level represented complex activities while nodes at the bottom level represented sub-activities.

Semantic description cannot be used for activities where no simple events can be defined, though its effectiveness has been seen in earlier attempts. More information, such as motion and appearance of a body, need to be considered to represent complex activities where semantic description is incapable.

2.1.2 Motion-Based Approach

The research on analyzing human motions for activity recognition can be dated back to 1973. Johansson [78] found that observers could instantly recognize human motion patterns where the only visible features were a few moving dots attached to human bodies, which are called moving light displays (MLDs). Based on a similar procedure, Cutting and Kozlowski [34, 86] further showed that human gait and sex could also be identified from MLDs. Much later, Ditterich [38] examined more complex activity categories including locomotory, instrumental and social actions with MLDs. These works triggered the question as to whether it is possible to model and recognize human activities from motions. A detailed review of MLD-related works can be found in [27].

Two theories are commonly accepted in the interpretation of MLD-type methods. The first one is that people use MLD motions to recover the 3D structure of human body and use the structure for activity recognition [159]. The second one is to use the motion information of MLDs directly without intermediate processing. It is difficult to detect human joints and their relative angles using computer vision techniques instead of MLDs, but the theory of using motion directly for recognition is of great interest. There are two types of motion that are widely adopted – the global motion and the motion trajectory.

The global motion is constructed by the motion of each individual pixel within the body, describing the movements of body pixels between two consecutive images. Pixel-wise optical flow [11, 96] is the most widely adopted technique for motion

Figure 2.1: The MEI image (a) and MHI image (b) of a ‘lying down’ action.

capture. Since the dimensionality of the global motion is high, the main task in activity recognition methods based on global motion is to extract compact features from the global motion to represent each frame. For example, Babu [8] divided the image into different regions under polar and Cartesian coordinates and quantized the global motion into discrete features so reduce the dimensionality of the feature space. Efros et al. [39] split the flow field of a human body into four energy fields according to its signed horizontal and vertical components, and use the blurred energy image of each field to represent the global motion.

Beside pixel-wise optical flow, other versions of global motion are also adopted. Bobick and Davis [19] proposed a representation of global motion, named motion history image (MHI). MHI is different from optical-flow motion description in two aspects. Firstly, a MHI only describes the place and the time when motion occurs whereas an optical-flow measures each pixel’s movement in detail. Secondly, an image sequence produces only one MHI whereas optical flow produces $N - 1$ motion images, where N is the length of the image sequence. Figure 2.1 shows a MHI of

an ‘lying down’ action. Moment features were used to measure the similarity of two activity sequences based on the derived MHIs. The main advantage of MHI lies in its highly compact representation of global motion. Experimental results showed that the recognition rate was high using MHI, and the system had a 30 degrees tolerance of change in viewing direction. Babu and Ramakrishnan [9] combined a motion flow history, which describes the spatial distribution of motion vectors over the video clip, with MHI to provide a more detailed representation of activity sequences in MPEG videos. Erol and Kossentini [43] proposed three local motion descriptors based on the angular and circular area variances of the video object, the variances of the angular radial transform (ART) coefficients of the object’s area and the variances of the Fourier coefficients of the object’s boundary. The advantage of their motion descriptors is that features can be derived using binary shape masks of video objects, or computed directly in the MPEG-4 compressed domain.

Another popular approach is to represent activities with motion trajectories. Motion trajectories can be constructed from the moving path of the centroid of a person or from a set of points representing the centroids of some body parts [22]. Compared with global motion, the motion trajectory is much more compact preserving only the most important information. Although a large amount of information is lost, previous efforts have shown that the motion trajectory presentation is very effective in many application fields, such as the recognition of gestures [18, 26, 144, 162], actions [98, 123, 169] and interactions [79, 107, 115]. The feature extraction in motion-trajectory based methods is relatively simpler than that in global-motion

based methods. From each frame, a vector representing the position of the centroid of interest and its gradient are always derived as features. Campbell et al. [26] compared the performance of different combinations of motion trajectory features for gesture recognition. Compared with global motion, the detection and recovery of motion trajectories are much easier and more robust than that of global motion, which is desirable for real-time applications. A potential advantage of motion-trajectory based methods over global-motion based methods is that it is possible to develop view-invariant activity recognition methods with motion-trajectory analysis. Cao et al. [123] proposed a method to recognize simple hand actions, such as the ‘opening/closing of a cabinet’. A hand-trajectory was represented by a sequence of dynamic instants and intervals. The dynamic instant represents an important change in the hand motion, and the interval represents the time-period between any two dynamic instants. Such representations do not depend on the viewpoints; hence the recognition system is view invariant.

A disadvantage of a motion-based approach is that its computational complexity can be very high in some applications. The computational cost of detecting global motion, for example based on optical flow, is very high if the the number of processing blocks or pixels is large. The motion trajectory representation can also be very complex, since different body parts need to be located, which is extremely difficult when they occlude each other. The appearance-based representation is an alternative representation of human activities, which avoids these computational complexities.

2.1.3 Appearance-Based Approach

Intuitively, an activity of a person can be viewed as a sequence of body postures that change orderly. Consequently, the recognition of a human activity can be divided into two consecutive processes. In the first process, body postures of the input activity are detected and recognized from each frame, and a sequence of body postures is formed. In the second process, the posture series are compared to reference prototypes or statistic models of known activities for classification. There are several advantages of the appearance-based approach over the previously mentioned motion-based approach. Firstly, the appearance-based approach requires less computation in representing human actions. In the appearance-based approach, the detected human bodies, normally in the form of silhouettes, can be directly used, whereas the motion-based approach requires further processing to derive motion information from the detection result. Also, the appearance-based approach does not require any extra processing to locate body parts of interest. Secondly, appearances provide extra information which are discarded in the motion-based approach. For example, a person waving his hand from one side to the other repeatedly means ‘good-bye’. Motion information is very effective in representing such an action. However, the appearances of the person can further reveal whether he says ‘good-bye’ to his friend in a standing or sitting posture.

The earliest attempt to recognize human actions with appearance information can be dated back to 1991. Yamato et al. [168] first used appearance information to recognize tennis actions. The appearance of each body posture was represented

Figure 2.2: The ‘star’ feature used in [51].

as a vector which is formed by a 10 by 10 mesh. The large dimension of the mesh feature caused the dimensional curse and resulted in a low recognition rate. Even though, Yamato’s approach inspired others to explore the usefulness of appearance information.

There are two approaches to derive representations from appearances. The first approach is a non-model-based representation, in which features extracted from the human appearances do not incorporate prior knowledge of human-body models. Fujiyoshi and Lipton [51] proposed a ‘star’ feature to represent human postures. The ‘star’ was a simplified skeleton derived from the contour of human bodies. The distances from the contour points to the centroid of the body were computed. The stars are constructed by points that are local maximums in the distance array. Figure 2.2 shows an example of the ‘star’ representation. The orientations and periods of the stars were used to recognize human gaits. A similar approach to extract features for representing human postures was proposed by Galata et al. [52]. Instead of selecting only a few extremities, a fixed number of control points of the contour of a human

body is used. A variable length Markov model is proposed to model human behaviors based on these control points. Elgammal et al. [42] also use contours to represent human postures. They proposed an exemplar-based method to recognize human gaits. Zivkovic et al. [180] select a set of features to recognize tennis strokes based on segmented binary body images. They defined a pie feature to represent binary shapes, which represents the percentage of the binary shape contained in a circle, cut out centered at the mass center. Petkovic et al. [119] further extend the feature set containing 27 kinds of features, and they tested these features with recognition techniques including spatio-temporal formalization, HMM and Bayesian network to determine the best feature sets. Angular radial transform (ART) [1, 46, 64, 126] is a regional based shape descriptor which is accepted in MPEG-7. In ART, a set of orthogonal moment basis functions is defined on a unit disk in polar coordinates (ρ, θ) . An example of the basis functions is shown in Figure 2.3. The basis functions are separable along the angular and radial directions. The ART of a binary shape is the inner product of the basis functions and the shape image over the unit disk. Since the basis functions are orthogonal, the ART coefficients render no redundancy and provide a compact representation of binary shapes.

A limitation of the 2D non-model-based approach is that it is view dependent. To achieve a representation which is robust against the change of views, a 3D non-model-based representation is necessary. 3D pose can be inferred directly from the silhouette extracted from a 2D image. A direct 2D-3D pose estimation approach is the exemplar-based approach [105, 139]. In this approach, a number of exemplar 2D

Figure 2.3: The masks generated by the basis functions of ART on an 65 by 65 image.

views of the human body in different body configurations and viewpoints are stored in the database. the test body feature is matched to each stored exemplar, the pose is determined by the most similar stored exemplar. Mori and Malik [105] sample a number of points from the edge of human bodies and a shape context matching is used to measure the similarity between a test image and an exemplar. The pose of test body shape is determined by the exemplar with the lowest matching cost. A limitation of such a method is that when the number of exemplars is large, the search for the correct exemplar is very time consuming. Shakhnarovich et al. [139] introduced a hashing-based search technique to find the correct exemplar, where only a subset of the total exemplars need to be examined. Agarwal and Triggs [2] use a sparse Bayesian nonlinear regression to extract compact models with good generalization from a large database, where the direct searching can be avoid. Cohen and Li [30] integrate silhouettes captured by multiple cameras into a body hull, and use SVM to classify human postures based on the derived body hulls.

A sequence of continuous human postures can also be viewed as a sequence of high dimensional vectors (such as silhouettes) that lie on a low dimensional manifold.

Human pose can be determined by projecting the high dimensional vector to a learned low dimensional manifold. Brand [23] used hidden Markov model to learn the system's dynamical manifold. The pose is estimated by projecting sequences of input to the learned manifold. Beside learning the mapping from input to manifold, Elgammal and Lee [41] also learn the mapping from manifold to 3D pose, so that poses can be recovered from 3D-pose interpolation even if they are unseen.

Different from others, Rosenhahn et al. [83, 132] proposed a 2D-3D pose estimation method based on the free-form contours and surfaces. They applied a spectral domain representation (using DFT or 2D-DFT) of 3D contours and surfaces. Poses are estimated by combining the spectral representation with a version of ICP algorithm [171].

The second approach is a model-based representation, in which features are extracted from appearances guided by prior knowledge about the structure of human bodies. Popular human models used for activity recognition include the stick model, rectangle/cylinder model and triangulated-polygon model.

The stick model is a widely used 2D human model [15, 28, 75, 113]. A stick model is composed of points representing human joints and line-segments connecting these points representing torso, legs and arms. Bharatkumar et al. [15] used medial axes transform to recover the lower-limb stick figures. The stick figures were used to construct a generic kinematics model for gait analysis. Iwasawa et al. [75] proposed a real-time method to extract stick figures. The orientation of the upper half of the body was first calculated based on the centroid of the body image and the inertia

moment. Points representing the head, hands and feet were located heuristically and joints such as elbows and knees are detected by genetic algorithm (GA).

The rectangle/cylinder model is a well-known 2D model for human bodies [13, 48, 65, 94, 158, 163]. Principle parts of the body, such as head, torso, upper and lower arms, thighs and calves, are represented by rectangles, cylinders or blobs. Human bodies are first divided into segments according to pixel-level similarity in color or texture. The rectangle model is built by searching for the global optimal fits from the segmented parts. Wren et al. [163] proposed a method to model human bodies by blobs which were described by spatial Gaussian models. Kalman filters were used to predict and track each blob over time. Hidden Markov models were used by Felzenszwalb and Huttenlocher [48] to find the optimal alignment. Hu et al. [65] proposed a genetic-algorithm based searching methods to find the best fitting segments. Ben-Arie et al. [13, 158] proposed an expansion matching (EXM) algorithm which expand the signal with respect to shifted versions of a template. The expansion coefficients for a particular location indicate the extent of the presence of the template.

Another well-noticed human model is the triangulated-polygon model [47, 142, 175], in which the human subject is described by deformable triangles. Song et al. [142] proposed an automatic learning method of the graphic model using unlabeled data. The learning algorithm views known body part assignments as hidden variables to be estimated from unlabeled data using the EM algorithm. Zhang et al. [175] use a Bayesian graphic model, whose nodes correspond to point positions of

body outlines. The inference of this network is operated by sequential Monte Carlo sampling methods. Dynamic programming (DP) was employed by Felzenswalb [47] to find the global optimal match of a triangulated-polygon model to the image.

Although the model-based representations of human appearances are structural and easy to analyze, their computational cost are much higher than that of non-model-based representations. The best representation of human activities is still an open problem. The representation approaches reviewed in this thesis are summarized in Table 2.1.3 for a quick review.

Table 2.1: Summary of literatures of activity representation. (arranged according to the order of citation)

	Semantic	Motion		Appearance		
		Gloabl	Trajectory	non-model	2D-3D	model
Ref.	[6], [76], [104], [121], [120]	[96], [11], [8], [39], [19], [9], [43]	[22], [26], [18], [144], [162], [169], [98], [123], [79], [107], [115]	[168], [51], [52], [42], [180], [119], [64], [46], [126]	[105], [139], [2], [30], [23], [41], [83], [132]	[28], [15], [113], [75], [94], [163], [48], [65], [158], [13], [142], [175], [47]
Cost	low	high	medium, high	low	high	medium, high

2.2 Classification Approaches

For activity modeling, two approaches have been widely used, the template-matching approach and the state-space approach. In the template-matching approach, the image sequence is converted into a static template which is then compared with the reference of an activity. In the state-space approach, each frame in the image sequence is associated with a static state. The whole image sequence then can be

translated into a chain of these states connected by certain probabilities. The classification of the activities is executed by computing the joint probabilities through these states.

The most outstanding work based on template matching was done by Bobick and Davis [19]. They extracted Hu-moments from their proposed motion history images (MHIs), which is only a 7-dimensional vector. This highly compact template is representative for human actions viewed from a fixed direction. Frame-to-frame similarity is used by Efros et al. [39], where the similarity is computed by a normalized correlation. The similarity between two activity sequences are compared by a weighted sum of the frame-to-frame similarities. J. Ben-Arie and Zhiqian Zhang [13] proposed a multi-dimensional indexing method for human activity recognition. The human body was represented by nine cylinders for the torso, upper arms and legs (left and right), lower arms and legs (left and right). The likelihood between the given image sequence and a reference activity template is computed by integrating the votes of a number of key frames.

The advantage of template matching is its simplicity and low computational cost. However, the performance of template matching approaches depends on the robustness of the selected features, which should be invariant to the time durations of activities. When people perform a certain activity, the time duration of an activity may vary significantly from time to time. Certain parts of the activity may be prolonged while some other parts may be shortened. Even the full execution time of the activity can be different greatly from time to time. The time duration is an

important issue to solve for template-matching algorithms. Before extracting the activity template, activity boundaries need to be detected and frames need to be removed or interpolated in order to produce templates with the same time scale.

In order to avoid complex re-scaling processing, the state-space approach is often adopted, where an activity is viewed as a chain process visiting a finite number of states. Time-duration variations only change the number of times that the chain process visits certain states, but do not change the dynamic structure between these states. Hidden Markov models (HMMs) are the most commonly selected state-space methods, since they provide efficient dynamic time warping (DTW) between two temporal/time series, which is essential in the recognition of series-based patterns [122]. Motivated by the great success achieved in speech recognition [91], computer vision researchers have seen increasing interest in using HMMs to recognize hand and body gestures. Yamato et al. [168] first used HMMs to recognize tennis swings with appearance-change features. Although the recognition rate was relatively low due to the extensive long features and a large number of potential states that the author selected, their work encouraged researchers to implement HMMs in action recognition. Campbell et al. [26] applied Hidden Markov models to recognize 18 Tái Chi gestures. Starner et al. [144] combined HMMs with a strong grammar to recognize American Sign Language on the sentence level. Breglar [25] used a multiple level framework, where the recognition of human gaits, such as walking, running and skipping, was decomposed into three parts — blob extraction, tracking over time and HMM modeling. Krahnstöver et al. [87] established a unified framework for

tracking and analysis of human motions based on HMMs and a low-dimensional description of the spatial information obtained by PCA.

HMMs are also extended in different aspects in order to deal with particular problems in HAR. Wilson and Bobick [162] proposed a parametric hidden Markov model (PHMM), in which a distinctive parameter is added to the observations. The added parameter was used to reserve meaningful variations in gestures. Brand et al. [24] proposed a two state-layer HMM, named Coupled Hidden Markov Models (CHMMs), to model interactions between two hands. Unlike using traditional single state-layer HMM, in CHMM, each hand had its own state process. The state processes of two hands were connected at the state level. This two-state layer structure is more consistent to the inherent properties of hand interactions and thus provided a more accurate model of the dynamics of hand gestures. Instead of using first-order HMMs, Galata et al. [52] suggested that variable-length of Markov models (VLMM) is more suitable for activity recognition and applied VLMM to model human behaviors. In the exemplar-based method proposed by Elgammal et al. [42], the distribution of exemplars in each state of a HMM is estimated by a non-parametric density estimation method rather than the traditional parametric ones.

Beside HMMs, other types of state-space methods have also been used for HAR; one of these is the finite-state automaton [63, 154]. The advantage of these methods over HMMs is that their computational cost is much smaller than that of HMMs. In respect to the modeling ability, HMMs are still the best choice in most of HAR

applications.

In the following sections, two specific application domains of HAR are introduced — gait recognition and sign language recognition.

2.3 Gait Recognition

Gait recognition is a specific area of individual human activity recognition. Automated person identification systems are increasingly required by applications such as visual surveillance and access control. Like DNA, face, ear, iris, palm print and fingerprint, gait can also be used for person identification. Compared with other biometrics, gait is more suitable to be used when the identification is required at a distance, and which does not require close contact or high quality images. Since one's gaits may change with age, joints' injury and body weight, gait is not supposed to be super distinctive. But for human identification in low-security applications, gait is sufficiently discriminatory.

Gaits are characterized by two factors — period of the movement and the shape of the body, which give rise to two issues in gait recognition, period analysis and spatio-temporal analysis.

2.3.1 Gait Period Detection

Compared with other application domains of individual human activity recognition, a particular problem to be solved in gait recognition is the estimation of gait periods. Since a gait sequence often contains several gait cycles, the gait period can be

obtained from frequency analysis of the input features.

Little and Boyd [21, 95] proposed a non-structural motion-based method to recognize gaits. Optical flow images were first derived from gait-image sequences. A few scalars were computed from the flow images for representation and their fundamental frequencies and phases for each scalar were computed. The set of scalars shares the same fundamental frequency but their phases are different. Relative differences in phases are used for recognition. In [14, 33], self-similarity plots are used for gait period detection which is formed by the absolute correlation between a gait image and each image in the gait sequence. For a given image, it has a minimum correlation with a similar image and a large correlation with different images. Therefore, the frequency and phase of human gaits can be determined by finding the local minima of the self-similarity plots. Lee and Grimson [92] divide the binary body image into seven parts, and moment features of each part are extracted. Fourier frequency transformation is then used to obtain the dominant frequency for each part based on these moment features. A spectral component feature is then used to represent the gait sequence. Foster et al. [49] use masking functions to measure area as a time-varying signal from a sequence of silhouettes. Each masking function computes the number of the foreground pixels inside a specific mask and generate an area curve from a gait sequence. Minima in the area curve are used to detect the period of gaits. For a single gait period, a cubic spline curve is fitted, and thirty evenly spaced samples are selected from the fitted curve as the gait signature. Zhao et al. [177] applies fast Fourier transform (FFT) to the zero and first order mo-

ments of silhouettes to derive the amplitude spectrum from a given gait sequence. Based on the amplitude spectrum and the height of centroid, the period of the gait is computed and four ‘key poses’ are extracted for a single period. The average of the amplitudes of key poses are then used as the gait signature. Sarkar et al. [137] introduce the HumanID Gait Challenge Problem, which provide a large data set and a baseline algorithm to compare the performances of gait recognition methods. The baseline algorithm uses the number of foreground pixels in the bottom half of the silhouette to represent a gait image. Since this number reaches a maximum when the two legs are far away from each other, and reaches a minimum when the two legs overlap, the period of the gait can be easily detected. A test sequence is then divided into several probe sequences, and the median value of the maximum correlation of the reference sequence with each of these probe sequences is chosen as the similarity measurement.

2.3.2 Spatio-temporal Analysis

For spatio-temporal analysis, silhouette is the most popular feature in gait recognition. There are several advantages of using silhouettes to represent gaits. Firstly, silhouettes do not change with the color of clothes. Secondly, silhouettes are easy to extract by using background subtraction techniques. Thirdly, silhouettes are suitable for night vision since they can be obtained from IR imaging devices. Significant efforts have been devoted to the recognition of human gaits using silhouettes.

In the earliest work, Niyogi and Adelson [111] extracted gait signatures from

spatio-temporal (XYT) patterns. The bounding contours of the body were recovered from XYT patterns using ‘snakes’ and fitted to a five stick model. The fitted model was then normalized by velocity. Linear interpolation was used to generate gait vectors of fixed height and fixed temporal duration. Nearest neighbors with weighted Euclidean distance measures on these normalized vectors were used for recognition. HMMs are used by Kale et al. [82] where the binary silhouette and its width of the outer contour are used as features. A frame-to-exemplar distance (FED) is proposed to reduce the high dimension feature vectors to N-dimensional vectors, where N is the number of exemplars. Hayfron-Acquah et al. [60] extract symmetry maps from the edge of each silhouette, and the signature of gait is obtained by averaging the symmetry maps of all gait images. Kale et al. [81] proposed a method to estimate the angle between the person’s walking direction and the image plane assuming that the walking person is far enough from the camera. The 3D human body is then viewed as a planar object, and the side-view of body shape can be synthesized from any arbitrary views, which makes it possible to perform view-invariant gait recognition.

The eigenspace transform is a popular technique for feature extraction in silhouette-based gait recognition. Murase and Sakai [108] extracted silhouettes from gait sequences using background subtraction. Each silhouette was viewed as a high-dimensional vector. The covariance matrix of gait images of all persons in the data set was computed and a corresponding eigenspace was generated. Feature vectors of both the test and reference sequences were projected to the eigenspace to

generate compact vectors. Linear interpolation was used to generate a variety of reference sequences from a single reference sample. The minimum distance from the test feature sequence to these references was chosen as the similarity measurement. Huang et al. [69] combined a spatial template and a temporal template for gait recognition. The spacial template was the silhouette of the body and the temporal template was based on the magnitude of flow vectors within the silhouette window. Eigenspace transform and canonical transform were combined to extract features from the two templates. Wang et al. [157] extract gait features from the contour of a silhouette, where the contour is represented as a sequence of the distance from each contour point to the contour centroid. The distance sequence is then projected to an eigenspace to reduce the dimension of feature vectors. Dynamic time warping (DTW) is then used to compute the similarity between two gait sequences. Another version of this contour-based silhouette analysis is [156]. Instead of using Euclidean distance, the Procrustes distance is used to compute the mean and the covariance of a set of contours. The eigenvector that corresponds to the maximum eigenvalue is set as the gait signature.

Beside silhouettes, optical flow has also been used for spatio-temporal analysis. For example, Little and Boyd [95, 21] used frequency analysis of flow images for recognition. Huang [68] extracted features from the magnitude of flow vectors. The computational cost of optical flow is usually much larger than that of silhouette extraction, since silhouette can be extracted by using background subtraction. Therefore, optical flow is less popular than silhouette in the spatio-temporal analysis

of gaits.

2.3.3 Model-based Gait Recognition

The extractions of both silhouette or optical flow are sensitive to noises. The changes of lighting conditions and camera motions can make a clean extraction of silhouettes or optical flow nearly impossible. In order to reduce the interference of noise, prior knowledge of human bodies can be used to enhance the accuracy of human body detection.

Bobick and Johnson [17] labeled three body parts including head, pelvis and foot by analyzing silhouettes. Head and pelvis were detected by their centroids. Feet were detected by finding those pixels within the foot regions that have a maximum distance to the head. After recovering the human body, the height of the body, the height of the torso, lengths of the two legs, and the distance between the two feet were used as features. Electro-magnetic motion markers were used by Tanawongsuwan and Bobick [146] to extract joint positions. The trajectories of lower body joint angles were used to represent body shapes in a gait sequence. The trajectories were normalized in time using DTW for distance computation. Yam et al. [166, 167] first used running pattern for identification as well as walking behavior. The thigh and one of the lower legs are extracted by temporal template matching. The gait signature is derived as the phase-weighted magnitudes of the lower order Fourier components of the thigh and lower leg rotation. Wang et al. [155] use a human body model that contains fourteen parts including lower torso, neck, two upper

arms, two lower arms, two thighs, two lower legs, two feet, and a head. The head is represented by a sphere and the remaining parts are represented by truncated cones. The Condensation framework is used to track each part of the body, and the joint angles of the two hips and knees are selected as features for gait recognition.

2.4 Sign Language Recognition

A particular application domain of interest in HAR is sign language recognition (SLR). Sign language is the language of choice for most deaf people all over the world and many countries have their own sign languages which are different from each other. It is difficult for deaf people to communicate with normal people who have not been trained to understand sign languages or with other deaf people who use other sign languages. The technology of SLR can build a bridge between deaf people and others by translating observed sign signals to speech or other type of sign signals.

