

國立交通大學

資訊工程學系

博士論文

利用 Isomap 學習及 VLMM 技術來分析人類之動作

A Study on Video-Based Human Action Analysis by Isomap

Learning and VLMM Techniques



研究生： 梁祐銘

指導教授： 廖弘源 教授

林正中 教授

中華民國 九十八 年 一 月

利用 Isomap 學習及 VLMM 技術來分析人類之動作

A Study on Video-Based Human Action Analysis by Isomap Learning and
VLMM Techniques

研究生：梁祐銘

Student：Yu-Ming Liang

指導教授：廖弘源

Advisor：Hong-Yuan Mark Liao

林正中

Cheng-Chung Lin



Submitted to Department of Computer Science

College of Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

January 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年一月

利用 Isomap 學習及 VLMM 技術來分析人類之動作

學生：梁祐銘

指導教授：廖弘源 博士
林正中 博士

國立交通大學資訊工程學系(研究所)博士班

摘要

人類動作分析是一個很基本的研究議題，並且被廣泛地應用在許多不同的研究領域。本論文提出了兩種適於人類動作分析之相關應用的視訊處理技術。首先，為了自動化分析一段冗長且尚未被切割過之人類動作視訊資料，我們提出一個以流形學習(manifold learning)技術為基礎之非監督式(unsupervised)人類動作分析架構。為了有效地分析人類動作，非監督式學習的方法比監督式學習的方法更為適合，主要是因為非監督式學習的方法事先不需要太多的人為介入。然而，複雜的人類動作使得非監督式學習的方法更具挑戰性。在這項研究中，我們首先從一個訓練用的動作序列中取得一個成對的人類姿勢距離矩陣。接著再利用等構映圖(Isomap)演算法從此矩陣中建構出低維度的結構。因此，訓練用的動作序列可以被映射到等構映圖空間中的流形軌跡(manifold trajectory)。為了有效地找出連續兩個單元(atomic)動作軌跡中的中斷點，我們將等構映圖空間中的流形軌跡描述為低維度的時間序

列。我們接著再利用時間分割的技術將此時間序列分割成許多次級序列，每個次級序列都代表一個單元動作。然後，我們利用動態時間校正(dynamic time warping)的技術來對這些單元動作序列分群。最後，我們依據分群結果來學習單元動作，並且再利用最近鄰算法(nearest neighbor rule)對單元動作做分類。假如介於輸入的動作序列與最相近的群平均單元動作序列的距離大於某個門檻值時，我們便將此輸入的動作序列視為未知的單元動作。

在第二項研究中，我們提出了一個利用可變長度馬可夫模型(variable-length Markov models)技術來學習及辨認單元人類動作的架構。本架構再包含兩個主要模組：姿勢標記模組及可變長度馬可夫模型之單元動作學習及辨認模組。首先，我們修改外形上下文(shape context)的技術來發展一個姿勢樣板(posture template)選擇的演算法。被選取的姿勢樣板可形成一個碼本(codebook)，利用此碼本我們可以將輸入的姿勢序列轉變為離散的符號序列。接著，我們利用可變長度馬可夫模型技術來學習對應於訓練用的單元動作之符號序列。最後，我們可將被建構的可變長度馬可夫模型轉換成隱藏式馬可夫模型(HMM)，並且再利用它來辨認輸入的單元動作。這項研究主要是結合可變長度馬可夫模型在學習方面的傑出好處及隱藏式馬可夫模型在容錯辨識能力的好處。

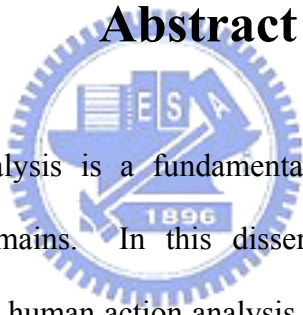
A Study on Video-Based Human Action Analysis by Isomap Learning and VLMM Techniques

Student : Yu-Ming Liang

Advisors : Dr. Hong-Yuan Mark Liao
Dr. Cheng-Chung Lin

Department of Computer Science
National Chiao Tung University

Abstract

The logo of National Chiao Tung University is a circular emblem. It features a gear-like outer border. Inside, there are stylized representations of a book, a building, and a graduation cap. The letters 'ES' are prominently displayed in the center, with 'A' to the right. Below these elements, the year '1896' is inscribed.