Researchers of computer vision have explored techniques for automatic recognition of various types of sign languages. A fuzzy min-max neural network was adopted by Kim et al. [84] to recognize the Korean sign language (KSL). Starner et al. [144] used HMMs to recognize the American sign language (ASL). Liang and Ming [116] also used HMMs for the recognition of the Taiwan sign language (TWL). Sagawa and Takuechi [136] describes words of the Japanese sign language (JSL) with primitives that were composed of hand shape, palm direction and hand linear motion. Bauer et al. used HMMs to recognize the German sign language (GSL)

[12]. Fang and Gao [44] combine a simple recurrent network and HMMs to recognize the Chinese sign language (CSL).

Representation and modeling methods for SLR are very similar to those for gesture recognition. The particular issue that requires special attention is the recognition of the continuous sign language with a large vocabulary. Two problems need to be solved for this task; they are sign segmentation and efficient search of candidate words.

Sign segmentation can be realized by analyzing time-varying parameters (TVPs) such as hand shape, hand velocity and hand positions. The changes in these parameter can be used to detect gesture borders. Liang and Ming [116] detected the gesture borders in a signed sentence by identifying the point that the change in hand shape is less than a selected threshold. Sagawa and Takuechi [136] used the hand velocity and movement direction to detect borders of signs. When the hand velocity was minimal or when the change of hand direction was larger than a threshold, a candidate hand border was detected. TVP-based border detection is easy to implement, but it is vulnerable to false segmentations since TVPs are sensitive to noise. Pruning of the detection results is usually necessary for sign segmentation. Different from the TVP-based methods, a simple recurrent network (SRN) is trained using labeled samples by Fang and Gao [44] to automatically segment sign words from continuous signals.

To reduce the search space of possible candidate sign words during recognition, sentence analysis is always used in SLR. Various approaches have been explored

for this task. Finite state automaton is a commonly method for sentence analysis. Kim et al. [85] divide hand gestures into five phases, each of which is represented as a state. Fuzzy logic is used to partition a continuous gestures into five states. A finite state automaton is used to model the transition between these states or segmented sentences. Language models can also been used to recover the sentences of continuous sign signals. Starner et al. [144] proposed a continuous ASL recognition using a single camera. Features were extracted from the hand blobs detected using color, including hand position, change of hand position, area and others. HMMs were used as the classifier of signs. A strong grammar was added to HMMs, which limited the possible connections between signs and reduced the search space for continuous recognition. Ma et al. [97] used the probability of word-transitions to indirectly encoded pragmatics by considering the local dependencies between two consecutive sign words. This probability was estimated by counting the frequency that two certain sign words appears one after another. Given a recognized sign word, the next adjacent word was only searched within those words who had a high connecting probability with the given word. A similar word-transition probability is obtained automatically from unlabeled sequences in [53]. A fuzzy decision tree is used by Fang et al. [45] to reduce the search space of a vocabulary consisting of 5113 signs.

In real-life applications, features of sign languages can be obtained using non-computer-vision technologies which is robust against environmental variations, such as using data-gloves that can easily detect the position and motion of hands and

even fingers. With these technologies, it is possible that deaf people will be able to communicate conveniently with normal people who do not know the sign language at all.

2.5 Hidden Markov Model (HMM)

In this section, a brief introduction of HMMs is presented. This will form the basis of our proposed algorithms for HAR.

HMM is the dominant model accepted in speech-recognition community [91, 122]. In recent years, HMMs are increasingly popular in image processing and computer vision. Typical applications of HMMs include handwriting recognition [89, 127], gesture recognition [32, 144, 162], human action recognition [62, 124, 168, 174], finger print recognition [57, 138], and scene detection [5, 40]. People show great interest in applying HMMs to applications in DNA segmentation and gene prediction [88, 143, 170]. Another promising application of HMM is economic-data analysis [148]. The proposed algorithms in this work are based on HMM; hence it is necessary to introduce some basic concepts of HMM before the details of this work are discussed.

The basic concept of HMM is the Markov property, which means that ‘for any sequence of time domain events, the conditional probability density of a current event given all the past and present events depends only on the most recent j events’ (quoted from [70]). A process that satisfies the Markov property is called a Markov process. If ‘ j ’ equals to 1, the process is called a first-order Markov process, which means that the current state only depends on the nearest predecessor and the

transition from one state to another is independent of time.

HMM use a Markov process (normally only the first-order Markov process is considered) to model the changing statistical characteristics of the observed sequences. The word ‘hidden’ in HMM means that the underlining Markov process can never be observed directly — only the outputs of each state can be observed, and the state itself can not be observed. A 4-states HMM with ergodic state-transition structure is shown in Figure 2.4. HMM has a two layers, one for observations and the other for hidden states. In the observation layer, observations are independent to each other, and each observation is connected to a hidden state with an output probability. In the state layer, states are connected with each other by state-transition probabilities, denoted by the arrow-headed arcs in Figure 2.4.

Figure 2.4: A 4-state HMM with an ergodic state-transition matrix; white circles represent states and black circles represent observations.

Fundamentally, an HMM can be described by the formulation of $\lambda = \{S, C, A, B, \Pi\}$,

where

- $S = \{s_1, s_2, \dots, s_N\}$ is the set of states, and N denotes the number of states.
- $C = \{c_1, c_2, \dots, c_K\}$ is the observation alphabet, and K denotes the number of elements in this alphabet. C is also called a codebook.
- $A = \{a_{ij} \mid i, j = 1, 2, \dots, N\}$ is the set of transition probabilities, where a_{ij} denotes the transition probability from state i to state j .
- $B = \{b_j(k) \mid j = 1, 2, \dots, N \text{ \& } k = 1, 2, \dots, K\}$ is the set of output probabilities, where $b_j(k)$ denotes the probability of observing c_k under state s_j .
- $\Pi = \{\pi_1, \pi_2, \dots, \pi_N\}$ is the set of initial state probabilities, where π_i denotes the probability that the initial state of the process is s_i .

The constraints of these parameters are given by

$$\sum_{i=1}^N \pi_i = 1, \quad (2.1)$$

$$\sum_{j=1}^N a_{ij} = 1, \text{ for each } i \quad (2.2)$$

$$\sum_{k=1}^K b_i(k) = 1, \text{ for each } i. \quad (2.3)$$

Both the number of states, N , and the observation alphabet, C , are predefined by users. Only A , B and Π need to be estimated from the training data. Therefore, an HMM is often written in the short form of $\lambda = \{A, B, \Pi\}$.

When implementing HMM in pattern recognition applications, three main issues need to be addressed; they are likelihood computation, HMM training and HMM decoding. Likelihood computation is to compute the likelihood of an observation sequence O for a given HMM λ ; this is mostly done using the Baum-Welch algorithm. HMM training concerns the estimation of parameters of HMMs from a set of training samples. This process is very important, since the parameters of HMMs must be obtained from a training database in most applications, rather than defined manually. The traditional training method of HMMs is based on the expectation maximization (EM) algorithm. HMM decoding is to recover the hidden states from the original observation sequence and the given HMM. This process is required for sequence segmentation since states are more reliable to be used in finding the beginning and end points of a pattern in a continuous input signal. The Viterbi algorithm is the most popular decoding algorithm for HMMs. The detail of these issues are introduced in the following sections.

2.5.1 Likelihood Computation

Let $O = o_1 o_2 \dots o_T$ be an input sequence, where T is the length of time steps, and λ be an HMM. The probability of observing O under λ is called the likelihood of O for λ and denoted by $P(O|\lambda)$. In the discussion of likelihood computation, it is assumed that the parameters of λ has already been obtained.

For a fixed state sequence $S = q_1 q_2 \dots q_T$, where q_t is the state at time step t ,

we have

$$P(O, S|\lambda) = P(O|S, \lambda)P(S|\lambda). \quad (2.4)$$

The most direct way to compute the likelihood of $P(O|\lambda)$ is to sum Eq. (2.4) over all valid permutations of S :

$$P(O|\lambda) = \sum_S P(O, S|\lambda) = \sum_S P(O|S, \lambda)P(S|\lambda). \quad (2.5)$$

A direct computation of this summation leads to an intractable computational complexity. Baum–Welch’s forward-backward algorithm [122] is the most widely used method for reducing the computational complexity in this summation process. This algorithm introduced two variables, the forward variable and the backward variable, for likelihood computation.

Forward Variable. $\alpha_t(i) = P[o_1 o_2 \dots o_t, q_t = s_i | \lambda]$, where

$$\alpha_1(i) = \pi_i b_i(o_1), \quad (2.6)$$

$$\alpha_t(i) = b_i(o_t) \sum_{j=1}^N \alpha_{t-1}(j) a_{ij}, \quad (2.7)$$

$$\begin{aligned} P(O|\lambda) &= \sum_{i=1}^N P(o_1 o_2 \dots o_T, q_t = s_i | \lambda), \\ &= \sum_i \alpha_T(i). \end{aligned} \quad (2.8)$$

Backward Variable. $\beta_t(i) = P[o_{t+1}o_{t+2}, \dots o_T, q_t = s_i | \lambda]$, where

$$\beta_T(i) = 1, \quad (2.9)$$

$$\beta_t(i) = \sum_{j=1}^N \beta_{t+1}(j) a_{ij} b_j(o_{t+1}), \quad (2.10)$$

$$\begin{aligned} P(O|\lambda) &= \sum_{i=1}^N P(o_1 o_2 \dots o_T, q_t = s_i | \lambda), \\ &= \sum_i \pi_i b_i(o_1) \beta_1(i). \end{aligned} \quad (2.11)$$

As shown in Eq. (2.8) and (2.11), $P(O|\lambda)$ can be computed using either the forward variable or the backward variable independently. Combining the two variables, a solution to the problem of parameter estimation of HMMs can also be formulated as described in the next section.

2.5.2 HMM Training

The training of HMMs is a process to refine the parameters of HMMs iteratively to fit the training samples. The classical re-estimation algorithm is derived from the expectation maximization (EM) approach. The updating formulas are obtained by maximizing an auxiliary function Q ,

$$Q(\lambda, \bar{\lambda}) = \frac{1}{P(O|\lambda)} \sum_{all S} P(O, S|\lambda) \log P(O, S|\bar{\lambda}), \quad (2.12)$$

where $\bar{\lambda}$ represents the updated model and λ refers to the original model. The term ‘log’ in Eq. (2.12) refers to the logarithm that are often used in information theory.

In the experiments of this work, the natural basis is used.

Two additional variables are defined based on the *forward* and *backward* variables,

$$\begin{aligned}\gamma_t(i) &= P[q_t = s_i | O, \lambda] \\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)}.\end{aligned}\tag{2.13}$$

$$\begin{aligned}\gamma_t(i, j) &= P[q_t = s_i, q_{t+1} = s_j | O, \lambda] \\ &= \frac{\alpha_t(i)\alpha_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)\alpha_{ij}b_j(o_{t+1})\beta_{t+1}(j)} \\ &= \frac{\alpha_t(i)\alpha_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{P(O|\lambda)}.\end{aligned}\tag{2.14}$$

The estimation process of HMM parameters can be formalized using the following equations:

$$\bar{\pi}_i = \gamma_1(i),\tag{2.15}$$

$$\bar{\alpha}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i, j)}{\sum_{j=1}^N \sum_{t=1}^{T-1} \gamma_t(i, j)},\tag{2.16}$$

$$\bar{b}_j(k) = \frac{\sum_{t \in \{o_t=k\}} \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)}, \quad (2.17)$$

where $t \in \{o_t = k\}$ means only the time steps that k is observed are included.

For detail derivation of above equations, readers are referred to the book of Huang [70].

2.5.3 Viterbi Algorithm

Viterbi algorithm is a dynamic programming procedure which is used to derive the most likely state sequence $Q^* = q_1^* q_2^* \dots q_T^*$ for a given observation sequence based on a HMM λ . At each time step t , the value δ is computed by

$$\delta_t(i) = \max_{q_1 q_2 \dots q_{t-1}} P(q_1 q_2 \dots q_{t-1}, q_t = s_i, o_1 o_2 \dots o_t | \lambda), \quad (2.18)$$

for every state s_i in the model. By induction,

$$\delta_{t+1}(i) = b_i(o_{t+1}) \max_{j=1,2,\dots,N} (\delta_t(j) a_{ji}). \quad (2.19)$$

The maximum likelihood is obtained by

$$P(O|\lambda) = \max_{i=1,2,\dots,N} \delta_T(i). \quad (2.20)$$

In order to retrieve the sequence Q^* leading to the maximum likelihood, the best state path generated by this maximization is stored in an array ϕ_t , with

$$\phi_t(i) = \arg \max_j \{\delta_{t-1}(j)a_{ji}\}, \text{ for } i, j = 1, 2, \dots, N. \quad (2.21)$$

The best state sequence is retrieved by

$$q_t^* = \phi_{t+1}(q_{t+1}^*), \quad t = T - 1, \dots, 1. \quad (2.22)$$

2.5.4 Using HMMs for HAR – An Example

In order to illustrate the usage of HMMs in HAR, this section presents the details of using HMMs to classify fourteen classes of human activities. The same dataset is also used in Chapter 3 and Chapter 4, and the HMMs described in this section found the basis of our higher level algorithms including automatic activity segmentation and non-meaningful activity rejection.

The activities are captured using a Sony DV camera. The original resolution of each frame is 720×576 . The frame rate of capturing is 25 fps. In order to save the processing time, the original video is converted to 10 fps with a frame resolution of 360×288 . All activities are captured from a lateral view, as shown in Figure 2.5 and Figure 2.6.

The first step of using HMMs to recognize these activities is to extract activity information from the original video. Background subtraction is applied to extract

human bodies. In our experiments, it is assumed that the background is static (no motion for background pixels) and a Gaussian background model is established for each pixel using a number of clean background frames. After this step, the original activity video is converted to a sequence of binary shapes. Figure 2.7 shows detected shape masks of the key frames of each activity.

The second step is to extract a feature from the binary shape in each frame. This feature should have the following properties: Firstly, it must reduce the dimensionality of the derived feature. This is to avoid the ‘dimension curse’ problem. Secondly, it should not be sensitive to the size (number of pixels inside a shape) of shapes. This is because activities can be captured with different scales of a body shape. In this work, Hu-moments [19] are used to extract features from binary shapes.

The feature sequences of each class of activities may vary from each other due to two factors, the dynamics of each activity and the postures of the person who performs the activity. Variations in the dynamics of an activity is caused by the performing speed of the activity. For the same activity, some people perform it faster and some people perform it slower. Even for one person, he/she can perform the same activity with different speed. Some parts of the activity may be performed faster in some sequences while slower in other sequences. Variations in postures is mainly caused by the silhouettes of different people and the noise from background subtraction. Figure 2.8 shows the different postures of three persons performing SitDown. People may also vary their postures arbitrarily when performing the same activity. Figure 2.9 shows the posture variation when a person performs BendDown

with different body configurations. These variations can be handled by HMMs. The variations in dynamics are modeled by state-transition parameters in HMMs, while the variations in postures are modeled by the distribution of the features in each state.

In the third step, a number of training sequences are selected for each activity class. The HMMs used for in this work have a left-right state transition structure, which is illustrated in Figure 2.10. The number of states N is set to 5. This state transition structure has less parameters to estimate, since only a limited number of transitions between states are permitted. At the beginning of HMM training, the initial parameters of an HMM include Π , A and B . Since the left-right state transition structure is used, Π is set to

$$\begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

This results in that the state sequence always starts from state 1. This initial Π is used in all activity classes.

The state transition matrix A is set to

$$\begin{bmatrix} 0.7 & 0.3 & 0 & 0 & 0 \\ 0 & 0.7 & 0.3 & 0 & 0 \\ 0 & 0 & 0.7 & 0.3 & 0 \\ 0 & 0 & 0 & 0.7 & 0.3 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

In fact the initial value of A is not critical for HMM training. These values have more effects on the converging speed than on the final result. Since a state in a left-right HMM is more likely to transit to itself, the self-transition probability of each state is set to be higher (0.7) than cross-transition probabilities (0.3). The zeros inside A denotes where the transitions between certain states are forbidden. All activity classes use the same initial A .

The initial value of B is class dependent. In this experiment, a Gaussian distribution is used for each state, which is given

$$b_i(o) = \frac{1}{\sqrt{2\pi}\Sigma_i} \exp \frac{-(o - \mu_i)^2}{2\Sigma_i^2}, \quad (2.23)$$

where μ_i and Σ_i are the mean and standard variation of the Gaussian in state i .

μ_i and Σ_i are initialized by

$$\mu_i^0 = \frac{1}{\sum_{m=1}^M T_m} \sum_{m=1}^M \sum_{j=1}^{T_m} o_j, \quad (2.24)$$

$$\Sigma_i^0 = \frac{1}{\sum_{m=1}^M T_m - 1} \sum_{m=1}^M \sum_{j=1}^{T_m} (o_j - \mu_i)^2, \quad (2.25)$$

where M is the number of training sequences, and T_m is the number of frames in the m th sequence. Note that according to Eq. (2.24) and (2.25), all states in an HMM have the same initial distribution parameters.

Using these initial values, HMMs are trained for all activity classes. Table 2.2 and Table 2.3 show their converged state-transition matrix A , and the center of each state's distribution, respectively. The recognition ability of the trained HMMs are also tested. Figure 2.11 shows the recognition rate versus the length of inputs. It can be seen that the recognition rate is close to 100% when 85% percentage of the frames of each sequence are used as input. This experiment shows that the established HMMs are capable of activity recognition if the tested patterns are long enough. These HMMs are used for the experiments of our proposed activity segmentation and non-meaningful activity rejection algorithms, which is described in the following chapters.

Table 2.2: The estimated state-transition matrix A of the activity classes (1) SitDown, (2) StandUp, (3) Point, (4) LieDown, (5) LieUp, (6) SquatDown, (7) SquatUp, (8) BendDown, (9) BendUp, (10) Punch, (11) Kick, (12) JumpForward, (13) JumpUp and (14) JumpFrog.

Index	A	Index	A
1	$\begin{bmatrix} 0.7964 & 0.2036 & 0 & 0 & 0 \\ 0 & 0.4357 & 0.5643 & 0 & 0 \\ 0 & 0 & 0.6036 & 0.3964 & 0 \\ 0 & 0 & 0 & 0.7763 & 0.2237 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$	2	$\begin{bmatrix} 0.7444 & 0.2556 & 0 & 0 & 0 \\ 0 & 0.7528 & 0.2472 & 0 & 0 \\ 0 & 0 & 0.7226 & 0.2774 & 0 \\ 0 & 0 & 0 & 0.7364 & 0.2636 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$
3	$\begin{bmatrix} 0.7645 & 0.2355 & 0 & 0 & 0 \\ 0 & 0.5162 & 0.4838 & 0 & 0 \\ 0 & 0 & 0.4599 & 0.5401 & 0 \\ 0 & 0 & 0 & 0.6880 & 0.3120 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$	4	$\begin{bmatrix} 0.7814 & 0.2186 & 0 & 0 & 0 \\ 0 & 0.6984 & 0.3016 & 0 & 0 \\ 0 & 0 & 0.3586 & 0.6414 & 0 \\ 0 & 0 & 0 & 0.6902 & 0.3098 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$
5	$\begin{bmatrix} 0.7939 & 0.2061 & 0 & 0 & 0 \\ 0 & 0.4955 & 0.5045 & 0 & 0 \\ 0 & 0 & 0.5000 & 0.5000 & 0 \\ 0 & 0 & 0 & 0.7098 & 0.2902 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$	6	$\begin{bmatrix} 0.8796 & 0.1204 & 0 & 0 & 0 \\ 0 & 0.8146 & 0.1854 & 0 & 0 \\ 0 & 0 & 0.7428 & 0.2572 & 0 \\ 0 & 0 & 0 & 0.6927 & 0.3073 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$
7	$\begin{bmatrix} 0.6767 & 0.3233 & 0 & 0 & 0 \\ 0 & 0.4756 & 0.5244 & 0 & 0 \\ 0 & 0 & 0.0000 & 1.0000 & 0 \\ 0 & 0 & 0 & 0.4782 & 0.5218 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$	8	$\begin{bmatrix} 0.6224 & 0.3776 & 0 & 0 & 0 \\ 0 & 0.4410 & 0.5590 & 0 & 0 \\ 0 & 0 & 0.3601 & 0.6399 & 0 \\ 0 & 0 & 0 & 0.4524 & 0.5476 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$
9	$\begin{bmatrix} 0.3976 & 0.6024 & 0 & 0 & 0 \\ 0 & 0.7555 & 0.2445 & 0 & 0 \\ 0 & 0 & 0.6776 & 0.3224 & 0 \\ 0 & 0 & 0 & 0.7027 & 0.2973 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$	10	$\begin{bmatrix} 0.7655 & 0.2345 & 0 & 0 & 0 \\ 0 & 0.4574 & 0.5426 & 0 & 0 \\ 0 & 0 & 0.2074 & 0.7926 & 0 \\ 0 & 0 & 0 & 0.5003 & 0.4997 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$
11	$\begin{bmatrix} 0.8182 & 0.1818 & 0 & 0 & 0 \\ 0 & 0.6946 & 0.3054 & 0 & 0 \\ 0 & 0 & 0.5699 & 0.4301 & 0 \\ 0 & 0 & 0 & 0.5652 & 0.4348 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$	12	$\begin{bmatrix} 0.7564 & 0.2436 & 0 & 0 & 0 \\ 0 & 0.2831 & 0.7169 & 0 & 0 \\ 0 & 0 & 0.0000 & 1.0000 & 0 \\ 0 & 0 & 0 & 0.8841 & 0.1159 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$
13	$\begin{bmatrix} 0.7916 & 0.2084 & 0 & 0 & 0 \\ 0 & 0.0217 & 0.9783 & 0 & 0 \\ 0 & 0 & 0.5585 & 0.4415 & 0 \\ 0 & 0 & 0 & 0.2934 & 0.7066 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$	14	$\begin{bmatrix} 0.6825 & 0.3175 & 0 & 0 & 0 \\ 0 & 0.4119 & 0.5881 & 0 & 0 \\ 0 & 0 & 0.4137 & 0.5863 & 0 \\ 0 & 0 & 0 & 0.7499 & 0.2501 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$

Figure 2.5: Examples of tested activities - part I.

Figure 2.6: Examples of tested activities - part II.

Figure 2.7: Body shape sequences of activities used in experiments.

Figure 2.8: Three different person perform SitDown activity, their postures are different.

(a) The arms' position differs in two BendDown activities.

(b) The postures are different in two Kick activities.

Figure 2.9: Examples of posture variations in activities that are performed by the same person.

Figure 2.10: State-transition structure of the left-right HMMs used in experiments.

Figure 2.11: The recognition rate of the trained HMMs versus the length of the sequences. The horizontal axis represents the percentage of frames that are used as input.

2.5 Hidden Markov Model (HMM)

62

Table 2.3: The estimated output probabilities of the activity classes (1) Sit-Down, (2) StandUp, (3) Point, (4) LieDown, (5) LieUp, (6) SquatDown, (7) SquatUp, (8) BendDown, (9) BendUp, (10) Punch, (11) Kick, (12) Jump-Forward, (13) JumpUp and (14) JumpFrog.

	s_1	s_2	s_3	s_4	s_5
1	6.0406e-001	5.7176e-001	4.7604e-001	3.3940e-001	3.9805e-001
	3.3155e-001	2.8128e-001	1.5916e-001	4.0488e-002	9.9182e-002
	4.1860e-003	2.1587e-002	3.8358e-002	2.8597e-002	1.6772e-002
	1.3344e-003	4.7528e-003	6.7944e-003	4.7432e-003	1.8540e-003
	1.7748e-006	2.8272e-005	3.0430e-005	1.4201e-005	-2.8752e-006
	3.8472e-004	5.4665e-004	-1.1200e-004	2.1343e-005	-2.5220e-004
2	4.3429e-001	3.8437e-001	3.5722e-001	4.5173e-001	5.9167e-001
	1.2950e-001	8.2420e-002	6.4664e-002	1.5543e-001	3.1977e-001
	1.3083e-002	1.9903e-002	2.2174e-002	1.9794e-002	2.4468e-003
	1.8134e-003	1.7159e-003	2.1819e-003	2.2781e-003	9.8320e-004
	6.6471e-006	-3.7231e-006	-7.1414e-006	-7.8731e-006	1.8357e-006
	-1.8189e-005	-2.8397e-004	-3.1004e-004	-4.3929e-004	3.4995e-004
3	4.7025e-001	3.3697e-001	3.5535e-001	4.6985e-001	5.0287e-001
	1.4993e-001	4.2489e-002	4.2891e-002	1.3669e-001	1.8382e-001
	1.5330e-002	1.5683e-002	1.5168e-002	2.9466e-002	3.6356e-002
	1.7722e-003	2.3170e-003	2.6923e-003	8.1623e-003	1.2488e-002
	-2.9290e-006	2.7604e-006	-1.1470e-005	6.1785e-005	2.4916e-004
	-3.2126e-004	4.2571e-006	-3.6359e-004	1.2804e-003	4.8491e-003
4	6.2712e-001	5.8645e-001	4.9617e-001	3.0140e-001	2.2815e-001
	3.6021e-001	3.0724e-001	1.8961e-001	3.7042e-002	1.3543e-002
	2.1586e-003	1.0229e-002	3.3231e-002	1.8259e-002	4.6707e-003
	1.5183e-003	2.5372e-003	5.8960e-003	2.6509e-003	7.1706e-004
	4.0165e-006	8.8066e-006	2.8012e-005	9.1877e-006	1.0979e-006
	7.4256e-004	4.5989e-004	-5.2969e-005	4.5649e-005	4.5857e-005
5	2.3387e-001	2.2063e-001	2.3220e-001	3.3649e-001	5.8445e-001
	4.8853e-003	1.1683e-002	1.9956e-002	6.8525e-002	3.0947e-001
	8.9214e-003	3.5569e-003	2.0768e-003	7.5209e-003	4.0387e-003
	6.0385e-004	5.1101e-004	6.1019e-004	9.6533e-004	8.6281e-004
	1.4271e-006	6.4949e-007	8.2165e-007	-1.3010e-007	9.4841e-008
	3.1211e-005	5.2869e-005	8.2664e-005	-8.0782e-005	1.2379e-004
6	6.1628e-001	6.0830e-001	4.7013e-001	3.2596e-001	2.3511e-001
	3.4975e-001	3.2733e-001	1.7600e-001	6.4610e-002	1.7275e-002
	1.4150e-003	1.1236e-002	1.3007e-002	5.2383e-003	3.0545e-003
	7.8326e-004	5.0060e-003	2.1887e-003	8.3898e-004	5.8769e-004
	1.0973e-006	6.1928e-005	1.2465e-006	6.8444e-007	6.0650e-007
	3.8250e-004	2.3475e-003	-8.1221e-005	2.7883e-005	3.6568e-005
7	2.4517e-001	3.8950e-001	5.2495e-001	6.1929e-001	6.1743e-001
	2.2512e-002	1.0916e-001	2.2937e-001	3.4807e-001	3.5042e-001
	3.4404e-003	7.1591e-003	1.4282e-002	7.8330e-003	1.7760e-003
	7.7856e-004	2.1819e-003	7.7609e-003	4.1332e-003	7.8738e-004
	1.0746e-006	8.4988e-006	1.2076e-004	4.4087e-005	1.3821e-006
	6.6450e-005	5.4697e-004	3.6355e-003	2.2691e-003	3.5763e-004
8	7.4847e-001	7.0809e-001	6.3310e-001	5.1921e-001	4.0746e-001
	5.1877e-001	4.1993e-001	2.6689e-001	1.2042e-001	2.9823e-002
	8.2506e-003	6.2113e-002	1.0338e-001	8.2476e-002	4.2253e-002
	1.6765e-003	7.4117e-003	1.2829e-002	1.1677e-002	7.8054e-003
	1.9673e-006	-1.5645e-004	-4.2260e-004	-2.3693e-004	-3.3506e-005
	4.4535e-005	-4.3357e-003	-6.1216e-003	-2.8780e-003	-1.2858e-004
9	4.2554e-001	5.3197e-001	6.4160e-001	7.1395e-001	7.3882e-001
	4.2180e-002	1.3608e-001	2.9921e-001	4.4707e-001	5.0771e-001
	4.6837e-002	8.3121e-002	8.4833e-002	3.8022e-002	3.3978e-003
	8.3861e-003	1.2243e-002	1.0675e-002	4.7861e-003	5.9859e-004
	-3.0171e-005	-1.6517e-004	-2.3967e-004	-4.2293e-005	-5.0351e-007
	-9.5287e-005	-2.3846e-003	-4.6829e-003	-2.3665e-003	-5.1227e-005
10	4.6698e-001	5.0464e-001	5.6006e-001	5.3788e-001	5.2803e-001
	1.6570e-001	1.9114e-001	2.4527e-001	2.1182e-001	2.0634e-001
	7.2141e-003	2.0275e-002	1.9543e-002	3.1491e-003	9.6355e-003
	1.5173e-003	5.6755e-003	5.8718e-003	1.1182e-003	3.2086e-003
	5.3760e-006	7.5466e-005	9.0914e-005	5.1061e-007	1.2604e-005
	5.3141e-004	2.5027e-003	2.8978e-003	1.8210e-005	7.2426e-004
11	4.9301e-001	4.1331e-001	4.6751e-001	4.6129e-001	4.7178e-001
	1.8937e-001	5.4957e-002	6.7402e-002	3.5325e-002	1.6427e-001
	1.4750e-002	4.2780e-002	9.2669e-002	7.7699e-002	9.6255e-003
	3.7533e-003	6.5639e-003	1.5104e-002	6.6985e-003	1.1917e-003
	3.7482e-005	1.0753e-004	2.7361e-004	-5.6521e-006	4.1549e-006
	1.3646e-003	1.1777e-003	2.5496e-003	9.2320e-004	2.8755e-004
12	4.4523e-001	3.3509e-001	5.6644e-001	3.3177e-001	3.1675e-001
	1.0511e-001	4.6695e-002	2.8322e-001	4.4504e-002	6.3964e-002
	2.9635e-002	1.1620e-002	9.0949e-003	1.7149e-002	3.3417e-003
	4.2381e-003	1.2162e-003	2.8958e-003	2.0372e-003	6.5695e-004
	4.9875e-005	2.7608e-006	1.4785e-005	-8.0198e-006	7.8808e-007
	9.7338e-004	1.1081e-004	8.8807e-004	-1.5425e-004	3.7907e-005
13	3.3881e-001	3.9456e-001	4.8835e-001	4.8673e-001	3.6315e-001
	8.0642e-002	4.9974e-002	1.2173e-001	1.4151e-001	7.7129e-002
	2.5567e-003	2.9067e-002	5.6349e-002	3.5801e-002	8.3605e-003
	1.9551e-003	4.8881e-003	5.0054e-003	2.2544e-003	2.9015e-003
	6.8226e-006	7.0855e-005	8.3690e-005	2.1700e-005	1.7371e-005
	6.1352e-004	1.1438e-003	1.6538e-003	7.5893e-004	7.8804e-004
14	2.8356e-001	5.1129e-001	4.6648e-001	3.7691e-001	2.5430e-001
	3.1235e-002	2.1024e-001	1.5144e-001	3.8772e-002	1.3985e-002
	9.2176e-003	1.5709e-002	2.2995e-002	4.7586e-002	9.7160e-003
	1.1149e-003	4.7284e-003	2.4245e-003	5.5593e-003	1.1349e-003
	8.2697e-007	4.0730e-005	-1.2678e-005	-4.0532e-005	3.3092e-007
	1.9632e-005	1.3067e-003	-7.0903e-004	-6.2176e-004	3.6098e-005

Chapter 3

Automatic Segmentation of Human Actions

The automatic segmentation of activities plays a very important role in HAR systems. In real life applications, activities are needed to be recognized from continuous video streams. Therefore, activities should be segmented before recognition. Note that 'activity segmentation' refers to the segmentation of an activity sequence from a long image sequence; it is not the segmentation of human body from an image. In most of the previous efforts of HAR technology, activities are often manually segmented, where the beginning and end points of each activity are, more or less, accurately determined. Although manual segmentation of activities are acceptable in comparing the performance of various classification algorithms, automatic segmentation is necessary to make HAR systems fully automatic.

The difficulty in activity segmentation lies in two aspects. Firstly, the beginning

and end points of human actions are ambiguous. As a result, the transition between two consecutive actions is often difficult to determine. To locate the action boundaries, our human eyes rely more on the grammar and logic of the activities than on the observed features. The same task of activity segmentation is difficult to achieve on computers since fully transmitting our grammars and logics to a computer is still beyond the ability of current artificial intelligence (AI) technologies. Secondly, various human activities may have similar features for their beginning and end frames. For example, activities such as ‘sitting down’ and ‘squatting down’ begin with the same standing poses.

In computer vision technologies, most of the video segmentation algorithms, such as shot detection, are based on the discontinuity analysis of some global features such as color histogram, texture and motion between two consecutive frames. These algorithms are helpful to divide a long video sequence into segments, extract key frames from each segment, and cluster segments into groups according to the similarity between their key frames. However, these algorithms do not make use of the narrative meaning of each segment and hence cannot assure that each activity of interest would be correctly segmented into isolated segments. Moreover, the transition between two consecutive actions are often so ambiguous that shot-detection techniques usually cannot find the action boundaries.

In this chapter, a new segmentation method is presented. Activity models (based on HMMs) are used in the segmentation process in order to ensure that the segmented video clips are consistent with our human understandings about the ac-

tivities. A background state is added to each activity HMM to detect irrelevant frames in the video sequence. The associated parameters of the background state, such as its output density function and transition probability, are not obtained by the traditional training process but designed manually. The difference between the proposed method and other existing methods is that the proposed method can be constructed by using only relevant training samples whereas other methods need to establish alternative models using a large amount of irrelevant samples.

In the sections that follow, Section 3.1 presents a brief review of related works. The proposed background state method is proposed in Section 3.2. Section 3.3 discusses the decoding process using the HMM with a background state. The segmentation results are validated in Section 3.4, while Section 3.5 further supports the proposed approach with experimental results from continuous activity segmentation using an individual activity database.

3.1 Previous Work

The task of automatically segmenting patterns of interest from continuous input signals was first investigated in speech recognition systems. Normally, the input signal of a continuous speech recognition system only contains background sounds and words to be recognized. If the input signals also contain speech signals of unknown words, the segmentation task is also called word spotting, which means that only designated keywords are the targets to be detected. The problem of action segmentation is very similar to that of word spotting, where non-meaningful actions

largely exist.

In continuous speech recognition, the segmentation process often includes the search for beginning and end points of each keyword. Since the word boundaries are difficult to be detected directly, all possible beginning and end points have to be searched. Such an exhaustive search is not feasible since its computational cost increases exponentially with the number of classes. Several sub-optimal search algorithms, such as the forward-backward search [128, 160] and the Viterbi decoding [100, 130], were used instead. The forward-backward search algorithm has two steps. Firstly, the end point of a word is detected by searching, in the forward direction, for local maxima of the posterior probability $P(s_t = e_n | x_1 x_2 \dots x_t)$, where s_t is the state label for time t , e_n is the state index of the last state of word n , and $x = x_1 x_2 \dots x_t$ is the observed sequence. Secondly, a backward search is executed to find the beginning point of the word. Alternative models that represent background sounds or irrelevant words are required for the forward-backward search to reduce the number of false boundary detection. In a different manner, the Viterbi decoding algorithm does not require such a two-pass search. It translates an speech sequences into a state sequence by using an hyper-HMM in which meaningful HMMs (HMMs representing meaningful sounds) and alternative HMMs (HMMs representing background sounds) are combined. Figure 3.1 shows the structure of a hyper-HMM used for the Viterbi decoding. In the hyper-HMMs, meaningful HMMs provide the states for relevant speech signal and alternative HMMs provide the states for irrelevant sounds. The segments of the meaningful words can be easily extracted from the decoding results.

Figure 3.1: The HMM for Viterbi decoding in which alternative HMMs are embedded. $\{\lambda_i | i = 1, 2, \dots, N\}$ is the set of meaningful HMMs, $\{\bar{\lambda}_j | j = 1, 2, \dots, M\}$ is the set of alternative HMMs. S_0 and E_0 are pseudo beginning and end states respectively.

It is difficult to apply speech segmentation technologies to human activity segmentation directly. Since both the forward-backward search and the Viterbi decoding require alternative models, the training set of a speech recognition system should contain a large amount of irrelevant sound signals for the establishment of alternative HMMs. In order to ensure that the alternative models represent any irrelevant sound better than the meaningful HMMs, the included irrelevant samples in the training set has to be carefully selected to cover the irrelevant space as large as

possible. In HAR systems, the establishment of alternative models is very difficult since the space of irrelevant activities is too huge to be covered. Moreover, certain non-meaningful activities may be partially similar to some meaningful ones, which makes the training of the alternative models more difficult. In HAR systems, the segmentation methods that are less dependent on irrelevant samples are much more desirable.

Segmenting human actions from continuous video can also be viewed as a particular task of video segmentation. Existing algorithms for video segmentation can be roughly divided into two categories. The first category mainly address the problem of dividing a video into 'shots' according to the scene properties. A shot is a group of homogeneous frames that share similar scene properties, such as pixel intensity, motion, color histogram or texture, of the entire frame [72], or certain regions within the frame [178, 179], or a mixture of the previous two [4]. The second category focuses on the problem of segmenting videos into different types such as news, sports or forecast [20, 40, 109]. Algorithms in this category extract features from each frame and establish models to represent different video contents. The features extracted by algorithms in this category are similar to that in the previous one. More recently, there is also a trend in incorporating multiple cues, such as audio and text, in video segmentation [7, 141]. These cues may not be suitable for segmenting the entire video sequence independently since they may not be available throughout the whole video sequence. Nevertheless, there will be scenes that are difficult to be detected simply by analyzing motion, color histogram and texture, and for such scenes the

use of alternative cues come in handy. For example, highlights in a tennis game are similar to normal plays in terms of scene properties. It is difficult to distinguish a good forehand-stroke from a bad one. But the sound coming from the audience and the change in score text may help to distinguish good and bad plays easily.

These video segmentation techniques are not applicable for the task of activity segmentation in HAR systems. This is because consecutive human actions normally do not cause a significant change in scene properties; two activities may smoothly transit from one to another. Compared with shot and scene detections, activity segmentation requires more details about the contents of a video sequence. Moreover, cues like audio and texts are very domain specific. For general activity segmentation, these cues should not be relied on.

Several other methods are proposed recently to divide a video sequence into segments according to human actions, each of which has their own advantages and weakness. Rui and Anandan [134] proposed an algorithm to detect action boundaries in a video sequence containing arbitrary actions. Their algorithm is based on temporal discontinuities of the spatial pattern describing image motions. For a sequence of length $N + 1$, N optical-flow images of a human body (between each two consecutive frames) are computed. Each optical flow image is rearranged using the raster-scan order so that the sequence of optical-flow images are represented as a sequence of $1D$ flow vectors. SVD is then applied to the vector sequence which project the flow vectors to the SVD bases. The principle SVD bases are selected and the discontinuities in the coefficients of these bases are used to detect action boundaries.

Naphade and Huang [109] used a network of embedded HMMs to segment a video sequence into clusters. The segmentation is based on the expectation-maximum (EM) training process. Similar video sequences were assigned to the same branch of hidden states in the network. Compared with traditional shot-detection algorithms, algorithms in [109, 134] can obtain smaller segments according to the changes in activities within a shot. However, since no activity model is used in segmentation, the detected activities may not be consistent with our human understanding about activities, in which a typical activity of interest may be divided into several sub-activities. Zelnik-Manor and Irani [173] proposed a measurement for the similarity between a given sequence with an reference-action sequence. Each sequence is represented by a temporal pyramid in which the sequence is divided by various temporal scales. At each temporal scale, the spacial gradients, S_x and S_y , and temporal gradient, S_t , were computed for each pixel in the frame. An experimental distribution is associated with each of these gradients. The similarity between two sequences is computed by the summations of the distances between their experimental distributions at all scales. Since this algorithm is also based on the global property of images, it is sensitive to background motions. Lee et al. [90] proposed a hand-gesture segmentation algorithm based on their non-gesture rejection model, named threshold HMM (THMM). Their segmentation method is similar to the forward-backward search algorithm used in speech recognition. However, the performance of THMM depends on the number of training samples for each class in activity recognition. To assure the performance of THMM, the number of training samples must

be much larger than that for the traditional HMM training.

3.2 The Background State of HMM

In this work, we are interested in activity segmentation methods which do not require any irrelevant samples to establish alternative models that represent irrelevant observations. Although irrelevant samples largely exist and are easy to obtain, it is difficult to use a finite set of non-meaningful samples to build alternative models that can represent an infinite space of irrelevant observations. Instead of building alternative models, a segmentation method requiring only meaningful samples is proposed in this work.

Let $Y = y_1 y_2 \dots y_N$ be a video sequence of length N and λ_i be the HMM of i th action, the segmentation task is to find the set of boundary points $\{p_s^j, p_e^j | j = 1, 2, \dots, Z\}$, where Z is the number of actions inside Y , and p_s^j, p_e^j are the beginning and end points of the j th action. The difficulty of this task lies in two aspects. Firstly, Y contains multiple actions that should be separated before matching Y with λ_i . Secondly, irrelevant observations (those observations that do not belong to λ_i) exist between two meaningful actions, which deteriorate the likelihood computation of Y for λ_i .

In speech recognition, an efficient way to solve the segmentation problem is to use the Viterbi algorithm, which assigns null-state labels to irrelevant observations and meaningful state labels to each action segment in Y . However, as discussed in Section 3.1, the Viterbi decoding methods require additional models to provide

Figure 3.2: The structure of a HMM after adding the background state S_0 .

null-state labels for irrelevant observations. Although such additional models have been successfully established in speech recognition systems, they are difficult to be trained in HAR systems, since there are infinite possibility of human activities, while only a tiny portion of these activities are of interest. In this section, a new method for Viterbi decoding is proposed which does not require alternative models.

Instead of using alternative models, a *background state* is directly added to each action HMM. The background state functions similarly to the null-states in the additional models in the sense that they provide labels to irrelevant observations. The difference is that the background state is unique, while a null-states is always embedded in an additional HMM, and each additional HMM can have multiple null-states. Due to the variety of human activities, the transition probabilities between the background state and other states are difficult to be trained with a finite set of activity samples. Instead, the state-transition parameters of the background state are determined manually in this work. Figure 3.2 shows the structure of a HMM having K states after adding the background state to it.

Let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1K} \\ a_{21} & a_{22} & \dots & a_{2K} \\ \vdots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nK} \end{bmatrix}$$

be the state transition matrix of HMM λ_i (see Section 2.5 for details of HMM). The background state is added to λ_i as the $(K + 1)$ th state and denoted by the number of 0. The new state-transition matrix $\overline{\mathbf{A}}$ is

$$\overline{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1K} & 0 \\ a_{21} & a_{22} & \dots & a_{2K} & 0 \\ \vdots & \vdots & \dots & \vdots & 0 \\ a_{K1} & a_{K2} & \dots & a_{KK} & a_{K0} \\ a_{01} & 0 & \dots & 0 & a_{00} \end{bmatrix},$$

where a_{K0} , a_{01} and a_{00} represent the transition probabilities of $P(s_{t+1} = 0 | s_t = K)$, $P(s_{t+1} = 1 | s_t = 0)$ and $P(s_{t+1} = 0 | s_t = 0)$, which must be non-zero. Note that except for a_{K0} , a_{01} and a_{00} , other transition probabilities between the background state and activity states are set to 0. This form of $\overline{\mathbf{A}}$ indicates that the decoded state process can only visit the background state after visiting the last state S_K ; when the state process goes into the background state, it can only visit either the background state or the first state S_0 .

The distribution density function of observations in the background state needs

to be decided for the Viterbi decoding. Obviously, this density function is difficult to estimate experimentally. Instead of training a distribution using a selected set of irrelevant observations, the sparse distribution of all meaningful observations is used as the density function for the background state. This distribution should have the characteristic that if an observation is relevant to the tested activity, its likelihood for the background state is less than that for at least one meaningful state; if the observation is irrelevant to the tested activity, its likelihood for the background state should be higher than that for any meaningful state.

In this work, Gaussian distributions are used for meaningful states (states from trained activity HMMs, see Section 2.5.4). In order to assure that the distribution of the background state does not provide a higher likelihood for a relevant observation that meaningful states do, the background state is also described by a Gaussian density function. The background Gaussian is derived from the observations from all meaningful activity classes, hence it is sparser than the distribution of any meaningful state. The usage of the background state can be viewed as using a single state HMM, in which only self state-transition is possible, as an additional model. The newly constructed HMM is denoted by $\hat{\lambda}_i$.

3.3 The Decoding Process

Given a video sequence $Y = y_1 y_2 \dots y_N$ and an activity HMM λ_i , the goal of the decoding process is to find out boundary points of all activities that belong to λ_i . A sliding window W_n is used to limit the length of the segmentation to be

decoded, where n represents the size of the window. The sliding window is forward shifted by one frame at each step, which means Y is divided into $N - n$ windowed sequences for decoding. In order to assure that each target activity locates at least in one windowed sequence, n is set to be larger than the the maximum length of i -type activities. Part of the windowed sequence (sometimes the whole sequence) is constructed by irrelevant observations. $\hat{\lambda}_i$ is then constructed for the Viterbi decoding where the irrelevant observations are labeled with the background state.

During the decoding process, the number of times, that the state process visits the end state of an activity, needs to be constrained. This is because there are occasions when some irrelevant observations have low likelihoods for both the end state and the background state. When these observations are observed after the end point of an activity, they may be labeled as the end states instead of the background states. In order to correctly label these irrelevant observations, the duration of the end state should be considered. Empirically, this duration is set to be a little longer than the maximum duration observed in training samples.

Figure 3.3 shows the decoding result using a sequence containing a JumpForward activity and irrelevant frames using the HMM with a background state for JumpForward. The state index ‘0’ is used to label irrelevant observations. Obviously, the beginning and end points, denoted by 1 and 5 respectively, can be easily determined from the decoding result.

As presented in Chapter 1, the segmentation module is independent from the modules of irrelevant pattern rejection and classification. However, in real systems,

Figure 3.3: Viterbi decoding result of a sequence by using the JumpForward HMM with a background state; only a few key frames are selected (evenly extract from the original sequence) in order to reduce the number of shown frames.

these modules are intertwined with each other and are difficult to separate. In order to reduce the number of false segmentation results, a validation process is further added, which combines the modules of irrelevant pattern observation and classification to the segmentation process. The validation process is presented in the next section.

3.4 Validating the Segmentation Results

After decoding the N windowed sequences with the Viterbi algorithm, multiple pairs of beginning and end points can be found. Since the background states are not highly representative of irrelevant observations, some irrelevant segments may be falsely detected by the proposed decoding process. In order to find the true locations of the activity boundaries, further processing of the found boundary-point pairs is need.

The decoding results are validated by the proposed pairwise likelihood ratio (PLR) test. A detailed discussion of the PLR test is presented in Chapter 4. Here,

only the definition of the PLR test is given.

Let $l_i(\mathbf{x}) = \{l_{i,j}(\mathbf{x}) \mid l_{i,j}(\mathbf{x}) = \log \frac{P(\mathbf{x}|\lambda_j)}{P(\mathbf{x}|\lambda_i)}, j = 1, 2, \dots, M \text{ \& } j \neq i\}$ be the set of pairwise log-likelihood ratios generated from $\mathbf{x}$, a pattern from λ_i . $l_i(\mathbf{x})$ can be represented as a vector by taking each set member as an element in the vector. This is called a pairwise log-likelihood ratio vector and is expressed as

$$l_i(\mathbf{x}) = [l_{i,1}(\mathbf{x}), l_{i,2}(\mathbf{x}), \dots, l_{i,i-1}(\mathbf{x}), l_{i,i+1}(\mathbf{x}), \dots, l_{i,M}(\mathbf{x})]^T.$$

For each class, the multi-variate distribution of their pairwise likelihood ratio vectors is estimated to reject non-meaningful patterns. More conveniently, the Mahalanobis distance from a PLR vector to the center of a class's distribution is used,

$$\left[(l_i(\mathbf{x}) - \bar{l}_i(\mathbf{x}))^T \Sigma_i^{-1} (l_i(\mathbf{x}) - \bar{l}_i(\mathbf{x})) \right] \begin{cases} > \epsilon_i \text{ reject } \mathbf{x} \\ \leq \epsilon_i \text{ accept } \mathbf{x}. \end{cases} \quad (3.1)$$

where $\bar{l}_i(\mathbf{x})$ and Σ_i are the mean and the covariance matrix of $l_i(\mathbf{x})$, which can be estimated using the training data. This is equivalent to using PLRs from multiple support classes to establish a PLR space and use the distribution of the PLR vectors to decide if a detected segment should be accepted. In order to enhance the accuracy of rejection, the PLR vectors and the distributions can be projected to a subspace using the principle component analysis (PCA) [80]. The details of this operation is described in Section 4.2.2.

The PLR test can reject most of the falsely detected segments. However, there

Figure 3.4: Part of the detection result of an activity sequence containing two ‘punch activities’.

are occasions when some falsely detected segments may survive the PLR test. A few steps are necessary to further process the segmentation results. Firstly, the length of the detected activity, represented by the difference between the beginning and end positions, should be within a proper range. If the length of the detected activity is too short, the corresponding pair of boundary points is not true. This often happens when a few irrelevant frames are similar to the beginning part of an activity. Secondly, since the sliding window moves forward one frame at each step, the boundary points of an activity may be detected for multiple times. This causes variations in the position of the detected boundary points. Figure 3.4 shows part of the decoding result of a sequence containing two consecutive Punch activities, which are shown in Figure 2.7. Clearly, the detected boundaries can be clustered into two

groups denoted by A and B . The positions of the beginning point in Group A and that of the end point in Group B , varies slightly within each group. The median values of the positions of the detected beginning points and end points are chosen in this work as the true position.