Human Action Analysis is a fundamental issue that can be applied to different application domains. In this dissertation, we propose two video processing techniques for human action analysis. First, to automatically analyze a long and unsegmented human action video sequence, we propose a framework for unsupervised analysis of human action based on manifold learning. To analyze of human action, unsupervised learning is superior to supervised one because the former does not require much human intervention beforehand. However, the complex nature of human action analysis makes unsupervised learning a challenging task. In this work, a pairwise human posture distance matrix is derived from a training action sequence. Then, the isometric feature mapping (Isomap) algorithm is applied to construct a low-dimensional structure from the distance matrix. Consequently, the training action sequence is mapped

into a manifold trajectory in the Isomap space. To identify the break points between any two successive atomic action trajectories, we represent the manifold trajectory in the Isomap space as a time series of low-dimensional points. A temporal segmentation technique is then applied to segment the time series into sub-series, each of which corresponds to an atomic action. Next, the dynamic time warping (DTW) approach is used to cluster atomic action sequences. Finally, we use the clustering results to learn and classify atomic actions according to the nearest neighbor rule. If the distance between the input sequence and the nearest mean sequence is greater than a threshold, it is regarded as an unknown atomic action.

In our second work, we propose a framework for learning and recognizing atomic human actions using variable-length Markov models (VLMMs). The framework comprises two modules: a posture labeling module, and a VLMM atomic action learning and recognition module. In the first stage, a posture template selection algorithm is developed based on a modified shape context matching technique. The selected posture templates form a codebook which can be used to convert input posture sequences into discrete symbol sequences for subsequent processing. Then, the VLMM technique is applied to learn the training symbol sequences of atomic actions. Finally, the constructed VLMMs are transformed into hidden Markov models (HMMs) for recognizing input atomic actions. This approach combines the advantages of the excellent learning function of a VLMM and the fault-tolerant recognition ability of an HMM.

誌謝

二十幾年來的求學過程，終於隨著本論文的完成而告結束。在博士求學期間，對我而言，是影響我人生最大的一段歷程。在這段過程中，遭遇了一些人生的大事，很慶幸有良師、益友陪我走過，讓我能堅持地走完這段漫長的求學過程。

首先，要感謝中研院資訊所的指導教授廖弘源老師，在我大四時帶領我進入這個領域，並且從碩士一路指導到現在完成博士學位。在這段日子裏，他不但教導我許多作研究的態度與方法，也教導了我許多做人做事的原則，並且給予我許多生活上的協助，真的很有福氣能當他的學生。其次，要感謝中研院資訊所施純傑老師，不管我在研究上或生活上遇到困難時，都可以提供我許多寶貴的建議與思考的方向，讓我獲益良多。另外，要感謝暨南大學資工系石勝文老師，在這段研究過程中，給了我許多研究上的建議，並且教導我論文寫作的技巧。此外，也要感謝校內的指導教授林正中老師，在博士求學過程中，給予我許多校內事務上的協助。因為有了這些老師的付出，這篇論文才得以在此呈現。

感謝佛光大學副校長范國清教授、台科大電子系方文賢教授、本系蔡文祥教授與莊仁輝教授，在百忙之中仍撥空指導並審查論文，實在惠我良多。同時也要感謝中研究資訊所多媒體技術實驗室的伙伴們：陳敦裕博士、蘇志文博士、林學億博士、曾易聰博士、孫士韋博士、張俊雄博士、賴育駿、羅

興源、凌誌鴻、林思瑤、林宏儒、施秀滿、洪亦雲，大家在研究室一起努力，使得研究的旅程不孤單。

最後，也是最重要的，要感謝我最親愛的家人，他們總是給予我最大的關心與支持，讓我能夠全心全力投入研究中，才造就了今日的我。

一聲聲的感謝，仍道不出我感恩之心，說不盡我感激之情，謹於此，獻上我最大的祝福，願大家都幸福美滿。



Table of Contents

Abstract in Chinese	i
Abstract in English	iii
Acknowledgements in Chinese	v
Table of Contents	vii
List of Tables	x
List of Figures	xi
1. Introduction	1
1.1 Motivation	1
1.2 Related Work	2
1.2.1 Survey on Atomic Action Segmentation	2
1.2.2 Survey on Atomic Action Recognition	4
1.3 Overview of the Proposed Methods	9
1.4 Dissertation Organization	11