3.5 Experiments of Continuous Activity Segmentation

In this section, HMMs, which are built in Section 2.5.4 (see Table 2.2 and Table 2.3), are used to detect the fourteen classes of human activities as shown in Figure 2.5 and Figure 2.6. These activity classes construct the basis of the set of meaningful activities.

The background state is added to each of the meaningful HMMs class. The parameters of the transition matrix of the new HMM containing a background state is set by

$$\overline{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 & 0 & 0 \\ 0 & 0 & a_{33} & a_{34} & 0 & 0 \\ 0 & 0 & 0 & a_{44} & a_{45} & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix},$$

where a_{ij} represent the original state-transition probabilities. The bottom row rep-

resents the transition probabilities from the background state to other states: The transition probability from the background to the initial state is 0.5, and the self-transition probability of the background state is also 0.5. The right most column shows the transition probabilities from other states to the background state: Only the end state (the 5th state in this work) can transit to the background state, with a probability of 0.5.

Twenty-eight video sequences containing meaningful activities and random non-meaningful activities have been collected. The length of each video ranges from 300 to 1000 frames. Figure 3.5 shows a portion of the test sequence containing two activities. The total number of detectors is fourteen. These detectors are applied to each video sequence independently to detect its corresponding activities. Two type of errors are measured in this experiment. One is the false-detection error denoting that a non-activity segment is detected. The other is the missed-detection error denoting that a true segment is not detected by its correspondent detector.

The length of the sliding window used for Viterbi decoding is set to be twice the maximum sequence length observed during training for each activity detector. In the rejection process, a set of distance threshold is tested for each class,

$$\{\epsilon_i(j) \mid \epsilon_i(j) = D_i^{max} * j, j = 1, 2, \dots, 50\}.$$

where D_i^{max} is the maximum Mahalanobis distance among the PLR vectors of training samples. For each class, the best $\epsilon_i(j)$, which gives a high detection rate and a low false alarm rate, is selected and its detection result is show in Table 3.1.

Table 3.1: Activity segmentation results. N_a is the number of activities in each video, N_f is number of the falsely detected segments and N_c is the number of the correctly detected segments.

Type	Total Number (N_a)	False Detection (N_f)	Correct Detection (N_c)	Missed Detection	Detection Rate (P_d) (%)	Reliability (R) (%)	$\epsilon_i = D_i^{max} * j$
SitDown	34	3	30	4	88.24	90.9	2.4
StandUp	30	0	22	8	73.33	100	1.2
PointTo	20	0	20	0	100	100	1.9
LieDown	25	4	20	5	80	83.33	2.9
LieUp	25	0	21	4	84	100	1.9
SquatDown	25	0	24	1	96	100	3.1
SquatUp	25	0	22	3	88	100	2.9
BendDown	20	0	20	0	100	100	1.6
BendUp	20	0	20	0	100	100	1.4
Punch	42	0	39	3	92.86	100	2.9
Kick	25	0	24	1	96	100	2.6
JumpForward	40	12	35	5	87.5	74.47	1.4
JumpUp	42	0	39	3	92.86	100	2.0
JumpFrog	39	8	28	11	71.79	77.78	2.9
Total	412	27	364	48	88.38	93.09	

The detection rate P_d is computed by

$$P_d = \frac{N_c(i)}{N_a(i)}, \quad (3.2)$$

where N_c is the number of correctly detected activities and N_a is the number of true activities to be detected. A correct detection result is determined when the detected boundary points are within a proper range H of the true positions of the boundary points. If the position of a detected boundary point locates in previous H or next H frames of the true position, it will be determined to be a correct detection. The range H is set according to the length of activities to be detected. In our experiments, H is set to 3 to 5 depending on the length of each type of activity. The reliability of

each detector is measured by

$$R(i) = \frac{N_c(i)}{N_c(i) + N_f(i)}. \quad (3.3)$$

It can be seen that most of the detectors are capable of providing a high detection rate. The low detection rates in 'StandUp', 'JumpForward' and 'JumpFrog' are because that they have similar features and dynamics to each other. In these activity detectors, the PLR test has to be set tighter in order to suppress false-positive alarms. When the activities are highly distinguishable from others, such as 'BendDown', 'BendUp', 'SquatDown' and 'SquatUp', both the detection rates and the reliability are very high.

In order to measure the overall performance of the proposed activity spotting framework, receiver operating Characteristic (ROC) curves are obtained by altering the rejection threshold to generate various number of falsely detected activities. These curves are estimated as the fraction of true detection for a given number of falsely detected activities. The same set of $\epsilon_i(j)$ for each class is used, and the error range is fixed for every class. At each j , a point $(\bar{P}_d(j), F(j))$ is obtained for the average recognition ROC curve, and a point $(\bar{R}(j), F(j))$ is obtained for the average accuracy ROC curve, where $F(j)$ is the total number of false detected activities

across all classes, and

$$\begin{aligned}\bar{P}_d(j) &= \frac{\sum_{i=1}^M N_c(i)}{\sum_{i=1}^M N_a(i)}, \\ \bar{R}(j) &= \frac{\sum_{i=1}^M N_c(i)}{\sum_{i=1}^{N_c(i)+N_f(i)} N_c(i)}.\end{aligned}$$

Figure 3.6 show the ROC curves for average detection rate and average accuracy under different error range H . It can be seen that the proposed framework can provide both acceptable recognition rate and accuracy with a small number of falsely detected activities.

3.6 Summary

This chapter presents a segmentation framework based on the Viterbi decoding using the background state, which is established with only relevant observations. The background state, which is added to activity HMMs, is used to label irrelevant observations in the Viterbi decoding. Compared with previous video segmentation methods, the Viterbi decoding is used in the proposed framework, which decode activities using the activity models. This result in that the detected boundaries are consistent with our human understanding about activities. Compared with other segmentation methods using the Viterbi decoding, the proposed method does not require any non-meaningful patterns to build alternative models, which simplifies the training process. The experimental results have shown that the proposed segmentation method performs well with individual action segmentation.

(a)

(b)

(c)

Figure 3.5: Parts of the used sequences. The black circles represent the beginning point of activities, while the white circles represent the end points

(a)

(b)

Figure 3.6: ROC curves of the averaged true-detection rate (a), and the averaged accuracy versus the number of falsely detected activities (b).

Chapter 4

Rejection of Non-Meaningful Patterns Based on HMMs

In Chapter 3, an auto-segmentation method based on the background state is presented. In the validation process of the segmentation results, the pairwise likelihood ratio test (PLRT) is used. This chapter presents the details of PLRT for non-meaningful pattern rejection. The word ‘meaningful’ and ‘non-meaningful’ in this work does not refer to the generic definition of what is ‘meaningful’ or ‘non-meaningful’ in our daily life. Instead, ‘meaningful’ activities refers to a set of predefined activities which are of interest, while ‘non-meaningful’ activities refers to any other activities that do not belong to the predefined set.

The techniques for non-meaningful pattern rejection are motivated by the task of *human-activity spotting* since a major difficulty in activity spotting stems from the existence of non-meaningful activity patterns, which can be infinite and un-

predictable. The rejection of such non-meaningful patterns is a highly challenging task due to two aspects. Firstly, images captured from a typical human activity are 2D projections of a series of 3D body postures. This characteristic makes it difficult to extract highly discriminative features from human activities considering variations in both posture and projection. Features adopted in existing human activity recognition methods, such as meshes [168], sticks [51], contours [42], flows [39] and moments [36], are view-dependent and sensitive to noise. Their discriminating powers depend on the application environments. As a result, non-meaningful classes may overlap with meaningful classes, which aggravates the difficulty in non-meaningful pattern rejection. Secondly, in order to deal with the overlaps among classes caused by insufficient features, the state-space approach such as HMMs is often used, in which a feature is viewed as an observable variable and associated with an unobservable variable, the hidden state. Confusion among overlapped classes is alleviated by modeling their temporal properties in the state space. However, the state-space approach use the maximum likelihood (ML) criteria and is subjected to ‘forced classification’ where a pattern is classified into one of the meaningful categories even though it does not belong to any of them. The maximum likelihood score is not an absolute measurement of the hit hypothesis and is meaningful only when it is compared to those of other hypotheses produced for the same pattern. It is inevitable that non-meaningful patterns will be given meaningful labels when the pure ML rule is adopted.

The main focus of this chapter is to distinguish meaningful patterns from non-

meaningful ones. A new rejection algorithm is proposed based on an extension of the traditional likelihood ratio test (LRT). Instead of using the Neyman-Person criteria, the statistical distribution of the pairwise likelihood ratios are used as a measurement to reduce the false alarm rates.

The rest of this chapter is organized as follows: Section 4.1 reviews the existing algorithms for non-meaningful pattern rejection. The proposed pairwise likelihood ratio (PLR) test is presented in Section 4.2. The performance of the proposed algorithm is compared with other algorithms in Section 4.3 and 4.4.

4.1 Review of Related Works

Several techniques are used to alleviate the forced-decision characteristic of the ML rule. Let $\mathbf{x}$ be an input pattern and $\{\lambda_i \mid i = 1, 2, \dots, M\}$ be the set of meaningful models, the recognition of $\mathbf{x}$ is normally decomposed into two steps. Firstly, the likelihood of $\mathbf{x}$ for each class is computed and a hypothesis class λ_i is generated according to the ML rule, where

$$\lambda_i = \arg \max_i [P(\mathbf{x}|\lambda_i), i = 1, \dots, M].$$

Secondly, the rejection of non-meaningful patterns is operated as a hypothesis test between

Hypothesis 1

$$\begin{cases} H_1 : x \text{ is from the hypothesized model } \lambda_i, \text{ for } i = 1, \dots, M \\ H_0 : x \text{ is from the non-meaningful model.} \end{cases}$$

If H_1 hits, x is accepted as a pattern from λ_i , otherwise x is rejected.

Likelihood ratio test (LRT), which is widely adopted in the research of speaker identification [61], word spotting [129], DNA sequencing [71] and protein matching [10], is the most popular method for the above hypothesis test and is defined as

$$\Lambda(x) = \frac{P(x|\lambda_i)}{P(x|\lambda_0)} \begin{cases} \geq \theta & \text{accept } H_i \\ < \theta & \text{accept } H_0, \end{cases} \quad (4.1)$$

where $\Lambda(x)$ is likelihood ratio, λ_0 is the model for the patterns to be rejected, which is also called an *alternative model*, $P(x|\lambda_i)$ and $P(x|\lambda_0)$ are the likelihoods of x in hypotheses H_i and H_0 , respectively and θ is the rejection threshold.

The logarithm function of $\Lambda(x)$ is often used, given by

$$L(x) = \log P(x|\lambda_i) - \log P(x|\lambda_0). \quad (4.2)$$

If the likelihood function $P(x|\lambda_0)$ is known exactly, LRT is optimal for this hypothesis test. However, it is nearly impossible to explicitly define the model λ_0 since it is required to represent the entire space of possible alternatives to λ_i . The key issue in LRT methods is the construction of the alternative model λ_0 .

Three main approaches have been taken to obtain λ_0 . The first approach is to build a set of alternative models using a large number of non-meaningful patterns. Such models are called *filler models* [129] or *garbage models* [161]. Discussion on various techniques for filler model construction can be found in the work of Rose and Paul [130]. Although this approach sounds ideal, it is difficult to realize in most of real applications due to the infinity of non-meaningful patterns.

The second approach is to approximate $\bar{\lambda}$ with a set of meaningful models. This approximation is given by

$$P(\mathbf{x}|\lambda_0) = \mathcal{F}(\{P(\mathbf{x}|\lambda_j)\} : j = 1, 2, \dots, M; j \neq i), \quad (4.3)$$

where $\mathcal{F}()$ is a function of the likelihood values from meaningful models other than λ_i . Typical $\mathcal{F}()$ functions includes the maximum function [61] and the average function [131] (which is also called ‘cohort’). However, λ_0 may be poorly approximated in such a way. It is possible that some non-meaningful patterns may have higher likelihood ratios as well.

The third approach is to train an alternative model using patterns from various meaningful classes. Such models are called background models. Instead of representing the entire non-meaningful space, the background models represents either the entire meaningful space or a specific space for a meaningful class. Rosenberg and Parthasarathy [131] compared the performance of different construction methods for background models. Reynolds et al. [125] used patterns from a set of meaningful classes to train a universal background model (UBM) which took the form of a

Gaussian mixture model (GMM) rather than an HMM. Siohan et al. [140] proposed a perturbed training algorithm to train a background model for each meaningful class specifically. The associated background model of a class λ_i is also an HMM, which is trained using the same training samples of λ_i . But the background model has a less number of hidden states than λ_i has thus its representative ability is weak. The background model is further weakened by reversing the dynamics of its hidden states.

Besides the LRT methods, several other rejection methods have also been proposed. Jepson et al. [77] proposed a test called reliable inference (RI) which is applied by Davis and Tyagi [36] to human gait recognition. The form of RI is given by

$$\frac{P(H_1|x)}{P(H_0|x)} \begin{cases} \geq \delta & \text{accept } H_1 \\ < \delta & \text{accept } H_0, \end{cases} \quad (4.4)$$

where $P(H_1|x)$ and $P(H_0|x)$ are posterior probability functions for H_1 and H_0 , respectively, and δ is the rejection threshold similar to θ in Eq. (4.1). According to Bayesian theory,

$$P(H_1|x)/P(H_0|x) = P(x|H_1)P(H_1)/P(x|H_0)P(H_0),$$

RI is different from LRT only in the aspect that it takes advantage of using the prior probability of each class. The construction methods of the alternative model developed for LRT can also be adopted in RI. Lee and Kim [90] proposed a dynamic threshold method to reject non-meaningful patterns, which is so called threshold

hidden Markov model (THMM). THMM represents the entire meaningful space by giving any meaningful pattern a likelihood value which is little smaller than that the pattern's own class gives. No hypothesis test is required in THMM and the ML rule can be directly applied. For a meaningful pattern, the highest likelihood is produced by its true model. For a non-meaningful pattern, the highest likelihood is that the threshold model produces. However, from our experiments, we observed that the threshold model may produce a larger likelihood value for meaningful patterns than the true model does if the meaningful classes are highly overlapped.

In this chapter, a new rejection algorithm is proposed based on pairwise likelihood ratios (PLRs). The troublesome hypothesis test described above is solved with a new test which no longer requires any alternative model. The details of PLR test are provided in next section.

4.2 Pairwise Likelihood Ratio Test

The burden of existing rejection methods is to establish an alternative model which represents either the entire meaningful space or the entire non-meaningful space. The alternative model needs to be carefully defined and estimated in order to improve the rejection accuracy. However, an ideal rejection algorithm should be simple for implementation and should not require any complex training for the alternative model. To meet these requirements, a new hypothesis test algorithm is proposed in this chapter, which is called pairwise likelihood ratio test (PLRT).

4.2.1 Definition of PLR

We are interested in utilizing the information from the meaningful classes for rejection. Suppose we have M classes, $\{\lambda_i \mid i = 1, \dots, M\}$, which are estimated from a finite set of training samples. Let $\mathbf{x}$ be a test pattern and λ_i be the winning class selected by ML,

$$\lambda_i = \arg \max_i [P(\mathbf{x}|\lambda_i) \mid i = 1, \dots, M].$$

The task of rejection is to validate if $\mathbf{x}$ is generated from λ_i . Although λ_0 for non-meaningful patterns is not available, any other model, $\{\lambda_j \mid j = 1, 2, \dots, M; j \neq i\}$, can be used as an alternative model. Suppose a class λ_j is chosen as the alternative model. The hypothesis test can be rephrased as

Hypothesis 2

$$\begin{cases} H_0 : \mathbf{x} \text{ is from Class } i \\ H_1 : \mathbf{x} \text{ is not from Class } i. \end{cases}$$

The pairwise likelihood ratio is derived from Class i and the selected class j ,

$$\mathcal{L}_{i,j}(\mathbf{x}) = \frac{P(\mathbf{x}|\lambda_j)}{P(\mathbf{x}|\lambda_i)}, \quad (4.5)$$

where λ_j is called a *support class*. In this paper, all models are HMMs. $P(\mathbf{x}|\lambda_i)$ and $P(\mathbf{x}|\lambda_j)$ are *duration normalized likelihoods* of HMM λ_i and λ_j , respectively. The

LRT test for this hypothesis test can be expressed as

$$\begin{cases} \mathcal{L}_{i,j}(\mathbf{x}) \geq \eta & \text{reject } H_0 \\ \mathcal{L}_{i,j}(\mathbf{x}) < \eta & \text{not reject } H_0. \end{cases} \quad (4.6)$$

Note the decision ‘not reject H_0 ’ does not mean H_0 is true, it only suggests that there is insufficient evidence against H_0 in favor of H_1 .

Since λ_j is not the real alternative model, Eq. (4.6) may not be a proper test. Figure 4.1 shows an example using three simulated classes, C_1 , C_2 and C_3 . Each class is represented by a 2D Gaussian model, which is shown in Figure 4.1 (a) and (b). In this example, C_1 is chosen as the winning class, while C_2 and C_3 are used as the support classes. Figure 4.2 (a) shows the mesh of log-likelihood ratios between C_1 and C_2 , $P(\mathbf{x}|C_2)/P(\mathbf{x}|C_1)$. The log-likelihood ratio decreases monotonically as $\mathbf{x}$ moves to the left side of the mean of C_1 . Figure 4.2 (b) shows the equipotential contours of $P(\mathbf{x}|C_2)/P(\mathbf{x}|C_1)$. If η is set to 0, any pattern generated from the left region of 0 equipotential curve will be accepted, even though the pattern’s likelihood for C_1 is very small.

Instead of using a fixed threshold, the distribution of the PLRs is used as the measurement in PLRT. This distribution is characterized by the mean value of the PLRs generated by the training samples of λ_i ,

$$\bar{\mathcal{L}}_{i,j}(\mathbf{x}) = \frac{1}{N_i} \sum \mathcal{L}_{i,j}(\mathbf{x}), \mathbf{x} \in \lambda_i, \quad (4.7)$$

(a)

(b)

Figure 4.1: Three simulated 2D Gaussian classes: 3D mesh of the probability densities of each class (a), and 2D view of the distribution of each class (b).

and their standard deviation,

$$\mathcal{C}_{i,j}(x) = \frac{1}{N_i - 1} \sum [\mathcal{L}_{i,j}(x) - \bar{\mathcal{L}}_{i,j}(x)]^2, x \in \text{Class } i, \quad (4.8)$$

(a)

(b)

Figure 4.2: The mesh of log-likelihood ratio between C_1 and C_2 , $P(x|C_2)/P(x|C_1)$ (a) and the equipotential curves of $P(x|C_2)/P(x|C_1)$ in the original 2D space (b).

where N_i is the number of training samples for λ_i .

Let $\mathcal{L}_{i,j}(x)$ be the PLR generated from a test pattern x , the Mahalanobis distance from $\mathcal{L}_{i,j}(x)$ to $\bar{\mathcal{L}}_{i,j}(x)$ is used as the measurement for Hypothesis Test 2. This is

expressed by

$$\frac{[\mathcal{L}_{i,j}(\mathbf{x}) - \bar{\mathcal{L}}_{i,j}(\mathbf{x})]^2}{\mathcal{C}_{i,j}(\mathbf{x})} \begin{cases} > \epsilon_i & \text{reject } H_0 \\ \leq \epsilon_i & \text{not reject } H_0. \end{cases} \quad (4.9)$$

The effect of using the distribution of PLRs as the measurement is equivalent to limiting the acceptance region by two PLR thresholds, as shown in Figure 4.3 (a), where ϵ is set to 3. In this way, the acceptance region is reduced and fits the training data better.

Equation (4.9) can be viewed as a weak test which decides if a recognition result should be accepted. However, This decision is only based on the information from one support class. It is reasonable that a better decision can be made if the information of all available support classes are utilized. In the next section, we propose a grouping method to combine multiple PLR tests so as to improve the rejection performance.

4.2.2 Grouping Multiple PLR Tests

Let

$$\mathcal{L}_i(\mathbf{x}) = [\mathcal{L}_{i,1}(\mathbf{x}), \dots, \mathcal{L}_{i,i-1}(\mathbf{x}), \mathcal{L}_{i,i+1}(\mathbf{x}), \dots, \mathcal{L}_{i,M}(\mathbf{x})]^T$$

be the PLR vector generated from $\mathbf{x}$, which is classified as from λ_i . During the training process, each training sample generates a PLR vector. Since the distribution measurement is used in PLRT, the combination of multiple PLRTs can be translated to estimate the multi-variate distribution of these PLR vectors and use this distribution to reject non-meaningful patterns. This is to establish a PLR space and

Figure 4.3: The acceptance region of C_1 provided by PLRT, as the double arrow shows (a) and the distribution of PLR vectors of C_1 , using C_2 and C_3 as the support classes, in the PLR space (b).

use the distribution of PLR vectors in this space to decide if x should be accepted.

Equation (4.9) can be rewritten as

$$\left[(\mathcal{L}_i(\mathbf{x}) - \bar{\mathcal{L}}_i(\mathbf{x}))^T \Sigma_i(\mathbf{x})^{-1} (\mathcal{L}_i(\mathbf{x}) - \bar{\mathcal{L}}_i(\mathbf{x})) \right] \begin{cases} > \epsilon_i & \text{reject } H_0 \\ \leq \epsilon_i & \text{not reject } H_0. \end{cases} \quad (4.10)$$

where $\mathcal{L}_i(\mathbf{x})$ is a $(M - 1)$ dimensional vector, $\bar{\mathcal{L}}_i(\mathbf{x})$ and $\Sigma_i(\mathbf{x})$ are the mean vector and the covariance matrix of PLR vectors for λ_i . Since Equation (4.10) decide if H_0 should be rejected in favor of all the meaningful classes (except λ_i itself), it is more trustworthy to accept H_0 if ‘not reject H_0 ’ is suggested.

Figure 4.3 (b) shows the PLR space generated by the simulated classes used previously, where C_1 is chosen as the winning class. It can be seen that the patterns from C_1 can be bounded by a 2D distribution in the PLR space. When the number of PLR test is large, this is equivalent to using a hyper-volume in the PLR space as the acceptance region.

PCA can be used to further enhance the performance of the PLR test. Let K ($K \leq M - 1$) be the selected number of components, $\{e_{ik}, k = 1, \dots, K\}$ be the set of eigenvalues and $\{v_{ik}, k = 1, \dots, K\}$ be the set of eigenvectors satisfying $\Sigma_i v_{ik} = e_{ik} v_{ik}$.

Equation (4.10) can be modified as

$$\left[V_i^T (\mathcal{L}_i(\mathbf{x}) - \bar{\mathcal{L}}_i(\mathbf{x})) \right]^T \left[V_i^T \Sigma_i(\mathbf{x}) V_i \right]^{-1} \left[V_i^T (\mathcal{L}_i(\mathbf{x}) - \bar{\mathcal{L}}_i(\mathbf{x})) \right] \begin{cases} > \epsilon_i & \text{reject } H_0 \\ \leq \epsilon_i & \text{not reject } H_0, \end{cases} \quad (4.11)$$

where $V_i = [v_{i1}, v_{i2}, \dots, v_{iK}]$.

If the PLR test suggests that H_0 should not be rejected for a given pattern $\mathbf{x}$, then the classification result of $\mathbf{x}$ is decided as a true recognition. The optimal threshold ϵ for the PLR test of each class can be obtained experimentally.

4.3 Experiments with Simulated Data

Simulated data is used in this section to show the limitation of existing rejection algorithms and the advantages of PLRT. Five simulated HMMs are generated, which are constructed as shown in Figure 4.4, where HMM_1 , HMM_2 and HMM_3 are defined as meaningful models, while HMM_4 and HMM_5 are defined as non-meaningful models. Each of these HMMs has two states and generates observations in terms of 2D vectors. The output probability function of each state is a 2D Gaussian function, whose center is shown in Figure 4.4, with a diagnostic covariance matrix given by

$$\sigma = \begin{bmatrix} 0.078 & 0 \\ 0 & 0.078 \end{bmatrix}.$$

All these HMMs use the same state-transition matrix A

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix},$$

Figure 4.4: The topology of five simulated HMMs; each circle represents a state and its entry $(\cdot, \cdot)$ represent the mean of its output probability function.

and the same prior probability Π

$$\Pi = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix},$$

where the value 0.5 means that every state has an equal chance for a sequence to begin with.

The HMMs of human activities are much complex than the simulated models. However, the simulated HMMs have clear definition of their states and the similarity between them is easy to determine. This is helpful to show the intrinsic characteristics of each rejection algorithm.

4.3.1 LRT-Based Algorithms vs. PLRT

The accuracy of the alternative model is essential to the performance of LRT. If the alternative model is poorly estimated, the acceptance region can be much larger than

the true one. In [61], the maximum function is used in Eq. (4.3) for alternative model estimation, which means to select the nearest meaningful class as the alternative model. It can be seen from Figure 4.2 that such an approximation can result in a high false acceptance rate.

In face, even if the average function is used for Eq. (4.3), such as the cohort-based LRT [131], the approximated model may not be a good alternative model either. This is shown in the following experiments based on the simulated data.

Figure 4.5: Log-likelihoods of sequences from HMM_3 , HMM_4 and HMM_5 ((a), (b) and (c)) and their log-likelihood ratios computed by cohort-based LRT using an average function ((d)).

For every sequence from non-meaningful classes HMM_4 and HMM_5 , the winning class is always HMM_3 — this is obvious since HMM_4 and HMM_5 are closer to HMM_3 than to HMM_1 and HMM_2 . Figure 4.5 (a), (b) and (c) show the log-likelihoods of 20 sequences from HMM_3 , HMM_4 and HMM_5 for HMM_3 , respectively. Figure 4.5 (d) shows the likelihood ratios where the alternative model λ_0 is estimated by an average function

$$P(x|\lambda_0) \approx \frac{1}{2}[P(x|HMM_1) + P(x|HMM_2)]. \quad (4.12)$$

It can be seen from Figure 4.5 (d) that the sequences from HMM_5 can be successfully rejected since their log-likelihood ratios are lower than that of true HMM_3 sequences. However, the likelihood ratios for HMM_4 samples are much higher than that of HMM_3 , hence the sequences from HMM_4 will be accepted as meaningful patterns generated by HMM_3 .

Comparing Figure 4.5 (a) and (b), it can be seen that the log-likelihoods of the sequences from HMM_4 in HMM_3 are much smaller than those of the true HMM_3 sequences. The reason that HMM_4 sequences have a higher log-likelihood ratios in HMM_3 is that their log-likelihoods in HMM_1 and HMM_2 are even smaller, which increases the ratios. While, compared with the true HMM_3 sequences, the log-likelihoods of HMM_5 sequences in HMM_3 decrease and those log-likelihoods in HMM_1 and HMM_2 increase, hence the ratio decrease.

Suppose all states from the meaningful HMMs construct a sub state-space as shown in Figure 4.6, if the non-meaningful patterns are generated from a place

outside the sub state-space, the cohort-based LRT algorithm cannot reject them.

Figure 4.6: The sub state-space constructed by LRT-Cohort and the source of non-meaningful classes.

The advantage of PLRT over LRT-based algorithms is that PLRT make use of the distribution of PLRs. Even if HMM_4 patterns have higher PLRs in HMM_3 , their PLRs do not located in the distribution formed by true HMM_3 patterns, hence they can be rejected easily.

4.3.2 THMM vs. PLRT

Although THMM is not LRT-based, it has its own limitations. Figure 4.7 shows the log-likelihoods of the HMM_4 and HMM_5 sequences in HMM_3 and their log-likelihood thresholds in a THMM that is built using HMM_1 and HMM_2 . It can be seen that the thresholds provided by the THMM for these non-meaningful sequences are not higher than their likelihoods in HMM_3 , hence these non-meaningful sequences cannot be rejected. This limitation of THMM can be explained by its large

Figure 4.7: Log-likelihoods and THMM thresholds for (a) the sequences from HMM_4 and (b) the sequences from HMM_5 .

state space. According to the property of HMMs, the likelihood of a pattern x for a HMM λ_i is computed by

$$\begin{aligned}
 P(x|\lambda_i) &= \pi(s_1)b_{s_1}(x_1) \prod_{t=2}^T a_{s_{t-1}s_t} b_{s_t}(x_t) \\
 &= \pi(s_1) \left[\prod_{t=1}^T b_{s_t}(x_t) \right] \left[\prod_{t=2}^T a_{s_{t-1}s_t} \right], \tag{4.13}
 \end{aligned}$$

where a , b and π are parameters of λ_i .

The performance of THMM in non-meaningful pattern rejection depends on the state characteristics of non-meaningful classes. Since HMM_3 is the closest meaningful class to HMM_4 , every observation generated from HMM_4 will be assigned to a state in HMM_3 in the THMM. This result in that, for a pattern from HMM_4 , the state sequences decoded by HMM_3 and the THMM are identical. Thus the

likelihood of the pattern in the THMM is larger than that in HMM_3 only if

$$\pi(s_1) \prod_{t=2}^T a_{s_{t-1}s_t}^{THMM} > \pi(s_1) \prod_{t=2}^T a_{s_{t-1}s_t}^{HMM_3}. \quad (4.14)$$

where $a_{s_{t-1}s_t}^{THMM}$ is the transition probability from s_{t-1} to s_t in THMM, $a_{s_{t-1}s_t}^{HMM_3}$ is the transition probability from s_{t-1} to s_t in HMM_3 . This means, let x be a sequence from HMM_4 , the difference between $P(x|HMM_3)$ and $P(x|THMM)$ only depends on the dynamics of each model. Thus $P(x|HMM_3)$ has a high chance to exceed $P(x|THMM)$.

Another limitation of THMM is that THMM is sensitive to distortions of the density distribution of each meaningful state. To illustrate this comment, another experiment is conducted in which HMMs are set as that in Figure 4.8. The structures

Figure 4.8: Settings of HMMs in the second simulated experiment.

and parameters of HMM_1 and HMM_2 in this experiment are the same with that of HMM_1 and HMM_3 , respectively, in the previous experiment. A new HMM is set with two states having mixture gaussian density functions and used to replace

HMM_3 in the previous experiment. The components of the mixture density are illustrated by large ellipses in Figure 4.8. Each state in the new HMM_3 have two components whose weights are set to be equal. The transition matrix and prior probabilities of the new HMM_3 are also set to be A and π . One state in HMM_3 is overlapped with states of HMM_1 and HMM_2 by taking one state of each HMMs as its components. All the three classes are viewed as meaningful classes.

In this experiment, HMM_1 and HMM_2 are trained with correct structure and sufficient samples so that the estimated parameters are close to the true values. But in the case of HMM_3 , single Gaussian density distribution is used in training for each state instead of the true mixture Gaussian density function. In this way the estimated density function in each state is biased while state-transition matrix and prior probability are close to true values. The likelihoods of 20 samples from HMM_3 for its own HMM and THMM are shown in Figure 4.9. It can be seen that at some

Figure 4.9: Likelihoods of 20 samples from HMM_3 in the second experiment for THMM and HMM_3 .

points the likelihoods for THMM are higher than that for HMM_3 . This means that some meaningful patterns will be rejected. Let O be a sample from HMM_3 , since the density distributions of the states of HMM_3 are not accurate, its likelihood for HMM_3 may be smaller than that for THMM. The chance that $P(O|THMM)$ exceeds $P(O|HMM_3)$ depends on the distortion of the trained distribution of each state to their true values. This is because that THMM inherit all states from the meaningful classes. For each observation, THMM can provide a better state than HMM_3 (the state is either from HMM_1 or HMM_2), since the distributions of the states in HMM_3 are not accurate, and they do not represent their own observation better than those states from HMM_1 or HMM_2 . The more distortion, the higher chance that O is rejected by THMM. In other words, THMM requires that the parameters of HMMs must be accurate so that it can perform well. However, the proposed algorithm, PLRT, is not so sensitive to the distortions in the estimated distributions of the states. Figure 4.10 shows the Mahalanobis distances from PLRs of HMM_1 , HMM_2 and HMM_3 samples to the PLR-distribution center of HMM_3 . It is clear that the distribution of PLRs of HMM_3 does not overlap with those of HMM_1 and HMM_2 , even though the distributions of the states are not accurate. Therefore using the Mahalanobis distance as the rejection criteria will not falsely reject HMM_3 samples.

In human activity recognition, the distribution of each state in activity models is always biased, since the feature used for modeling will not be highly representative. The assumption about the form of density distribution functions may also be in-

Figure 4.10: Histograms of the Mahalanobis distances from the 100 PLRs of each class to the PLR-distribution center of HMM_3 .

accurate, since the distribution is decided experimentally rather than theoretically. Thus THMM may meet difficulties in rejecting human activities. This is also illustrated by the comparison of the performances of the PLR test with that of THMM using real data.

4.4 Experiments with Real Data

The experiments with real data are based on the database that is used in Section 2.5.4. The setting of the HMMs for activity classes are the same as those adopted in Section 2.5.4. In order to show the effectiveness of our proposed algorithm, its performance is compared with that of LRT-cohort [131], LRT-PBM [140] and THMM [90]. Three kinds of tests are performed, namely *leave-one-out test*, *group-based test* and *random activity test*. These three tests are presented in the following.

4.4.1 Leave-One-Out Test

In this experiment, each activity is selected as a non-meaningful class once, while leaving the remaining thirteen classes as meaningful ones. This is to test whether the rejection algorithm can reject activities similar to meaningful activities. For example, SitDown is similar to StandUp having almost the same set of observed postures but with reversed dynamics of posture changing; JumpForward and JumpFrog have similar postures in their middle part of the activities but have different initial and ending postures. It can be observed that these kind of similarities exist between activities shown in Figure 2.7. A good algorithm should have the ability to reject non-meaningful patterns even though they are similar to one or more meaningful classes.

In order to measure the performances of these tests, the receiver operating characteristic (ROC) curves for these tests are estimated. The ROC curves for the candidate tests are determined by selecting different scales of the threshold in each algorithm, such that various number of false alarms can be observed. A set of distance threshold is tested for each class,

$$\{\epsilon_i(j) \mid \epsilon_i(j) = D_i^{max} * j, j = 1, 2, \dots, 50\},$$

where D_i^{max} is the maximum Mahalanobis distance among the PLR vectors of training samples. It should be noted that the adaptive threshold provided by THMM does not need to be scaled, since the threshold HMM works like other meaningful

Figure 4.11: Average ROC curves for (a) LRT-Cohort, (b) LRT-PBM and (c) PLR test based on leave-one-out test.

HMMs and the ML rule can be directly applied. The fraction of true detections for all meaningful patterns is defined as the *detection probability*, P_D . The fraction of false detections for all non-meaningful patterns is defined as the *false-positive-alarm probability*. There are two other kinds of false alarms: The false negative alarm, denoting that a meaningful pattern is rejected as non-meaningful, and the insertion error that a meaningful pattern is accepted as meaningful but given a wrong class label. In the experiments conducted in this section, the insertion error is zero for all meaningful classes. Thus the false-negative-alarm rate equals to $1 - P_D$. Therefore, the ROC curves are estimated as the total fraction of true detections for

a given false-positive-alarm rate only. Figure 4.11 shows their ROC curves derived by leave-one-out test based on our activity database.

It can be seen that our proposed algorithm performs better than LRT-cohort and LRT-PBM by providing a higher detection rate with a lower false-positive alarm rate. The similarity existing between meaningful and non-meaningful patterns deteriorate the performances of the two LRT-based algorithms.

Table 4.1: Detection and rejection results of THMM

Select Class	Detection Rate	False Positive Rate
SitDown	0.783	0
StandUp	0.758	0
PointTo	0.734	0
Squat	0.747	0
SquatUP	0.758	0
LieDown	0.769	0
LieUp	0.775	0
Paunch	0.717	0
Kick	0.719	0
BendDown	0.738	0
BendUp	0.722	0.16
Jump	0.746	0
Jumpup	0.734	0
JumpFrog	0.742	0

The recognition result of THMM is shown in Table 4.1. Except when using BendUp action as the non-meaningful class, THMM rejects all the non-meaningful patterns. However, due to the insufficient discrimination of Hu-moment features, THMM provides an even higher likelihood value than the true HMM. As a result, a portion of meaningful patterns are rejected as non-meaningful, since the threshold HMM fits them most. This result is consistent with that obtained using simulated

data.

One problem might arise from our proposed method in real applications: When the number of meaningful classes becomes extremely large, the dimension of pairwise log-likelihood ratio vectors will become too large to deal with. This is possible since we have seen that modern speech recognition system need to classify tens of thousands words and phrases. A further experiment is conducted to test the effects caused by various number of support classes. For an activity class HMM_i , its support classes are ranked by the their average log-likelihoods for patterns from HMM_i . The first N classes with the highest ranks are selected to construct the pairwise log-likelihood ratio vectors. The result is shown in Figure 4.12.

Figure 4.12: ROC curves of different number of support classes.

It can be seen that the performance of the PLR test depends on the number of support classes. The more support classes used the better is the performance. It can also be seen that the performance of using ten support classes is close to that of

Table 4.2: Rejection Result of THMM using Group 1 as meaningful set, Group 2 as non-meaningful set.

Meaningful Class	Number	Accepted	Rejected	Missed	Detection Rate
SitDown	54	22	32	0	0.407
SquatDown	45	40	5	0	0.889
LieDown	48	44	4	0	0.917
BendDown	40	40	0	0	1.0
JumpForward	42	34	8	0	0.81
PointTo	40	36	4	0	0.9
Kick	50	48	2	0	0.96
Non-Meaningful Class	Number	Accepted	Rejected		Rejection Rate
StandUp	54	0	54		1.0
SquatUP	45	0	45		1.0
LieUp	48	0	48		1.0
BendUp	40	0	40		1.0
JumpFrog	39	0	39		1.0
Paunch	45	0	45		1.0
Jumpup	42	0	42		1.0

using twelve support classes. This shows the existing trade-off between performance and number of support classes.

4.4.2 Group-based Test

In this test, the activities are divided into two groups. Group 1 contains activities shown in the left column of Figure 2.7, including SitDown, SquatDown, LieDown, BendDown, JumpForward, PointTo and Kick. The rest of the activities are shown in the right column of Figure 2.7 from Group 2. If either group is selected as the set of meaningful activities, the other group will be composed of non-meaningful activities similar to the meaningful one.

Figure 4.13: The ROC curves of (a) LRT-cohort, (b) LRT-PBM and (c) PLR test, using Group 1 as meaningful set, Group 2 as non-meaningful set.

Figure 4.13 shows the ROC curve of LRT-Cohort, LRT-PBM and PLR test using Group 1 as the set of meaningful activities and Group 2 as the set of non-meaningful activities. Again, the PLR test performs better than the other LRT-based algorithms. The result of THMM in the same condition is shown in Table 4.2. THMM rejects all non-meaningful activities successfully but also reject many meaningful activities simultaneously.

The results of the tested algorithms by using Group 2 as the meaningful set and Group 1 as the non-meaningful set are shown in Figure 4.14 and Table 4.3. The

Figure 4.14: The ROC curves of (a) LRT-cohort, (b) LRT-PBM and (c) PLR test, using Group 2 as meaningful set, Group 1 as non-meaningful set.

performance of PLR test is still better than that of other algorithms.

4.4.3 Random Activity Test

In this test, all the fourteen activities are used as meaningful activities. The set of non-meaningful activities are built by 196 random activities including random walking, running and other 'unseen' activities. Figure 4.15 shows several examples of these 'unseen' activities.

The results of the tested algorithms are shown in Figure 4.16 and Table 4.4. In

Figure 4.15: Examples of the 'unseen' activities

Table 4.3: Rejection Result of THMM using Group 2 as meaningful set, Group 1 as non-meaningful set.

Meaningful Class	Number	Accepted	Rejected	Missed	Detection Rate
StandUp	54	30	24	0	0.556
SquatUP	45	45	0	0	1.0
LieUp	48	34	14	0	0.708
BendUp	40	40	0	0	1.0
JumpFrog	39	34	5	0	0.872
Paunch	45	28	17	0	0.622
Jumpup	42	30	12	0	0.714
Non-Meaningful Class	Number	Accepted	Rejected		Rejection Rate
SitDown	54	0	54		1.0
SquatDown	45	0	45		1.0
LieDown	48	0	48		1.0
BendDown	40	0	40		1.0
JumpForward	42	0	42		1.0
PointTo	40	0	40		1.0
Kick	50	0	50		1.0

this test, PLRT also performs better than other algorithms. The large number of meaningful classes aggravates the overlaps between the meaningful classes; hence the performance of THMM becomes even worse.

4.5 Summary

This chapter presents an algorithm to distinguish meaningful and non-meaningful patterns in pattern classification systems based on hidden Markov models (HMMs). The classification task is decomposed into two steps: Firstly, input patterns are classified by HMMs using the maximum likelihood (ML) rule. Secondly, the classification results are verified through our proposed pairwise likelihood ratio (PLR)

Figure 4.16: The ROC curves of (a) LRT-cohort, (b) LRT-PBM and (c) PLR test, using all activities as the meaningful set and random activities as non-meaningful.

test and invalid classification results of non-meaningful patterns are rejected. In the second step, a new statistic is used as the measurement for PLRs instead of using a truncating threshold which is widely adopted in traditional likelihood-ratio-test (LRT) based algorithms. Both simulated and real experiments have been conducted to show that the proposed algorithm is better than the other existing algorithms in non-meaningful pattern rejection.

Table 4.4: Rejection Result of THMM using all activities as the meaningful set and random activities as non-meaningful.

Meaningful Classes	Number	Accepted	Rejected	Missed	Detection Rate
SitDown	54	20	34	0	0.588
StandUp	54	25	29	0	0.441
SquatDown	45	37	8	0	0.822
SquatUP	45	20	25	0	0.444
LieDown	48	34	14	0	0.708
LieUp	48	33	15	0	0.688
BendDown	40	40	0	0	1.0
BendUp	40	39	1	0	0.975
JumpForward	42	30	12	0	0.714
JumpFrog	42	33	9	0	0.786
PointTo	40	36	4	0	0.9
Paunch	45	34	11	0	0.756
Kick	50	47	3	0	0.94
Jumpup	42	33	9	0	0.786
Meaningful Class	Accepted	Rejected		Rejection Rate	
Random	0	196		1.0	

Chapter 5

Modeling Multi-Agent Activities

In previous chapters, the discussion is concentrated on individual activity recognition. However, in real applications, the scenes of interest often contain multiple persons and complex interactions between them. Figure 5.1 shows eight key frames of a multi-agent activity simulating a ‘Snatch Theft’, which contains three persons. Understanding the interaction between these persons is another crucial task for the HAR system. This kind of recognition is called multi-agent activity recognition, in which an ‘agent’ refers to an active object that is autonomous to achieve its goal, can perceive other agent’s state and cooperate with others if it is necessary.

Multi-agent activity recognition has several particular difficulties, which makes the recognition of multi-agent activities more difficult than that of individual activities. The first difficulty is the *large feature space problem*. In visual surveillance system, the information obtained in each frame is always represented by a feature vector. In the case of multi-agent activity recognition, since the information required

Figure 5.1: Key frames of an activity simulating a multi-agent ‘Snatch Theft’.

by the system comes from multiple agents, the feature vector is normally obtained by combining all the individual information into a single vector. This causes the length of the feature vector to increase proportionally with the number of agents in the scene. Generally, large feature spaces suffer from the curse of dimensionality with increasing errors in vector clustering or vector quantization.

The second difficulty is the *changing number of agents*. Unlike the recognition of individual activities or gestures, which have a constant number of agents throughout the whole activity sequence, multi-agent activity recognition faces the problem of changing number of agents. This change may be caused by occlusion of objects, tracking failure or loss of track due to a restricted monitoring area, which often happen in multi-agent activities. Such changes also exist in some particular activities in which the number of agents inherently changes, depending on the characteristic of the activities of interest. As a result, the feature vectors derived from a multi-agent activity may not belong to the same feature space. This requires that the multi-

agent activity recognition system should have the ability to deal with the change of feature spaces.

The third difficulty is about the *agent assignment*. Agent assignment is a special problem of multi-agent activity recognition. To recognize a multi-agent activity, we must map each agent in the data set to an agent in the activity model. However, for an activity that contains N agents, there are $N!$ possible alignments. That is to say, if the combined feature is used, there are $N!$ possible ways to form the representing feature vector for N channels of individual information. If we know the correct agent assignment for our input data, the computational cost of matching can be reduced considerably. However, it is nearly impossible to know the role of each agent even before the activity is identified. Perhaps, the only way to deal with the agents' roles is to view the roles of the agents as unknown parameters and integrate the matching values of all feasible role assignments to remove the effect of such parameters. Obviously, this increases the computational cost considerably. Even if we can integrate these unknown parameters, the above method is still not desirable, since we may prefer to know the exact roles of the agents. For example, in security visual surveillance system, we want to know both the occurrence of an incidence and the discrimination between criminal and victim in order to apply a weighted tracking. Therefore, a multi-agent activity recognition scheme is expected to give both a matching of the input data with an activity model and an estimation of the agent assignment.

The purpose of this chapter is to present a new method which can model and

recognize multi-agent activities directly from raw image sequences. The proposed method is based on the hidden Markov model (HMM). Traditional HMMs require that at each time step there should be only one observation and observations observed at different time steps must be of the same length [91, 122]. These two requirements make it difficult to apply traditional HMMs in multi-agent activity recognition which may contain a variable number of agents and varying length of the observations as described previously. To overcome these difficulties, a new structure of HMM is proposed which we termed as observation decomposed hidden Markov model (ODHMM). In ODHMM, the original observation is decomposed into a set of sub-observations in order to keep the dimensionality of the feature space consistent. Sub-observations belonging to the same agent at different time steps are organized by a novel role parameter, which is designed to realize fast mapping of agents from input data to activity models.

The rest of this chapter is arranged as follows: Section 5.1 reviews previous works related to multi-agent activity recognition. Section 5.2 describes the theoretical basis of our proposed method. Both discrete and continuous version of ODHMMs are discussed. Section 5.3 presents experiments with seven street activities to show the performance of our proposed method. The sensitivity of our method to incomplete data is also presented. An automatic outlier detection method based on ODHMMs is proposed in Section 5.4. Section 5.5 presents the experimental results of the proposed outlier detection method.

5.1 Related Work On Multi-Agent Activity Recognition

To model interactions between multiple persons or that between persons and other objects, an event-based approach is often used. In this approach, an activity is assumed to be composed of a set of simple events and is classified based on the relationship between these events. The advantage of using an event-driven approach is that events are easy to detect and quite representative of activities in which the interactions are not complex. Its shortcoming is that it often requires designers to provide details of the activity and define all possible events as well as their relationships in advance. This requirement results in that event-based approaches are always project dependent.

In the work by Rota and Thonnat [133], events were detected by observing changes in the object states, which were defined by n-ary trees. The activities were recognized by analyzing these events in a semantic way. Ivanov and Bobick [74] proposed an event-based work to recognize interactions between vehicles and people in a parking lot. Events were generated using a simple environmental model. A context-free stochastic parsing algorithm was used to analyze these events and identify input interactions. Hongeng et al. [63] adopted a chained hierarchical representation of activities. At the lower level, the objects were tracked and simple events are recognized from the objects' properties. At the higher level, a finite-state automaton was used to verify the dependencies of these detected events. Intille and

Bobick [73] employ Bayesian networks to detect individual goals (similar to events) of players in the American football match. These individual goals and their relationships are then used to form multi-agent temporal goal belief networks for each activity respectively. An input activity is recognized by searching for its maximum likelihood with these belief networks. Differing from others, Wada and Matsuyama [154] use Non-deterministic Finite Automaton (NFA) as an event sequence analyzer to recognize multi-agent behaviors. At each time step, all feasible NFA active states are generated and then verified by finding their supporting events in subsequent images.

The event-based approach encounters difficulties when modeling activities that contain a large number of agents, and when the interactions between these agents are complex. For example, in a soccer game, when an attacking player has possession of the football, the event that a defending player approaches him is common. But the way in which the defender approaches the attacker affects the attacker's decision on his next action. Therefore, events describing the different approaches must be defined in order to represent such an activity meaningfully, which is often intractable.

Oliver et al. [114] propose a novel Coupled HMM (CHMM) to model the interaction between two persons. In this work, interactions between two persons can be recognized directly from the raw image features without utilizing any events. The CHMM is an extension of the original HMMs by combining two individual HMMs at the state level. However, this method is difficult to be extended to model interactions between more than two persons, since it is intractable to combine more

than two HMMs in practice. Gong and Xiang [56] proposed a dynamically multi-linked HMM (DML-HMM) to interpret group activities. DML-HMM has a similar structure to that of CHMM, but with a reduced number of permitted state transitions. Since DML-HMM has a smaller number of parameters, the inference process of DML-HMM is easier than that of CHMM.

Our proposed method is also based on the framework of HMMs and aims to model and recognize multi-agent activities directly from input observations, without seeking any help from the use of events. The details of the proposed method are described in the following sections.

5.2 Observation Decomposed HMM (ODHMM)

5.2.1 ODHMM Structure

In conventional HMMs, each state of an HMM should only have one fixed-length observation whose elements are arranged in a fixed order (i.e. all the observations are in the same feature space). Figure 5.2 shows the structure of a HMM with full state transition. This requirement becomes an obstacle for applying HMMs in multi-agent activity recognition. As mentioned previously, the problem of ‘changing number of agents’ causes the feature length to vary and the problem of ‘agent assignment’ makes it difficult to obtain the correct order of the elements in the feature vector. To overcome the above problems, this section presents an approach, the Observation Decomposed HMM (ODHMM), to model hidden Markov processes for multi-agent

activities. The key idea is to decompose each original observation, according to the number of agents, into a set of sub-observations. These sub-observations are of the same length and are assumed to be independent. In the hidden Markov chain, each state emits a set of sub-observations instead of a single observation. The ODHMM is shown in Figure 5.3 where the white circles represent the states and the black circles represent the sub-observations. The number of sub-observations under each state is equal to the number of agents in the scene.

Figure 5.2: Traditional HMM: S_t and O_t are the state sequence and observation sequence respectively, where $t = 1, 2, \dots, M$.