2. Background Knowledge on Shape Context, Manifold Learning, and Variable-Length Markov Model	13
2.1 Shape Context	13
2.2 Manifold Learning	16
2.2.1 Isomap Algorithm	19
2.3 Variable-Length Markov Model	22
2.3.1 VLMM Learning	24
2.3.2 VLMM Recognition	27
3. Unsupervised Analysis of Human Action Based on Manifold Learning	29
3.1 Introduction	29
3.2 The Proposed Approach	32
3.2.1 Posture Representation and Matching	33
3.2.2 Isomap Learning of Human Action	37
3.2.3 Temporal Segmentation	38
3.2.4 Atomic Action Clustering	40
3.2.5 Atomic Action Learning and Classification	41
3.3 Experiments	45
3.4 Concluding Remarks	54

4. Learning Atomic Human Actions Using Variable-Length Markov Models	57
4.1 Introduction	57
4.2 The Proposed Method for Atomic Action Recognition	60
4.2.1 Posture Labeling	61
4.2.2 Human Action Sequence Learning and Recognition	64
4.3 Experiments	69
4.4 Concluding Remarks	81
5. Conclusions and Future Work	83
5.1 Conclusions	83
5.2 Future work	85
Bibliography	87
Publication List	99



List of Tables

4.1. The results of atomic action recognition using the training data	73
4.2. Comparison of our method's recognition rate with that of the HMM computed with the test data obtained from nine different human subjects	77
4.3. Comparison of our method's recognition rate with that of the AME plus NN method and the MMS plus NN method for the public database	80



List of Figures

2.1. Shape context computation and matching: (a) and (b) show the sampled points of two shapes; and (c)-(e) are the local shape contexts corresponding to different reference points. A diagram of the log-polar space is shown in (f), while (g) shows the correspondence between points computed using a bipartite graph matching method.	14
2.2. The 2-dimensional embedding manifolds of “Swiss Roll” computed with five different dimensionality reduction techniques: (a) Original 3-D data set, (b) PCA, (c) MDS, (d) Isomap, (e) LLE, and (f) LE.	18
2.3. An example of a VLMM	23
2.4. The PST for constructing the VLMM shown in Figure 2.3	24
3.1. The flowchart of the proposed method	33
3.2. Human action consists of a series of discrete human postures	34
3.3. Convex hull-shape contexts matching: (a) and (b) show the convex hull vertices of two shapes; (c) shows the correspondence between the convex hull vertices determined using shape matching.	36
3.4. The residual variance of Isomap on the training data	37

3.5. The constructed 4-D Isomap space: the manifold trajectory of the action sequence projected on to (a) the first three dimensions (dims. 1-3), and (b) the last three dimensions (dims. 2-4).	38
3.6. The time series of data points and corresponding magnitudes after Gaussian smoothing	39
3.7. The five classes of atomic actions used for training	46
3.8. The Isomap space constructed from the training data: the 4-D manifold trajectory projected on to (a) the first three dimensions (dims. 1-3), and (b) the last three dimensions (dims. 2-4).	46
3.9. The results of temporal segmentation of (a) the time series, and (b) the human posture sequence.	47
3.10. Five mean trajectories representing the five classes of atomic actions are plotted in different colors.	47
3.11. The atomic action trajectories constructed from test data sequence 1 and the classification results	49
3.12. The atomic action trajectories constructed from test data sequence 2 and the classification results	49
3.13. The atomic action trajectories constructed from test data sequence 3 superimposed on to the five learned exemplar trajectories.	50

3.14. The selected key data points and the reconstructed Isomap space	52
3.15. The average distance between the original Isomap and the reconstructed Isomap using different percentages of selected key points over the total number of data points	52
3.16. The reconstructed atomic action trajectories derived from test data sequence 1 and the classification results based on the simplified action classification approach.	53
3.17. The reconstructed atomic action trajectories derived from test data sequence 2 and the classification results based on the simplified action classification approach.	53
3.18. The reconstructed atomic action trajectories derived from test data sequence 3 based on the simplified action classification approach superimposed on to the five learned exemplar trajectories.	54
4.1. Block diagram of the proposed posture labeling process	62
4.2. (a) The VLMM constructed with the original input training sequence; (b) the original VLMM constructed with the preprocessed training sequence; (c) the modified VLMM, which includes the possibility of self-transition.	67
4.3. The distribution of observation error, obtained using the training data.	68
4.4. The ten categories of atomic actions used for training	70