The decomposition of the original observations into sets of sub-observations offers several advantages. Firstly, the new structure allows changes in the number of agents since multiple sub-observations, of arbitrary number, can be generated by each hidden state. Secondly, it is not necessary for an initial agent assignment when

Figure 5.3: The Observation Decomposed HMM: o_t is decomposed into multiple layers, and each layer is associated with a role parameter r_j ($j = 1, 2, 3$ in this example); the role parameters are combined as a vector h .

computing the likelihood between an input pattern and an ODHMM. For example, vectors $[A, B, C]$, $[A, C, B]$ and $[B, A, C]$ will have the same likelihoods with a given ODHMM, since they all produce the same set of sub-observations $[A]$, $[B]$ and $[C]$. Thirdly, and obviously, by decomposing the original observations, the ODHMM reduces the dimensionality of the feature space considerably and this helps to cluster processes such as density estimation and Vector quantization. Another significant advantage of ODHMM is that the optimal state chain can be obtained in situations where information of some agents is lost for a period of time. This is because the optimal state at each time step is inferred by considering the output probabilities of a set of sub-observations instead of the probability of a single observation. When some sub-observations are lost, the remaining sub-observations still carries information which may help ODHMMs to verify which state is more likely to be the optimal state for that time step.

On the other hand, a limitation of ODHMMs is that they have difficulties in distinguishing classes that are inherently different but have similar decomposed sub-observations. This is because that ODHMMs only consider the sub-observations being observed at each time step but neglect their relationships over time. Another aspect of this limitation is that ODHMMs cannot preserve the details of the modeled activities which might be of high interest in particular tasks. For example, in multi-agent activity recognition, we are interested in ‘what is the role of each agent’ besides ‘what are these agents doing’. In visual surveillance, not only do we want to know what was happened but also would like to have some extra information

like ‘who is the criminal’ and ‘who is the victim’, when observing an incident. In order to compensate for this omission of role information, a hidden component h is introduced into the ODHMM to represent the ‘role’ of each agent. Each element h_i represents an alignment between an agent in the data set and an agent in the activity model. For example, if an activity includes three agents labeled by 1, 2 and 3, then $h = [2 \ 1 \ 3]$ has actually three alignment pairs and represents that Agent 1 plays the second role, Agent 2 plays the first role and Agent 3 plays the third role in the activity model. The role parameter is illustrated in Figure 5.3. Hence, an ODHMM contains two hidden components: the hidden state path s and the hidden agent alignment. In order to estimate h , a new parameter R is introduced into the ODHMM, which represents the output probability of each role.

5.2.2 Parameter Estimation in ODHMM

There are two sets of parameters that have to be estimated in ODHMMs. The first set includes the parameters identical to that of original HMMs, such as state initialization probabilities, state transition probabilities and output probabilities of each state. The second set contains parameters that are used to describe the distributions of sub-observations in each ‘role’.

In traditional HMMs, an HMM can be represented by the notation $\lambda_{HMM} = (A, B, \pi)$. Parameters A , B and π represent:

- $A = \{a_{ij} \mid a_{ij} = P(s_{t+1} = j \mid s_t = i), \text{ for } i, j = 1, 2, \dots, N\}$ is the state transition probability matrix,

- $B = \{b_j(o_t) | b_j(o_t) = P(o_t | s_t = j) \text{ for } j = 1, 2, \dots, N\}$ is the output probability for each state,
- $\pi = \{\pi_i | \pi_i = P(s_1 = i) \text{ for } i = 1, 2, \dots, N\}$ is the initialization probability for each state,

where o_t is the observation at time t and N is the number of states.

Similarly, an ODHMM can be represented as $\lambda_{ODHMM} = (A, B, \pi, R)$. A , B and π are identical to that in traditional HMMs. R is defined as

$$R = \{R_j^r(o_t^r) | R_j^r(o_t^r) = P(o_t^r | s_t = j, r)\}, \text{ for } j = 1, 2, \dots, N, r = 1, 2, \dots, Z.$$

where o_t^r is the sub-observation of the r th agent at time t and Z is the number of agents in the model. Comparing with the definition of B , R_j^r can be viewed as the output probability of role r in state j .

Since the state structure of the ODHMM remains the same as compared with that of traditional HMMs, the estimation of parameters A and π in ODHMMs remains the same. Only the estimation of B need to be modified in ODHMMs.

In ODHMM $b_j(o_t)$ is computed by multiplying the output probabilities of all the sub-observations in each frame. This is given by

$$b_j(o_t) = \prod_{r=1}^{Z_t} b_j(o_t^r), r = 1, 2, \dots, Z \quad (5.1)$$

where Z_t is number of agents at time step t , $o_t = \{o_t^1, o_t^2, \dots, o_t^{Z_t}\}$, is the original

observation, and o_t^r is the sub-observation of agent r in o_t .

Since the output probability $b_j(o_t)$ is changed in ODHMM as shown in Eq. 5.1, the formulas to estimate B in an ODHMM is different from those in a traditional HMM (see Section 2.5.2 for the estimation formulas of traditional HMMs). In the following sections, the estimation of B , after using the new $b_j(o_t)$, in both discrete and continuous version of ODHMMs are described. The classical Lagrange method is used in the estimation of the discrete version of ODHMM, since the estimation of B is an optimization problem, whereas the EM algorithm is used in the continuous version of ODHMM. For the usage of Lagrange method and EM algorithm in traditional HMM, the readers are referred to [70].

Discrete ODHMM

For a discrete ODHMM with a codebook C , the probability of observing a codeword k in a state j is represented by a discrete value $b_j(k)$ which is normally stored in a lookup table B . The estimation of each element $b_j(k)$ in B is always operated by updating its value iteratively to achieve the maximum value of $\log P(O|\lambda)$. According to the constraints of HMM, the updated models must satisfy the constraint

$$\sum_{k=1}^C b_j(k) = 1. \quad (5.2)$$

The estimation of $b_j(k)$ thus can be viewed as a constrained optimization of $\log P(O|\lambda)$.

This optimization problem can be easily solved using the Lagrange method. Let Q

be the auxiliary function with respect to the constraints on $b_j(k)$ which is given by

$$Q = \log P(O|\lambda) + \sum_j l_j \left(\sum_k b_j(k) - 1 \right), \quad (5.3)$$

where the l_j are the Lagrange multipliers. The critical point of $\log P(O|\lambda)$ can be obtained by using the partial differential function

$$\frac{\partial Q}{\partial b_j(k)} = \frac{1}{P(O|\lambda)} \frac{\partial P(O|\lambda)}{\partial b_j(k)} + l_j = 0. \quad (5.4)$$

From Eq. (5.4), the critical point for $b_j(k)$ can be derived by [70]

$$\bar{b}_j(k) = \frac{b_j(k) \frac{\partial P(O|\lambda)}{\partial b_j(k)} \frac{1}{P(O|\lambda)}}{\sum_m b_j(m) \frac{\partial P(O|\lambda)}{\partial b_j(m)} \frac{1}{P(O|\lambda)}}. \quad (5.5)$$

$P(O|\lambda)$ can be expressed as

$$P(O|\lambda) = \sum_i \sum_j \alpha_t a_{ij} b_j(o_{t+1}) \beta_{t+1}(j). \quad (5.6)$$

Replacing $b_j(o_{t+1})$ by Eq. (5.1), we have

$$\frac{\partial P(O|\lambda)}{\partial b_j(k)} = \sum_{t \in \{\delta(o_t, k)=1\}} \alpha_t(i) a_{ij} \frac{\partial \prod_{r=1}^{Z_{t+1}} b_j(o_{t+1}^r)}{\partial b_j(k)} \beta_{t+1}(j), \quad (5.7)$$

where the δ function is defined by

$$\delta(o_t, k) = \begin{cases} 1 & \text{if } o_t \text{ contains sub-observations equivalent to } k; \\ 0 & \text{otherwise.} \end{cases}$$

Using Eq. (5.7) to substitute for $\partial P(O|\lambda)/\partial b_j(k)$ in Eq. (5.5) gives

$$\bar{b}_j(k) = \frac{\frac{1}{P(O|\lambda)} \sum_{t \in \{\delta(o_{t+1}, k)=1\}} \alpha_t(i) a_{ij} b_j(k) \frac{\partial \prod_{r=1}^{Z_{t+1}} b_j(o_{t+1}^r)}{\partial b_j(k)} \beta_{t+1}(j)}{\frac{1}{P(O|\lambda)} \sum_{m \in C} \sum_{t \in \{\delta(o_{t+1}, m)=1\}} \alpha_t(i) a_{ij} b_j(m) \frac{\partial \prod_{r=1}^{Z_{t+1}} b_j(o_{t+1}^r)}{\partial b_j(m)} \beta_{t+1}(j)}. \quad (5.8)$$

Noting that

$$\begin{aligned} b_j(k) \frac{\partial \prod_{r=1}^{Z_{t+1}} b_j(o_{t+1}^r)}{\partial b_j(k)} &= b_j(k) \frac{\partial \{[b_j(k)]^{N_{t+1}(k)} \prod_{r \in \{o_t^r \neq k\}}^{Z_{t+1}} b_j(o_{t+1}^r)\}}{\partial b_j(k)} \\ &= b_j(o_t + 1) N_{t+1}(k) \{[b_j(k)]^{N_{t+1}(k)-1} \prod_{r \in \{o_t^r \neq k\}}^{Z_{t+1}} b_j(o_{t+1}^r)\} \\ &= N_{t+1}(k) b_j(o_{t+1}), \end{aligned} \quad (5.9)$$

where $N_{t+1}(k)$ is the number of sub-observations that takes codeword k in frame t ,

Eq. (5.8) can be rewritten as

$$\bar{b}_j(k) = \frac{\sum_{t \in \{\delta(o_{t+1}, k)=1\}} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) N_{t+1}(k)}{\sum_{m \in C} \sum_{t \in \{\delta(o_{t+1}, m)=1\}} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) N_{t+1}(m)}. \quad (5.10)$$

Substituting the summation term in Eq. (5.10) by Eq. (5.12), the final updating equation is given by

$$\bar{b}_j(k) = \frac{\sum_{t \in \{\delta(o_{t+1}, k)=1\}} \gamma(i, j) N_{t+1}(k)}{\sum_{m \in C} \sum_{t \in \{\delta(o_{t+1}, m)=1\}} \gamma(i, j) N_{t+1}(m)}. \quad (5.11)$$

where

$$\begin{aligned} \gamma_t(i, j) &= \frac{\alpha_t(i) \alpha_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) \alpha_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \\ &= \frac{\alpha_t(i) \alpha_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}. \end{aligned} \quad (5.12)$$

Continuous ODHMM

For continuous ODHMMs, the output-probability densities for each state j can be modeled by a Gaussian Mixture density function, which is given by

$$b_j(o_t) = \prod_{r=1}^{Z_t} \sum_{q=1}^{Q_j} w_{jq} N(o_t^r, \mu_{jq}, \Sigma_{jq}), \quad (5.13)$$

where Q_j is the number of components in state j , w_{jq} is the weight for the q th component, μ_{jq} is the mean of the q th component in state j , Σ_{jq} is the covariance of the q th component in state j , and $N(\cdot, \cdot, \cdot)$ specifies the Gaussian component. The parameters of these density functions can be obtained using expectation maximization (EM) algorithm as is used for traditional HMMs [70]. EM updates the

parameters of a HMM λ iteratively to maximize the likelihood $P(O|\lambda)$.

The EM estimation process employs an auxiliary function given by

$$Q(\lambda, \bar{\lambda}) = \frac{1}{f(O|\lambda)} \sum_{all S} \sum_{all M} f(O, S, M|\lambda) \log f(O, S, M|\bar{\lambda}), \quad (5.14)$$

where $f(\cdot)$ is the likelihood function of λ , O is the set of training observations, S is the set of all possible state sequences, M is the set of all possible mixture components, λ and $\bar{\lambda}$ are the current model and updated model, respectively. It has been proven [70] that $Q(\lambda, \bar{\lambda}) \geq Q(\lambda, \lambda)$ implies $f(O|\bar{\lambda}) \geq f(O|\lambda)$, where $Q(\lambda, \bar{\lambda}) = Q(\lambda, \lambda)$ only when $f(O|\bar{\lambda}) = f(O|\lambda)$.

Similar to traditional HMM, $f(O, S, K|\lambda)$ is expressed as

$$f(O, S, M|\lambda) = \prod_{t=1}^T a_{s_{t-1}s_t} b_{s_t m_t}(o_t) k_{s_t m_t}. \quad (5.15)$$

Replacing $f(O, S, M|\lambda)$ by Eq. (5.15), Eq. (5.14) is rewritten as

$$Q(\lambda, \bar{\lambda}) = \sum_S \sum_M \frac{f(O, S, M|\lambda)}{f(O|\lambda)} \left[\log \bar{\pi}_{s_1} + \sum_{t=1}^{T-1} \log \bar{a}_{s_t s_{t+1}} + \sum_{t=1}^T \log \bar{b}_{s_t m_t}(o_t) + \sum_{t=1}^{T-1} \log \bar{k}_{s_t m_t} \right]. \quad (5.16)$$

It can be seen that maximizing $Q(\lambda, \bar{\lambda})$ is equivalent to separately maximizing each term within the square brackets in Eq. (5.16).

In ODHMM, only the part related to $\log \bar{b}_{s_t m_t}(o_t)$ need to be re-formulated. The

auxiliary function related to this part is given by

$$Q_{b_j}(\lambda, \bar{b}_{jm}) = \sum_{t=1}^T f(s_t = j, m_t = m | O, \lambda) \log \bar{b}_{jm}(o_t). \quad (5.17)$$

Maximization of $Q_{b_j}(\lambda, \bar{b}_{jm})$ with respect to $\bar{b}_{jm}$ can be obtained through differentiation given by

$$\sum_{t=1}^T f(s_t = j, m_t = m | O, \lambda) \sum_{r=1}^{Z_t} \frac{\nabla_{\bar{b}_{jm}} \bar{b}_{jm}(o_t^r)}{\bar{b}_{jm}(o_t^r)} = 0, \quad (5.18)$$

where $\nabla_{\bar{b}_{jk}}$ denotes derivatives with respect to parameters of $\bar{b}_{jk}$. For the mixture Gaussian density functions in ODHMM, partial derivatives $\nabla_{\bar{b}_{jk}}$ are related to μ_{jm} and Σ_{jm} .

$$\frac{\partial \bar{b}_{jm}(o_t^r)}{\partial \mu_{jm}} = N(o_t^r, \bar{\mu}_{jm}, \bar{\Sigma}_{jm}) \bar{\Sigma}_{jm}^{-1} (o_t^r - \bar{\mu}_{jm}), \quad (5.19)$$

$$\frac{\partial \bar{b}_{jm}(o_t^r)}{\partial \Sigma_{jm}^{-1}} = \frac{1}{2} N(o_t^r, \bar{\mu}_{jm}, \bar{\Sigma}_{jm}) \left[\bar{\Sigma}_{jm}^{-1} - (o_t^r - \bar{\mu}_{jm})(o_t^r - \bar{\mu}_{jm})^T \right]. \quad (5.20)$$

Putting Eq. (5.19) and Eq. (5.20) into Eq. (5.18), two equations can be derived as given by

$$0 = \sum_{t=1}^T f(s_t = j, m_t = m | O, \lambda) \sum_{r=1}^{Z_t} \left[\frac{N(o_t^r, \bar{\mu}_{jm}, \bar{\Sigma}_{jm}) \bar{\Sigma}_{jm}^{-1} (o_t^r - \bar{\mu}_{jm})}{N(o_t^r, \bar{\mu}_{jm}, \bar{\Sigma}_{jm})} \right], \quad (5.21)$$

$$0 = \sum_{t=1}^T f(s_t = j, m_t = m | O, \lambda) \sum_{r=1}^{Z_t} \left[\frac{\frac{1}{2} N(o_t^r, \bar{\mu}_{jm}, \bar{\Sigma}_{jm}) \left[\bar{\Sigma}_{jm}^{-1} - (o_t^r - \bar{\mu}_{jm})(o_t^r - \bar{\mu}_{jm})^T \right]}{N(o_t^r, \bar{\mu}_{jm}, \bar{\Sigma}_{jm})} \right]. \quad (5.22)$$

From Eq. (5.21), the updating equation for μ_{jm} is

$$\bar{\mu}_{jm} = \frac{\sum_{t=1}^T \xi_t(j, m) \sum_{r=1}^{Z_t} o_t^r}{\sum_{t=1}^T \xi_t(j, m) Z_t}, \quad (5.23)$$

and that for Σ_{jm} is

$$\bar{\Sigma}_{jm} = \frac{\sum_{t=1}^T \xi_t(j, m) \sum_{r=1}^{Z_t} (o_t^r - \bar{\mu}_{jm})(o_t^r - \bar{\mu}_{jm})^t}{\sum_{t=1}^T \xi_t(j, m) Z_t}, \quad (5.24)$$

where $\xi_t(j, m) = f(s_t = j, m_t = m | O, \lambda)$.

Role Models

The role parameter R can be obtained given the value of A , B and π in an ODHMM λ . For each role r in the activity to be modelled, there is a corresponding role parameter R which can be computed from the training samples. For discrete ODHMMs, $R = \{R_j^r\}$ is given by

$$\begin{aligned} R_j^r(k) &= P(k | s_t = j, r, \lambda) \\ &= \frac{T_\lambda(k, j, r)}{\sum_r T_\lambda(k, j, r)}, \end{aligned} \quad (5.25)$$

where $T(k, j, r)$ is the number of times of obtaining code word k in state j assigning to role r in an ODHMM λ with all the training samples. For continuous ODHMMs,

R_j^r is modelled by Mixture Gaussian densities, which is given by

$$R_j^r(o_t^r) = \sum_{q=1}^{Q_j^r} w_{jq}^r N(o_t^r, \mu_{jq}^r, \Sigma_{jq}^r), \quad (5.26)$$

where w_{jq}^r is the weight for the q th component in role r , Q_j^r is the number of components in role r and μ_{jq}^r is the mean of the q th component in role r and Σ_{jq}^r is the covariance for the q th component in role r . The parameters of these density functions can be obtained by the traditional EM algorithm for Mixture Gaussian density estimation.

5.2.3 Likelihood Computation

In conventional HMMs, the likelihood between an input sequence O and an HMM $\lambda = (A, B, \pi)$ is computed by

$$P(O|\lambda) = \sum_{s \in S} P(O, s|\lambda) = \sum_{s \in S} P(O|s, \lambda)P(s|\lambda), \quad (5.27)$$

where S is the set of possible state pathes.

After adding the hidden component h into ODHMMs, the likelihood of an ODHMM is represented by

$$\begin{aligned} P(O|\lambda) &= \sum_{h \in H} \sum_{s \in S} P(O, s, h|\lambda) \\ &= \sum_{h \in H} \sum_{s \in S} P(O|s, h, \lambda)P(s|h, \lambda)P(h|\lambda), \end{aligned} \quad (5.28)$$

where H is the set of feasible agent assignments.

It is difficult to compute the likelihood directly from Eq. (5.28) since all feasible assignments must be tested and the likelihood between an input sequence and an referring ODHMM must be computed repeatedly. This requires that all the possible state sequence and feasible agent assignments should be explored. Fortunately, this computation can be simplified. Noting that the estimation of the optimal state path $\hat{s}$ in ODHMMs is independent of any agent assignment, since the estimation of $\hat{s}$ is based on A and B , both of which are not affected by the value of h , the ODHMM structure allows us to explore the two spaces separately.

The term, $\sum_{s \in S} P(O|s, h, \lambda)$, in Eq. (5.28) can be replaced by a single probability $P(O|\hat{s}, h, \lambda)$, where $\hat{s}$ represents the optimal state path. Eq. (5.28) can be revised to

$$P(O|\lambda) = \sum_{h \in H} P(O|\hat{s}, h, \lambda) P(\hat{s}|\lambda) P(h|\lambda). \quad (5.29)$$

The simplification applied in Eq. (5.29) is in fact a Maximum Likelihood Estimation (MLE) of the optimal state path in ODHMMs which can be obtained efficiently using Viterbi algorithm [70].

Eq. (5.29) still contain a summation which cannot be operated in the logarithm domain. Similar to the replacing procedure used in Eq. (5.29), the MLE of the optimal agent assignment $\hat{h}$ can be used to remove the summation. This is given by

$$\sum_{h \in H} P(O|\hat{s}, h, \lambda) P(\hat{s}|\lambda) P(h|\lambda) \approx P(O|\hat{s}, \hat{h}, \lambda) P(\hat{s}|\lambda) P(\hat{h}|\lambda), \quad (5.30)$$

where

$$\hat{h} = \arg \max_h \{P(O|\hat{s}, h, \lambda)\}.$$

The terms $P(\hat{h}|\lambda)$ is the prior probability of the optimal agent assignment in λ . Normally, it is reasonable to treat it as constants, assuming that it has uniform distributions and remove it from Eq. (5.30). The final likelihood of ODHMM after adding the parameter h is given by

$$P(O|\lambda) = P(O|\hat{s}, \hat{h}, \lambda)P(\hat{s}|\lambda). \quad (5.31)$$

In practice, the likelihood of a given activity sequence with an activity model λ is computed as follows: Firstly, the optimal state path of the input sequence, $\hat{s}$, is derived using the Viterbi Algorithm. For each agent k in this input activity, the likelihood that agent k plays a role r is computed using R^r , the role parameter of role r , and $\hat{s}$ which is denoted as $M(k, r)$, where

$$M(k, r) = \prod_{t=1}^T R_{\hat{s}_t}^r(o_t^k). \quad (5.32)$$

The M values of all the agents are built up in a lookup table, the *agent assignment table*. An example of the agent assignment table is shown in Table 5.1, which is built for a three-agent activity with a three-role ODHMM.

Secondly, the likelihood is computed by searching through the agent assignment

Table 5.1: Agent Assignment Table.

	Role 1	Role 2	Role 3
Agent 1	M(1,1)	M(1,2)	M(1,3)
Agent 2	M(2,1)	M(2,2)	M(2,3)
Agent 3	M(3,1)	M(3,2)	M(3,3)

table for the optimal agent assignment $\hat{h}$ that maximizes $P(O|\hat{s}, h, \lambda)$

$$\begin{aligned}
P(O|\hat{s}, h, \lambda) &= \prod_{t=1}^T P(o_t|\hat{s}, h, \lambda) \\
&= \max_h \prod_{t=1}^T \prod_{k=1}^{K_t} P(o_t^k|\hat{s}, h, \lambda) \\
&= \max_h \prod_{k=1}^{K_t} M(k, h_k),
\end{aligned} \tag{5.33}$$

where h_k is the k th element in agent assignment h representing the role label of the k th agent in O . Obviously, the logarithm function can be easily applied to Eq. (5.33).

Although the above procedure requires the search for all feasible agent assignments, this search is based on the Agent Assignment Table, where only direct summations are operated as long as the logarithm function is used. The matching between an input sequence with an ODHMM to obtain the optimal state path is only operated once. If we adopt traditional HMMs in multi-agent activity modelling, a sequence must be matched with a given HMM N times where N is the number of feasible agent assignments. This is because the traditional HMMs assign a particular optimal state path for each agent assignment since the corresponding

feature vectors are unique. Obviously, the proposed ODHMM algorithm reduces the computational complexity considerably.

5.3 Experiments of ODHMM with Three-Person Interactions

This section presents the experimental results of the proposed method with simulated street surveillance data. Seven street activities are selected as test activities. Each activity involves three persons who are divided into two groups: two players A and B for Group 1 and a single player C for Group 2. The first four activities are classified as normal activities and the last three as abnormal ones in which A and B are considered as snatch thieves. The trajectories of these activities are described as follows and illustrated in Figure 5.4.

- Activity 1: The two groups approach in opposite directions. After meeting, A and C form a new group and turn to a new direction while B keeps going in his original direction.
- Activity 2: The two groups walk into the same direction, with Group 1 being ahead of Group 2. For some reason, Group 2 speeds up and catches up with Group 1. After stopping for a while, A and C form a new group and turn back while B keeps going in his original direction.

Figure 5.4: Trajectories of each player in the seven tested three-agent activities.

- Activity 3: The beginning of this activity is similar to Activity 1. However, after meeting, both A and C turn back, whereas B keeps going in the original direction and forms a new group with C .
- Activity 4: This activity also begins with two groups approaching each other as in Activity 1 and Activity 3. However, after meeting, both groups turn around and move in opposite directions. No new group is formed in this activity.
- Activity 5: In this activity A and B are a group of snatch thieves while C is the victim. Initially A and B approaches C from opposite directions. When A and C meet, A tries to attract C 's attention, B approaches to C , snatches C 's belongings and runs off in the direction that he came from. Then C begins to chase B .
- Activity 6: The beginning of this activity is similar to Activity 5. Before A and C meet, C finds that A looks suspicious and, upon turning around, finds B approaching him from the back. B , likewise, seems suspicious too, and sensing danger, C runs off in another direction with A and B chasing him.
- Activity 7: A and B are initially behind victim C . They follow C for a while and then A speeds up and catches up with C . Similar to Activity 5, A tries to distract C . B approaches C , snatches C 's belongings and runs off in a new direction with C giving chase.

The activity data set for training and testing are simulated with different persons.

The lengths of the captured image sequences range from 20 to 50 frames depending

(a)

(b)

(c)

Figure 5.5: The image sequence of Activity 5 (a), Activity 6 (b) and Activity 7 (c).

on the execution time of the activities. Since the contribution of this work is on the modeling part of the activity recognition, the tracking procedure is simplified by manually localizing the positions of each person within the activities.

Figure 5.6: Feature extraction. A_t and B_t refer to the positions of A and B at time t .

In order to recognize activities starting from different initial positions of the persons, the agents in the experiments are defined as a *pair of persons*. The relative information between any two people is extracted to build the feature vectors, thus making the classifier tolerant to the changes of the distance between the image plane and the camera and less sensitive to the initial position and orientation of each person. Owing to not having an initial agent assignment, we take player pair AB and player pair BA as different agents in order to avoid confusion. The feature vector of each agent is defined as a five dimensional vector

$$v(AB) = \left[\frac{|A_t A_{t+1}|}{|A_t B_t|}, \frac{|B_t B_{t+1}|}{|A_t B_t|}, \frac{|A_{t+1} B_{t+1}|}{|A_t B_t|}, \cos(\theta_A), \cos(\theta_B) \right], \quad (5.34)$$

where A_t and B_t are, respectively, the positions of agents A and B at time t and θ_A and θ_B are their relative angles as shown in Figure 5.6.

These feature vectors are then used as inputs to the ODHMMs. For each activity model 20 sequences are used for training. For discrete ODHMMs, a codebook

containing 256 codewords is built using the standard LBG [165] algorithm from the training samples. For continuous ODHMMs, the number of components is chosen to be 5 for the Mixture Gaussian densities in each state. ODHMMs are trained for each activity using the proposed training algorithm. Each ODHMM contains 4 states with typically left-right state transition topology. After obtaining the required parameters for each ODHMM, A , B , π and R , 20 ‘unseen’ new sequences are used as the test sequences for each activity. The total number of tested sequences is 140.

5.3.1 Recognition Rate

To compare the performance of the proposed ODHMM and that of traditional HMM, traditional HMMs with 4 states are constructed for the seven activities based on the assumption that correct agent assignments are known. As discussed previously, the correct agent assignment is difficult to obtain in practice. At each time step, the feature vector for traditional HMMs is obtained by combining all the agents’ features according to the correct agent-assignment. Similar to the experiment with ODHMMs, a codebook with 256 codewords is also built for vector quantization in discrete HMMs and the number of components for the Mixture Gaussian densities in each state is set to 5 in continuous HMMs.

The performances of using traditional HMM, using ODHMM-H (ODHMM without role parameters) and using ODHMM+H (ODHMM with role parameters) are compared and shown in Table 5.2. When computing the recognition rate in ODHMMs,

Table 5.2: Recognition Rate.

Activity No.		1	2	3	4	5	6	7
HMMS	discrete	1	1	1	1	1	1	0.8
	continuous	1	0.95	1	1	1	1	0.85
ODHMM-H	discrete	1	1	0.8	0.45	1	1	1
	continuous	1	1	0.9	0.5	1	1	1
ODHMM+H	discrete	1	1	1	1	1	0.9	1
	continuous	1	1	1	1	1	1	1

if a sample is classified correctly but assigned with a wrong agent assignment, the result is treated as a classification failure.

The result shows that traditional HMMs can obtain high recognition rates with most of the activities if correct agent assignments are given. But the recognition rates of the traditional HMMs with Activity 7 are inferior to that of our proposed ODHMM. This is mainly due to the use of combined features in traditional HMMs, whose high dimensionality results in more errors in vector quantization. The performances of ODHMM-Hs are obviously non-acceptable, since they cannot distinguish Activity 3 and Activity 4. The likelihoods of a sample of Activity 3 with ODHMM-Hs versus the lengths of the sample are shown in Figure 5.7 (a). The likelihoods between Activity 3 and Activity 4 are very close even when the length of the sample increase. Figure 5.7 (b) shows the result of the same experiment using ODHMM+Hs. It can be seen that the ambiguities between Activity 3 and Activity 4 decreases rapidly as the length of the sample increases. The overall performances of ODHMM+Hs are also better than that of traditional HMMs, though correct agent assignments are unknown and have to be estimated.

(a) ODHMM only

(b) ODHMM with role models

Figure 5.7: The likelihoods versus the lengths of a sample from Activity 3.

5.3.2 Sensitivity Analysis

Experiments were also conducted for situations where only partial information of the agents is available. In this experiment, one person is randomly selected and deleted

in a number of continuous frames. This is equivalent to deleting four agents in these frames according to our definition of the ‘agent’. The reason for deleting continuous frames rather than deleting frames randomly is that HMMs suffers more from the former, since losing continuous pieces of the input signal results in a severe break in the state chain in HMMs. In fact, traditional HMM cannot handle such a frame loss and is not suitable to be applied in this situation. In this experiment, the percentage of dropping frames ranges from 5% to 100% at interval of 5%, where dropping 100% means one person is completely deleted from the sequence. Consequently, each sample used in the first experiment can generate 20 new test sequences. That is to say each activity model has 20×20 test samples to analyze its sensitivity.

ODHMMs can handle the loss of frames robustly. Figure 5.8 shows the recognition rates of the total $20 \times 20 \times 7$ tested activity samples from the seven multi-agent activities in conditions where a person is lost in certain frames. The horizontal axis indicate the percentage of losing frames and the vertical axis indicate the corresponding recognition rate. The results show that there is no significant decrease in the recognition rate in most of the situations. Even though sub-observations are missing for certain periods, the rest of the sub-observations can still help the ODHMM find the ‘nearly’ optimal state at each time step. It is this mechanism that makes the ODHMM less sensitive to the change of the number of agents as compared to traditional HMMs. It should be noted that the curve shown in Figure 5.8 is not monotonic. This is because the recognition rates in this experiment depend not only on the percentage of frames lost but also on the importance (uniqueness)

Figure 5.8: Sensitivity analysis of the proposed model.

of the person deleted and the positions of the deleted person.

The sensitive analysis shows that even if the information of some agents are lost, ODHMM still can recognize the activity based on the remaining informations. This is an advantage of ODHMM to be used for multi-agent activity recognition, since losing of track of some targets often happens in real applications. Note that the computationally cost of finding the optimal state path in ODHMMs is almost the same as that in traditional HMMs, since ODHMMs only add a few extra additions in computing $P(o_t|s_t, \lambda)$ at each time step. The procedure of searching for the best agent assignment (Eq. (5.33)) may be computational expensive, since its computational cost grows as the number of agents increases. In our experiments, a direct search is adopted in which all the possible assignments are tested and the best one is chosen as optimal. For an even larger number of agents, this search can be accelerated by using some traditional optimization methods, such as the Hungarian

method, or a genetic algorithm (GA).

5.4 Recognition of Activities Involving Outliers

Most of the existing work assumes that all the agents in the captured scene are relevant to the activity taking place. Obviously, this assumption is not true in most real surveillance or video indexing tasks. Normally, outliers often exist in the scene that contains an activity of interest. It is difficult to distinguish these outliers from persons in interaction using a low level processing, since outliers have similar movement characteristics. Hence, a challenging question is: How can we filter out those outliers which do not belong to the activity even before the activity is recognized?

It is shown above that the recognition of activities with ODHMM highly depends on the accuracy of optimal state sequence, since the reliability of agent assignment in ODHMM relies on the optimal state sequence. Obviously, outliers in the scene will increase errors in optimal state estimation and thus result in a decrease in recognition accuracy. Therefore, the estimation of the optimal state sequence is the key to recognize activities involving outliers.

To obtain the optimal state sequence, only the observations belonging to reliable agents should be considered in the optimal state estimation process. The likelihood of each agent with the activity model, $P(o^r|\lambda)$, is selected as a measurement of each agent's reliability. This is inspired by the observation that outliers normally have a smaller likelihood than that of true agents. Figure 5.9 shows an example of the

Figure 5.9: Likelihoods of agents in a tested activity. Agents 1, 2 and 3 are real agents, and agents 4, 5 and 6 are outliers.

likelihoods of a three-agent activity with three outliers. Agent 1, 2 and 3 are real agents belonging to the activity while Agent 4, 5 and 6 are outliers. Obviously, the likelihoods of true agents are considerably higher than that of the outliers.

In multi-agent activity recognition, the description of an activity is often based on the relative information between each pair of persons in the activity and the ‘agent’ may be defined as a pair of persons rather than a single person. In this case, we are looking for K reliable persons rather than K reliable agents. In this work, the reliable persons are defined as persons with the highest marginal likelihood which is given by

$$L(i) = \sum_{j \in U_i} P(o^j | \lambda), \quad (5.35)$$

where o^j is the observation sequence of agent j , U_i is a set that contains agents

related to person i .

Take an input data containing M persons as an example. To compute its likelihood with an activity ODHMM model having N persons, where N is decided during training, we first derive all agents' observation sequences according to the definition of 'agent' and compute each agent's individual likelihood with the ODHMM. Then marginal likelihoods of each person in the input data is computed based on the individual agent's likelihood and K persons with the highest marginal likelihoods are selected as reliable persons. The optimal states are then computed from only the selected K persons' observation sequences. K is not necessarily equal to N . It can be less than N as long as the estimation error would not cause a significant drop in recognition rate. This is given by

$$\hat{s} = \max_{s \in S} P(O', s | \lambda), \quad (5.36)$$

where s is a feasible state sequence drawn from feasible-state space S , O' is the observations sequences reconstructed from K selected persons, and $K \leq N$.

With the derived optimal states, the likelihood of each agent in the scene with the activity model is computed to build the assignment table R , where $R(i, r) = P(o^i | r, s, \lambda)$ and o^i is the observation sequence of the i th agent. The classification is to find the most likely N players from M input persons. The total number of feasible solutions is C_M^N .

5.5 Experiment with Outliers

For each activity described above, 20 training samples are used to establish the corresponding ODHMM. The left-right state transition is used in training ODHMMs and the number of the states in each ODHMM is set to 4. In the testing process, ‘unseen’ samples are used. Three trajectories are manually added into each testing sample which increase the number of persons to six and the total number of agents to thirty (fifteen pairs of persons).

As described in Section 5.4, marginal likelihood is computed for each person in each activity ODHMMs. Three most reliable persons ($K = 3$) are used to estimate the optimal state sequence. The recognition result is shown in Table 5.3. In multi-agent activity recognition, there are two type of errors: The substitution error occurs when a multi-agent activity is classified falsely and the assignment error occurs when a multi-agent activity is classified correctly but with a wrong agent assignment. In this experiments, the reliability of the proposed method is defined by

$$\text{Recognition rate} = \frac{\# \text{ of correctly classified with correct assignment}}{\# \text{ of tested samples}}.$$

By using ODHMMs, the outliers can be removed from the input sequences automatically. The performance of using the trajectories of only two reliable persons to estimate the optimal states is also examined. The results are shown in Table 5.4

We can see from Table 5.4 that the reduction of the number of reliable persons

Table 5.3: Experimental results with proposed outlier removing method.

Acti- vities	Total number	Substitution Error	Incorrect Assignment	Recognition Rate (%)
1	23	0	3	86.96
2	21	0	0	100.00
3	20	0	0	100.00
4	20	0	0	100.00
5	24	0	2	91.67
6	22	2	0	90.91
7	22	1	0	95.45

Table 5.4: Experimental results of selecting the most reliable person

Acti- vities	Total number	Substitute	Incorrect assignment	Recognition Rate (%)
1	23	2	6	65.22
2	21	1	0	95.24
3	20	7	4	45.00
4	20	2	1	85.00
5	24	2	0	91.67
6	22	9	0	59.09
7	22	9	0	59.09

affects the performance of the proposed method significantly. This is because the information is not sufficient for the estimation of the optimal states, since only two reliable agents are selected. However, if the number of reliable persons is set to a large number, as we did in the previous experiments, it is difficult to ensure that all the reliable persons we select belong to the activity to be verified, especially when the number of persons in an activity model is extremely large. Therefore, a trade-off exists between recognition rate and number of reliable person that we should set.

5.6 Summary

It is a highly challenging task to recognize multi-agent activities automatically. The difficulties in this task stem from two aspects: Firstly, the feature vectors derived from input data are of large dimensionality and variable length. Secondly, an efficient mapping of agents from input data to predefined activity models, known as agent assignment, is required. This chapter presented a new method to model and classify multi-agent activities based on the proposed observation decomposed hidden Markov models (ODHMMs). To handle the feature vectors, each original feature vector is decomposed into a set of sub-feature vectors to keep the explored feature space consistent. Agent assignment is realized using a role parameter, which represents the ‘role’ of each agent. The experimental results have shown that the proposed method can successfully classify three-person activities with high accuracy and is less sensitive to incomplete data input. Based on the proposed ODHMM, an algorithm for outlier detection was also presented. The outlier detection algorithm uses the likelihood of each agent for an ODHMM as the measurement of the reliability of each agent. Only the agents with high reliability are selected for recognition of the activity. The experimental results have shown that the recognition rate depends on the number of selected reliable agents.

Chapter 6

Conclusions and Recommendation

6.1 Conclusion

This thesis has presented solutions for various problems, including automatic activity segmentation, non-meaningful activity rejection and multi-agent modeling, in HAR systems. These problems should be carefully considered for any HAR system that is designed for real-life applications.

The first part of this project concentrated on the automatic segmentation of activities from continuous video sequences. The task of activity segmentation is to find the boundaries of each activity within a video sequence. The objective of this part is to develop a segmentation method that does not require any alternative model representing non-meaningful activities. Therefore, the existing segmentation methods based on the forward-backward search or the Viterbi decoding cannot be adopted directly since they all require alternative models. To circumvent this problem, a

background state is added to the original HMM of each activity class to establish an activity detector. The parameters of this state is trained from meaningful samples, which simplifies the establishment of these activity detectors. The background state is used to label observations that do not belong to meaningful activities (irrelevant observations). Purely using the background state is not sufficient for the task of segmentation since the background state is not representative enough of irrelevant observations. Therefore, the pairwise likelihood ratio (PLR) test is used to validate if the activity segments detected by activity detectors are true detections. With the help of the PLR test, most of the false detections can be removed.

The second part of this work is to propose a rejection algorithm for activity spotting. The rejection of non-meaningful patterns is an important issue for most of activity recognition systems. This is because the number of activities of interest is much smaller than that of non-meaningful activities. The activities to be recognized are always merged in videos that contains both meaningful and non-meaningful signals. If the non-meaningful activities cannot be distinguished from meaningful ones, the system will give a high rate of false alarms.

The existing methods for non-meaningful pattern recognition are mostly based on likelihood ratio test (LRT). LRT requires that the likelihood of a testing pattern for non-meaningful classes must exist. The model for non-meaningful classes are always trained with a finite set of non-meaningful samples. However, such training may not be applicable in some situations, such as activity recognition where the set of non-meaningful classes is infinitely large and much larger than that of meaningful

classes. An interesting approach to solve this problem is using meaningful samples to build the model to fit non-meaningful patterns. A new algorithm for non-meaningful activity rejection is presented, which also does not use any non-meaningful patterns to train alternative models. The proposed algorithm does not use a truncating threshold as the criteria of rejection as other LRT-based algorithms do. Instead, the distribution of the pairwise likelihood ratios are used in determining if a pattern needs to be rejected. Patterns that do not satisfy the distribution will be viewed as non-meaningful patterns. The use of PLR distribution make the proposed algorithm performs well even when the meaningful space is very small and the parameter of the activity models are not perfectly defined.

The last part of this project focused on the modeling of multi-agent activities. The central difficulty in multi-agent activity modeling is to find the alignment between agents in data set and that in the activity model. To recognize a multi-agent activity, each agent in the activity should be aligned with an agent in the model. However, it is nearly impossible for the system to decide the correct alignments before it recognizes the activity. If the alignments can be obtained explicitly, the recognition would be straightforward without any further processing. In reality, the alignment cannot be established easily without assistance from activity models. The optimal way to find the correct alignment is to test all possible alignments between the agents in the data set and those in the model and choose the one with the maximum likelihood as the correct alignment. However, this would require the search to be conducted in a large solution space, and the dimension of this solution

space increases factorially with the number of the agents in the activity. Consider the activity analysis of players in a soccer game, a strategy may involve more than 10 players, which will significantly increase the number of the agents in its activity model. For activities whose models are based on HMMs, the computational cost of such a thorough search is intractable.

A new approach, the observation decomposed hidden Markov model (ODHMM), is proposed to handle multi-agent activity recognition. A novel role parameter is introduced in ODHMMs, which is used to realize efficient agent mapping from the data set to the activity model. ODHMM is capable of doing this because the optimal state sequence of each activity does not depend on the role of each agent, according to the structure of ODHMM. This characteristic makes it possible to obtain the optimal state sequence before the search for the optimal agent alignment. In ODHMM, the likelihood of an activity pattern for an activity model is only computed once rather than $N!$ times, where N is the number of agents in the activity. The use of ODHMMs decreases not only the computational cost, but also the dimensionality of the feature space even when the number of agents changes. The ODHMM accomplishes this by decomposing the observation vector into several sub-observation vectors of fixed length rather than maintaining a single, large vector. Therefore, no matter how many agents are observed, the length of the feature for representation is always the same, which simplified the modeling process. Our experimental results have also shown that the ODHMM is not sensitive to the loss of sub-observations. Based on the structure of ODHMM, outliers in a multi-agent activity can also be detected.

When outliers exist in the activity, the reliability of each agent to the activity is measured by its likelihood for the activity. Only the most reliable agents are selected in the recognition process so that the recognition result would not be interfered by the outliers.

6.2 Recommendation for Further Research

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

Activity Feature Selection In this thesis, we mainly concentrate on the segmentation, rejection and modeling issues of HAR. Simple features are selected for human activity representation in this project in order to reduce the complexity of feature selection. It is still an open problem to define optimal human activity features. As reviewed in Chapter 2, various representations of human activities have been proposed. With 2D images, it is nearly impossible to find an optimal representation of human activities which are not sensitive to changes of poses, background subtraction errors and view variations. A possible solution to this problem is to use domain specific representations for different applications. Each application of human activity recognition should have its own representation of activities, which need only to be optimal in its own domain rather than to be globally optimal. This would be very useful to expand our proposed works to different application domains.

Model-Based Body Detection Almost all the existing activity recognition tech-

niques rely on pre-processing of activity sequences including background subtraction, morphological operation, and contour or edge detection. These processes are purely based on cues such as pixel intensity, color and motion. Therefore, the detected regions in the image may not necessarily be human subjects. Other moving objects or some patches of the background may also be detected. This will increase the difficulty in human activity recognition. Although these issue can be alleviated by using a tracking system (in fact a tracking system is necessary to any activity recognition system), tracking systems are only good for data association and they are not sufficient in deciding whether a moving object is a human or not. An ideal solution to solve the pre-processing problem is to use machine learning methods to aid the detection process. Some good efforts have already been noticed in this direction [93, 102, 106, 152, 153]. By using machine learning, it is also possible to recover the 3D body-part configuration of human bodies. This can facilitate the modeling activities under view variations.

Improve the Performance of the Background State In our proposed method, the static background state is similar to a one-state HMM, which is trained using the same data in training activity HMMs. The performance of the background-state based activity detector can be improved by using more sophisticated construction methods. A possible improvement of the proposed method is to use some non-meaningful patterns in to build the background state. By adding non-meaningful patterns into the training process, the dis-

tribution of the observations in the background state can be made flatter and represents irrelevant observations better. However, the selection of irrelevant observations that are cooperated into the training process must be manipulated carefully. If the selected non-meaningful patterns are not spread around the meaningful space, the trained background state will be distracted from its true center. New criteria for irrelevant-observation selection should be made in order to make use of non-meaningful patterns. Intuitively, these criteria depends on the application domain. Different applications should have their own selection criteria.

Combining Individual and Multi-agent Activity Recognition In previous works,

individual and multi-agent activities are often treated separately. In real-life applications, those two kind of activities are required to be recognized simultaneously rather than individually. For example, in a tennis game, individual player's action and the interaction between two players are all important in deriving semantic description of the game. Therefore, it is necessary to combine the individual and multi-agent activity recognition techniques together to enhance the function of the activity recognition system. Such combination will increase the complexity of the system. New modeling structures should be proposed for such an integration. Applications such sport video indexing and visual surveillance will benefit from this combination.

Author's Publication List

- [1] Xiao-Hui Liu and Chin-Seng Chua, "Rejection of Non-Meaningful Activities for HMM-Based Activity Recognition System", submitted to *Image and Vision Computing*, 2006.
- [2] Xiao-Hui Liu and Chin-Seng Chua, "Multi-agent Activity Recognition with Observation Decomposed Hidden Markov Model", accepted on 7 September 2005 (available online on 21 November 2005), *Image and Vision Computing*, 2005.
- [3] Xiao-Hui Liu and Chin-Seng Chua, "Rejection of Non-meaningful Activities", in Proc. *International Conference on Automatic Face and Gesture Recognition*, 2006, Accepted.
- [4] Xiao-Hui Liu and Chin-Seng Chua, "Multi-agent Activity Recognition With Observation Decomposed Hidden Markov Model", in Proc. *Asian Conference on Computer Vision*, 2004, pp. 168-173.
- [5] Xiao-hui Liu and Chin-Seng Chua, "Multi-agent Activity Recognition Using Observation Decomposed Hidden Markov Model", in Proc. *International Con-*

ference on Computer Vision System, 2003, pp. 247-256.

Bibliography

- [1] “Mpeg-7, Tech. Rep. ISO/IEC JTC1/SC29/WG11/N4062, March 2001.
- [2] A. Agarwal and B. Triggs, “3d human pose from silhouettes by relevance vector regression,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, pp. 882–888.
- [3] J. Aggarwal and Q. Cai., “Human motion analysis: A review,” *Computer Vision and Image Understanding*, pp. 428–440, 1999.
- [4] E. Ardizzo and V. D. Gesú, “Content-based indexing of image and video databases by global and shape features,” in *Proc. IEEE International Conference on Pattern Recognition (ICPR’96)*, 1996, pp. 140–144.
- [5] J. Assfalg, M. Bertini, A. D. Bimbo, W. Nunziati, and P. Pala, “Soccer highlights detection and recognition using hmms,” in *Proc. IEEE International Conference on Multimedia and Expo (ICME’02)*, 2002, pp. 825–828.
- [6] D. Ayers and M. Shah, “Recognizing human actions in a static room,” in *Proc. 4th IEEE Workshop on Applications of Computer Vision (WACV’98)*, 1998, pp. 42–47.

-
- [7] N. Babaguchi, Y. Kawai, and T. Kitahashi, "Event based indexing of broadcasted sports video by intermodal collaboration," *IEEE Transactions on Multimedia*, pp. 68–75, Mar. 2002.
- [8] R. V. Babu, B. Anantharaman, K. R. Ramakrishnan, and S. H. Srinivasan, "Compressed domain action classification using hmm," *Pattern Recognition Letters*, pp. 1203–1213, 2002.
- [9] R. V. Babu and K. R. Ramakrishnan, "Recognition of human actions using motion history information extracted from the compressed video," *Image and Vision Computing*, pp. 597–607, 2004.
- [10] C. Barrett, R. Hughey, and K. Karplus, "Scoring hidden markov models," *Bioinformatics*, pp. 191–199, 1997.
- [11] J. L. Barronand, D. J. Fleet, S. Beauchemin, and T. Burkitt, "Performance of optical flow techniques," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1992, pp. 236–242.
- [12] B. Bauer, H. Hienz, and K. F. Kraiss, "Video-based continuous sign language recognition using statistical methods," in *Proc. 15th International Conference on Pattern Recognition*, 2000, pp. 463–466.
- [13] J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram, "Human activity recognition using multidimensional indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1091–1104, 2002.

-
- [14] C. BenAbdelkader, R. Cutler, and L. Davis, "Motion-based recognition of people in eigengait space," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 254–259.
- [15] A. G. Bharatkumar, K. E. Daigle, M. G. Pandey, Q. Cai, and J. K. Aggarwal, "Lower limb kinematics of human walking with the medial axis transformation," in *Proc IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, 1994, pp. 70–76.
- [16] A. Bobick, S. Intille, J. Davis, F. Baird, C. Pinhanez, L. Campbell, Y. Ivanov, A. Schutte, and A. Wilson, "The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment, Tech. Rep. 398, June 1998.
- [17] A. F. Bobick and A. Y. Johnson, "Gait recognition using static, activity-specific parameters," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, 2001, pp. 423–430.
- [18] A. F. Bobick and A. D. Wilson, "A state-based approach to the representation and recognition of gesture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1325–1337, 1997.
- [19] A. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 257–267, 2001.
- [20] J. S. Boreczky and L. D. Wilcox, "A hidden markov model framework for video segmentation using audio and image features," in *Proc. IEEE International*

- Conference on Acoustics, Speech, and Signal Processing (ICASSP'98)*, 1998, pp. 3741–3744.
- [21] J. E. Boyd and J. J. Little, “Global versus structured interpretation of motion: moving light displays,” in *Proc. Nonrigid and Articulated Motion Workshop*, 1997, pp. 18–25.
- [22] K. J. Bradshaw, I. D. Reid, and D. W. Murray, “The active recovery of 3-d motion trajectories and their use in prediction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 219–233, 1997.
- [23] M. Brand, “Shadow puppetry,” in *Proc. IEEE International Conference on Computer Vision*, 1999, pp. 1237–1244.
- [24] M. Brand and V. Kettner, “Discovery and segmentation of activities in video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 844–851, 2000.
- [25] C. Breglar, “Learning and recognizing human dynamics in video sequences,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 1997, pp. 568–574.
- [26] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick, and A. Pentland, “Invariant feature for 3-d gesture recognition,” in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition (ICAFGR'96)*, 1996, pp. 157–162.

-
- [27] C. Cedras and M. Shah, "Motion-based recognition: A survey," *Image and Vision Computing*, pp. 129–155, 1995.
 - [28] Z. Chen and H.-J. Lee, "Knowledge-guided visual perception of 3-d human gait from a single image sequence," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 336–342, 1992.
 - [29] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, "Semisupervised learning of classifiers: theory, algorithms, and their application to human-computer interaction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1553–1566, 2004.
 - [30] I. Cohen and H. Li, "Inference of human postures by classification of 3d human body shape," in *Proc. IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003, pp. 74–81.
 - [31] Collinsa, Lipton, Kanade, Fujiyoshi, Duggins, Tsin, Tolliver, Enomoto, and Hasegawa, "A system for video surveillance and monitoring: Vsam final report," Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-00-12, May 2000. [Online]. Available: <http://www-2.cs.cmu.edu/vsam/vsamhome.html>
 - [32] A. Corradini and H.-M. Gross, "Implementation and comparison of three architectures for gesture recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'00)*, 2000, pp. 2361–2364.

-
- [33] R. Cutler and L. S. Davis, “Robust real-time periodic motion detection, analysis, and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 781–796, 2000.
- [34] J. Cutting, D. Proffitt, and L. Kozlowski, “A biomechanical invariant for gait perception,” *Journal of Experimental Psychology: Human Perception and Performance*, pp. 357–372, 1978.
- [35] N. Dalai and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886 – 893.
- [36] J. W. Davis and A. Tyagi, “A reliable-inference framework for recognition of human actions,” in *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)’03*, 2003, pp. 169–176.
- [37] N. Dimitrova, L. Agnihotri, R. Jasinschi, J. Zimmerman, G. Marmaropoulos, T. McGee, and S. Dagtas, “Video scouting demonstration: smart content selection and recording,” in *ACM Multimedia*, 2000, pp. 499–500.
- [38] W. H. Dittrich, “Action categories and the perception of biological motion,” *Perception*, pp. 15–22, 1993.
- [39] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, “Recognizing action at a distance,” in *Proc. IEEE International Conference on Computer Vision (ICCV’03)*, 2003, pp. 726–733.

-
- [40] S. Eickeler and S. Muller, “Content-based video indexing of tv broadcast news using hidden markov models,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, vol. 6, 1999, pp. 2997–3000.
- [41] A. Elgammal and L. Chan-Su, “Inferring 3d body pose from silhouettes using activity manifold learning,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, pp. 681–688.
- [42] A. Elgammal, V. Shet, Y. Yacoob, and L. Davis, “Learning dynamics for exemplar-based gesture recognition,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*, 2003, pp. 571–578.
- [43] B. Erol and F. Kossentini, “Local motion descriptors,” in *Proc. IEEE Fourth Workshop on Multimedia Signal Processing*, 2001, pp. 467 – 472.
- [44] G. Fang and W. Gao;, “A srn/hmm system for signer-independent continuous sign language recognition,” in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 297–302.
- [45] G. Fang, W. Gao, and D. Zhao, “Large vocabulary sign language recognition based on fuzzy decision trees,” *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, pp. 305–314, 2004.

-
- [46] J. Fang and G. Qiu, “Human face detection using angular radial transform and support vector machines,” in *Proc. IEEE International Conference on Image Processing (ICIP’03)*, 2003, pp. 669–672.
- [47] P. F. Felzenszwalb, “Representation and detection of deformable shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 208–220, 2005.
- [48] P. F. Felzenszwalb and D. Huttenlocher, “Efficient matching of pictorial structures,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’00)*, 2000, pp. 13–15.
- [49] J. P. Foster, M. S. Nixon, and A. Prugel-Bennett, “Automatic gait recognition using area-based metrics,” *Pattern Recognition Letters*, pp. 2489–2497, 2003.
- [50] A. F. Frangi, W. J. Niessen, and M. A. Viergever, “Three-dimensional modeling for functional analysis of cardiac images: a review,” *IEEE Transactions on Medical Imaging*, pp. 2–25, 2001.
- [51] H. Fujiyoshi and A. Lipton, “Real-time human motion analysis by image skeletonization,” in *Proc. IEEE Workshop on Applications of Computer Vision (WACV’98)*, 1998, pp. 15–21.
- [52] A. Galata, N. Johnson, and D. Hogg, “Learning variable-length markov models of behavior,” *Computer Vision and Image Understanding*, pp. 398–413, 2001.

-
- [53] W. Gao, G. Fang, D. Zhao, and Y. Chen, “Transition movement models for large vocabulary continuous sign language recognition,” in *Proc. International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 553–558.
- [54] S. Gauch, L. Wei, and J. M. Gauch, “The vision digital video library,” *Information Processing and Management*, vol. 33, no. 4, pp. 413–426, 1997.
- [55] D. Gavrilu, “The visual analysis of human movement: A survey,” *Computer Vision and Image Understanding*, pp. 82–89, 1999.
- [56] S. Gong and T. Xiang, “Recognition of group activities using dynamic probabilistic networks,” in *Proc. IEEE International Conference on Computer Vision (ICCV)’03*, 2003, pp. 742 – 749.
- [57] H. Guo, Z.-Y. Ou, and Y. He, “Automatic fingerprint classification based on embedded hidden markov models,” in *Proc. International Conference on Machine Learning and Cybernetics*, 2003, pp. 3033–3038.
- [58] A. Hanjalic, “Shot-boundary detection: unraveled and resolved?” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 90 –105, 2002.
- [59] I. Haritaoglu, D. Harwood, and L. S. Davis, “W4: real-time surveillance of people and their activities,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 809 – 830, 2000.

-
- [60] J. B. Hayfron-Acquah, M. S. Nixon, and J. N. Carter, "Automatic gait recognition by symmetry analysis," *Pattern Recognition Letters*, pp. 2175–2183, 2003.
- [61] A. Higgins, L. Bahler, and J. Porter, "Speaker verification using randomized phrase prompting," *Digital Signal Processing*, pp. 89–106, 1991.
- [62] M. A. T. Ho, Y. Yamada, and Y. Umetani, "An hmm-based temporal difference learning with model-updating capability for visual tracking of human communicational behaviors," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 163–168.
- [63] S. Hongeng and R. Nevatia, "Multi-agent event recognition," in *Proc. IEEE International Conference on Computer Vision (ICCV)'01*, 2001, pp. 84–91.
- [64] M. Hoyneck and J. R. Ohm, "Shape retrieval with robustness against partial occlusion," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, 2003, pp. 593–596.
- [65] C. Hu, Q. Yu, Y. Li, and S. Ma, "Extraction of parametric human model for posture recognition using genetic algorithm," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 518–523.
- [66] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, pp. 334–352, 2004.

-
- [67] C.-L. Huang and B.-Y. Liao, "A robust scene-change detection method for video segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1281–1288, 2001.
- [68] P. S. Huang, "Automatic gait recognition via statistical approaches for extended template features," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, pp. 818–824, 2001.
- [69] P. S. Huang, C. J. Harris, and M. S. Nixon, "A statistical approach for recognizing humans by gait using spatial-temporal templates," in *Proc. International Conference on Image Processing (ICIP'98)*, 1998, pp. 178–182.
- [70] X. Huang, Y. Ariki, and M. A. Jack, *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.
- [71] J. P. Huelsenbeck and B. Rannala, "Phylogenetic methods come of age: Testing hypotheses in an evolutionary context," *Science*, pp. 227–232, 1997.
- [72] F. M. Idris and S. Panchanathan, "Spatio-temporal indexing of vector quantized video sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, no. 5, pp. 728–740, 1997.
- [73] S. Intille and A. Bobick, "Recognizing planned, multiperson action," *Computer Vision and Image Understanding*, pp. 415–445, 2001.

-
- [74] Y. Ivanov and A. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 852–872, 1999.
- [75] S. Iwasawa, K. Ebihara, J. Ohya, and S. Morishima, "Real-time estimation of human body posture from monocular thermal images," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 17–19.
- [76] M. Izumi and A. Kojiam, "Generating natural language description of human behavior from video images," in *Proc. IEEE International Conference on Pattern Recognition*, 2000, pp. 728–731.
- [77] A. Jepson, W. Richards, and D. Knill, *Perception as Bayesian Inference*. Cambridge Univ. Press, 1996, ch. 2, pp. 63–92.
- [78] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception and Psychophys*, pp. 201–211, 1973.
- [79] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image and Vision Computing*, vol. 14, pp. 609–615, 1996.
- [80] I. T. Jolliffe, *Principal Component Analysis*. Springer-Verlag, 1986.
- [81] A. Kale, A. K. R. Chowdhury, and Chellappa, "Towards a view invariant gait recognition algorithm," in *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance*, 2003, pp. 143–150.

-
- [82] A. Kale, A. Sundaresan, A. Rajagopalan, N. Cuntoor, A. RoyChowdhury, V. Kruger, and R. Chellappa, "Identification of humans using gait," *IEEE Transactions on Image Processing*, pp. 1163–1173, 2004.
- [83] U. Kersting, B. Rosenhahn, D. Smith, J. Gurney, T. Brox, and R. Klette, "Silhouette-based human motion estimation," in *Proc. International Conference on Biomechanics of the Lower Limb in Health Disease and Rehabilitation*, 2005, pp. 78–79.
- [84] J.-S. Kim, W. Jang, and Z. Bien, "A dynamic gesture recognition system for the korean sign language (ksl)," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, pp. 354–359, 1996.
- [85] J.-B. Kim, K.-H. Park, W.-C. Bang, and Z. Z. Bien, "Continuous gesture recognition system for korean sign language based on fuzzy logic and hidden markov model," in *Proc. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'02)*, 2002, pp. 1574–1579.
- [86] L. Kozlowski and J. Cutting, "Recognizing the sex of a walker from a dynamic point-light display," *Perception and Psychophysics*, pp. 575–580, 1977.
- [87] N. Krahnstover, M. Yeasin, and R. Sharma, "Towards a unified framework for tracking and analysis of human motion," in *Proc. IEEE Workshop on Detection and Recognition of Events in Video*, 2001, pp. 47–54.

-
- [88] A. Krogh, “Two methods for improving performance of an hmm and their application for gene finding,” in *Proc. International conference on Intelligent Systems for Molecular Biology*, 1997, pp. 179–186.
- [89] A. Kundu, H. Yang, and M.-Y. Chen, “Alternatives to variable duration hmm in handwriting recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1275–1280, 1998.
- [90] H. Lee and J. Kim, “An hmm-based threshold model approach for gesture recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 961–973, Oct. 1999.
- [91] K.-F. Lee, *Automatic Speech Recognition: The development of the SPHINX System*. Kluwer Academic Publishers, 1989.
- [92] L. Lee and W. E. L. Grimson, “Gait analysis for recognition and classification,” in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition (ICAFGR’02)*, 2002, pp. 148–155.
- [93] M. W. Lee and I. Cohen, “Human upper body pose estimation in static images,” in *Proc. European Conference On Computer Vision*, 2004, pp. 126–138.
- [94] M. K. Leung and Y.-H. Yang, “First sight: A human body outline labeling system,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 359–377, 1995.

-
- [95] J. J. Little and J. E. Boyd, “Describing motion for recognition,” in *Proc. IEEE International Symposium on Computer Vision*, 1995, pp. 235–240.
- [96] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proc. Image Understanding Workshop*, 1981, pp. 121–130.
- [97] J. Ma, W. Gao, J. Wu, and C. Wang, “A continuous chinese sign language recognition system,” in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 428–433.
- [98] A. Madabhushi and J. K. Aggarwal, “A bayesian approach to human activity recognition,” in *Proc. IEEE Workshop on Visual Surveillance (VS’99)*, 1999, pp. 25–32.
- [99] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Finger Print Recognition*. Springer, 2003.
- [100] J. N. Marcus, “A novel algorithm for hmm word spotting, performance evaluation and error analysis,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’92)*, 1992, pp. 89–92.
- [101] A. McIvor, Q. Zang, and R. Klette, “The background subtraction problem for video surveillance systems,” in *Proc. International Workshop on Robot Vision*, 2001, pp. 176 – 183.

-
- [102] K. Mikolajczyk, C. Schmid, and A. Zisserman, “Human detection based on a probabilistic assembly of robust part detectors,” in *Proc. European Conference On Computer Vision*, 2004, pp. 69–82.
- [103] T. Moeslund and E. Granum, “A survey of computer vision-based human motion capture,” *Computer Vision and Image Understanding*, pp. 231–268, 2001.
- [104] D. Moore, I. Essa, and M. Hayes, “Exploiting human actions and object context for recognition tasks,” in *Proc. IEEE International Conference on Computer Vision (ICCV)’99*, 1999, pp. 80–86.
- [105] G. Mori and J. Malik, “Estimating human body configurations using shape context matching,” in *Proc. European Conference on Computer Vision*, 2002, pp. 666–680.
- [106] G. Mori, X. Ren, A. A. Efros, and J. Malik, “Recovering human body configurations: combining segmentation and recognition,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’04)*, 2004, pp. 326–333.
- [107] R. J. Morris and D. C. Hogg, “Statistical models of object interaction,” *International Journal of Computer Vision*, vol. 37, no. 2, pp. 209–215, 2000.
- [108] H. Murase and R. Sakai, “Moving object recognition in eigenspace representation: gait analysis and lip reading,” *Pattern Recognition Letters*, pp. 155–162, 1996.

-
- [109] M. R. Naphade and T. S. Huang, “Discovering recurrent events in video using unsupervised methods,” in *Proc. IEEE International Conference on Image Processing (ICIP’02)*, 2002, pp. 13–16.
 - [110] M. Naylor and C. I. Attwood, “Annotated digital video for intelligent surveillance and optimised retrieval,” The ADVISOR Consortium, Tech. Rep. IST1999-11287 : ADVISOR, May 2003. [Online]. Available: <http://www-sop.inria.fr/orion/ADVISOR/finalreport.html>
 - [111] S. A. Niyogi and E. H. Adelson, “Analyzing and recognizing walking figures in xyt,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’94)*, 1994, pp. 469–474.
 - [112] A. Oh, H. Fox, M. V. Kleek, A. Adler, K. Gajos, L.-P. Morency, and T. Darrell, “Evaluating look-to-talk: A gaze-aware interface in a collaborative environment,” in *Proc. The CHI 2002 Conference on Human Factors in Computing Systems*, 2002.
 - [113] J. Ohya and F. Kishino, “Human posture estimation from multiple images using genetic algorithm,” in *Proc. IAPR International Conference on Pattern Recognition - Conference A: Computer Vision & Image Processing*, 1994, pp. 750–753.
 - [114] N. Oliver, B. Rosario, and A. Pentland, “A bayesian computer vision system for modeling human interactions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 831–843, 2001.

- [115] N. M. Oliver, B. Rosario, and A. P. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 831 – 843, 2000.
- [116] R.-H. . M. Ouhyoung;, "A real-time continuous gesture recognition system for sign language," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 558–567.
- [117] N. V. Patel and I. K. Sethi, "Video shot detection and characterization for video databases," *Pattern Recognition*, no. 4, pp. 583–592, 1997.
- [118] V. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677–695, 1997.
- [119] M. Petkovic, V. Mihajlovic, and W. Jonker, "Techniques for automatic video content derivation," in *Proc. IEEE International Conference on Image Processing (ICIP)'03*, 2003, pp. 611–614.
- [120] C. S. Pinhanez, "Ph.d thesis, representation and recognition of action in interactive spaces," Massachusetts Intritute of Technology," Thesis, June 1999.
- [121] C. S. Pinhanez and A. F. Bobick, "Human action detection using pnf propagation of temporal constraints," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, 1998, pp. 898 – 904.

-
- [122] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pp. 257–268, 1989.
- [123] C. Rao and M. Shah, "View-invariant representation and learning of human action," in *Proc. IEEE Workshop on Detection and Recognition of Events in Video*, 2001, pp. 55–63.
- [124] H. Ren and G. Xu, "Human action recognition in smart classroom," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 399–404.
- [125] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, pp. 19–41, 2000.
- [126] J. Ricard, D. Coeurjolly, and A. Baskurt, "Generalization of angular radial transform," in *Proc. IEEE International Conference on Image Processing*, 2004, pp. 24–27.
- [127] G. Rigoll, A. Kosmala, J. Rattland, and C. Neukirchen, "A comparison between continuous and discrete density hidden markov models for cursive handwriting recognition," in *Proc. the 13th International Conference on Pattern Recognition*, 1996, pp. 205–209.
- [128] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent word spotting," in *Proc.*

- IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'89)*, 1989, pp. 627–630.
- [129] R. C. Rose, “Discriminant wordspotting techniques for rejecting non-vocabulary utterances in unconstrained speech,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)'92*, 1992, pp. 105–108.
- [130] R. C. Rose and D. B. Paul, “A hidden markov model based keyword recognition system,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'90)*, 1990, pp. 129–132.
- [131] A. E. Rosenberg and S. Parthasarathy, “Speaker background models for connected digit password speaker verification,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'96)*, 1996, pp. 81–84.
- [132] B. Rosenhahn, C. Perwass, G., and Sommer, “Pose estimation of free-form contours,” *International Journal of Computer Vision*, pp. 267–289, 2005.
- [133] N. Rota and M. Thonnat, “Video sequence interpretation for visual surveillance,” in *Proc. IEEE International Workshop on Visual Surveillance*, 59–68, p. 2000.
- [134] Y. Rui and P. Anandan, “Segmenting visual actions based on spatio-temporal motion patterns,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, 2000, pp. 111–118.

-
- [135] J. M. S. Belongie and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 509 – 522, 2002.
- [136] H. Sagawa and M. Takeuchi, “A method for recognizing a sequence of sign language words represented in a japanese sign language sentence,” in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 434–439.
- [137] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, “The humanid gait challenge problem: data sets, performance, and analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 162–177, 2005.
- [138] A. Senior, “A hidden markov model fingerprint classifier,” in *Proc Asilomar Conference on Signals, Systems & Computers*, 1997, pp. 306–310.
- [139] G. Shakhnarovich, P. Viola, and T. Darrell, “Fast pose estimation with parameter sensitive hashing,” in *Proc. IEEE International Conference on Computer Vision*, 2003, pp. 750–757.
- [140] O. Siohan, C.-H. Lee, A. Surendran, and L. Qi, “Background model design for flexible and portable speaker verification systems,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’99)*, 1999, pp. 825 – 828.

-
- [141] C. G. M. Snoek and M. Worring, “Multimodal video indexing: A review of the state-of-the-art,” *Multimedia Tools and Applications*, pp. 5–35, 2005.
- [142] Y. Song, L. Goncalves, and P. Perona, “Unsupervised learning of human motion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 814–827, 2003.
- [143] E. L. L. Sonnhammer, S. R. Eddy, and R. Durbin, “Pfam: A comprehensive database of protein families based on seed alignments,” *Proteins*, pp. 405–420, 1997.
- [144] T. Starner, J. Weaver, and A. Pentland, “Real-time american sign language recognition using desk and wearable computer based video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1371 – 1375, 1998.
- [145] C. Stauffer and W. Grimson, “Learning patterns of activity using real-time tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 747–757, 2000.
- [146] R. Tanawongsuwan and A. Bobick, “Gait recognition from time-normalized joint-angle trajectories in the walking plane,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. 726–731.
- [147] W. Tavanapong and J. Zhou, “Shot clustering techniques for story browsing,” *IEEE Transactions on Multimedia*, pp. 517–527, 2004.

-
- [148] R. Tobias, T. Terasvirta, and S. Asbrink, “Stylized facts of daily return series and the hidden markov model,” *Journal of Applied Econometrics*, 1996.
- [149] A. S. Tolba, A. H. El-Baz, and A. A. El-Harby, “Face recognition: A literature review,” *International Journal of Signal Processing*, pp. 88–103, 2005.
- [150] N. Vasconcelos and A. Lippman, “A bayesian framework for semantic content characterization,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR)’98*, 1998, pp. 566–571.
- [151] R. C. Veltkamp and M. Tanase, “Content-based image retrieval systems: A survey,” Department of Computing Science, Utrecht University, Tech. Rep. UU-CS-2000-34, 2001.
- [152] P. Viola, M. J. Jones, , and D. Snow, “Detecting pedestrians using patterns of motion and appearance,” in *Proc. IEEE International Conference on Computer Vision*, 2003, pp. 734–741.
- [153] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. 511–518.
- [154] T. Wada and T. Matsuyama, “Multiobject behavior recognition by event driven selective attention method,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 873–887, 2000.

-
- [155] L. Wang, H. Ning, T. Tan, and W. Hu, "Fusion of static and dynamic body biometrics for gait recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 149–158, 2004.
- [156] L. Wang, T. Tan, W. Hu, and H. Ning, "Automatic gait recognition based on statistical shape analysis," *IEEE Transactions on Image Processing*, pp. 1120–1131, 2003.
- [157] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette analysis-based gait recognition for human identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1505–1518, 2003.
- [158] Z. Wang and J. Ben-Arie, "Optimal ramp edge detection using expansion matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 11, pp. 1092–1098, 1996.
- [159] J. A. Webb and J. K. Aggarwal, "Visually interpreting the motion of objects," *IEEE Comput.*, pp. 40–46, 1981.
- [160] L. D. Wilcox and M. A. Bush, "Training and search algorithms for an interactive wordspotting system," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'92)*, 1992, pp. 97 – 100.
- [161] J. Wilpon, L. Rabiner, C.-H. Lee, and E. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pp. 1870–1878, 1990.

-
- [162] A. Wilson and A. Bobick, "Parametric hidden markov models for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 884–899, 1999.
- [163] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 780 – 785, 1997.
- [164] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," in *Proc. IEEE International Conference on Computer Vision*, 2005, pp. 90–97.
- [165] Y., A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Computers*, pp. 84–94, 1980.
- [166] C. Y. Yam, M. S. Nixon, and J. N. Carter, "Gait recognition by walking and running: A model-based approach," in *Proc. Asian Conference on Computer Vision, (ACCV'02)*, 2002, pp. 1–6.
- [167] C. Yam, M. S. Nixon, and J. N. Carter, "On the relationship of human walking and running: automatic person identification by gait," in *Proc. International Conference on Pattern Recognition*, 2002, pp. 287–290.
- [168] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92)*, 1992, pp. 379–385.

-
- [169] J. Yang, Y. Xu, and C. S. Chen, “Human action learning via hidden markov model,” *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, pp. 34–44, 1997.
- [170] M. M. Yin and J. T. L. Wang, “Application of hidden markov models to gene prediction in dna,” in *Proc. International Conference on Information Intelligence and Systems*, 1999, pp. 40–47.
- [171] Z. Zang, “Iterative point matching for registration of free-form curves and surfaces,” *International Journal of Computer Vision*, pp. 119–152, 1999.
- [172] A. Zelinsky and J. Heinzmann, “Real-time visual recognition of facial gestures for human-computer interaction,” in *Proc. the Second International Conference on Automatic Face and Gesture Recognition*, 1996, pp. 351–356.
- [173] L. Zelnik-Manor and M. Irani, “Event-based analysis of video,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’01)*, 2001, pp. 123–130.
- [174] D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan, and G. Lathoud, “Modeling individual and group actions in meetings: A two-layer hmm framework,” in *Proc. Conference on Computer Vision and Pattern Recognition Workshop*, 2004, pp. 117–124.
- [175] J. Zhang, R. Collins, and Y. Liu, “Representation and matching of articulated shapes,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, pp. 342–349.

-
- [176] Y. J. Zhang;, “A review of recent evaluation methods for image segmentation,” in *Proc. Sixth International Symposium on Signal Processing and its Applications*, 2001, pp. 148–151.
 - [177] G. Zhao, R. Chen, G. Liu, and H. Li, “Amplitude spectrum-based gait recognition,” in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition (ICAFGR’04)*, 2004, pp. 23–28.
 - [178] D. Zhong and S.-F. Chang, “Spatio-temporal video search using the object based video representation,” in *Proc. IEEE International Conference on Image Processing (ICIP’97)*, 1997, pp. 21–24.
 - [179] D. Zhong and S.-F. Change, “Structure analysis of sports video using domain models,” in *Proc. IEEE International Conference on Multimedia and Expo (ICME)’01*, 2001, pp. 713–716.
 - [180] Z.Zivkovic, F. der Heijden, M.Petkovic, and W.Jonker, “Image processing and feature extraction for recognizing strokes in tennis game videos,” in *Proc. 7th Annual Conference of the Advanced School for Computing and Imaging (ASCI)’01*, 2001, pp. 262–266.