4.5. Posture templates extracted from the training data	71
4.6. The histograms of the log-likelihood of the random sequences and the positive sequences for an action model	73
4.7. Nine test human subjects	76
4.8. Some typical postures of a human subject exercising action 1: (a) the input posture sequence; (b) the corresponding minimum-CSC-distance posture templates.	77
4.9. Recognition rates with respect to different τ_c	78
4.10. Sample images in the public action database	80



Chapter 1

Introduction

1.1 Motivation

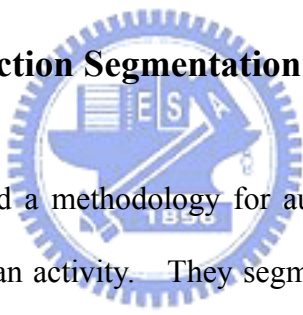


In recent years, visual analysis of human action has become a popular research topic in the field of computer vision. This is because it has a wide spectrum of potential applications, such as smart surveillance [14, 27], human computer interfaces [35, 55], content-based retrieval [39, 57], and virtual reality [21, 67]. Comprehensive surveys of related work can be found in [1, 23, 64]. In [64], Wang et al. pointed out that a human action analysis system needs to address two low-level processes, namely human detection and tracking, and a high-level process of understanding human action. While the low-level processes have been studied extensively, the high-level process has received relatively little attention. In this dissertation, we put our emphasis on video-based human action understanding.

1.2 Related Work

Human action usually consists of a series of atomic actions, each of which indicates a basic and complete movement. Therefore, understanding human action involves two key issues: 1) how to segment an input action sequence into atomic actions; and 2) how to recognize each segmented atomic action. Many approaches have been proposed for these two issues, which we describe in the following two subsections, respectively.

1.2.1 Survey on Atomic Action Segmentation



Ali and Aggarwal [2] proposed a methodology for automatic segmentation and recognition of continuous human activity. They segmented a continuous human activity into separate actions and correctly identified each action. First, they computed the angles subtended by three major components of the human body with the vertical axis, namely the torso, the upper component of the leg and the lower component of the leg. Then, they classified frames into breakpoint and non-breakpoint frames using these three angles as a feature vector. Breakpoints indicated an action's commencement or termination. Finally, each action between any two breakpoint frames was trained and classified using the corresponding sequence of feature vectors. In [68], a new method for temporal segmentation of human actions was proposed based on a 2D inter-frame similarity

plot. A similarity matrix involved relevant information for analysis of cyclic and symmetric human activities was used. The pattern associated to a periodic activity in the similarity matrix is rectangular and decomposable into elementary units. Thus, a morphology-based approach for the detection and analysis of activity patterns was proposed, and pattern extraction was then applied for the detection of the temporal boundaries of the cyclic symmetric activities. In [11], Chen et al. proposed a framework for automatic atomic human action segmentation in continuous action sequences. They used a star figure enclosed by a bounding convex polygon to effectively and uniquely represent the extremities of the silhouette of a human body. Thus, a sequence of the star figure's parameters was used to represent a human action. Then, they applied Gaussian mixture models (GMM) for human action modeling. Finally, they automatically segmented a sequence of continuous human actions using the underlying technique of the description model.

Cuntoor and Chellappa [17] proposed an antieigenvalue-based approach to detect key frames by investigating properties of operators that transformed past states to observed future states. The theory of antieigenvalues is based on changes in the data, and it is sensitive to how much a data vector is turned from a known direction, rather than the direction of persistence. On the other hand, eigenvectors represent the direction of maximum spread of the data and the eigenvalues are proportional to the amount of dilation. In [42], a method for segmentation and recognition of human body behavior data was proposed by

Nakata. He proposed a two-step scheme for human behavior recognition: analysis of movement correlations among limbs and temporal segmentation of motion data. Inter-limb movement correlations were widely observed in various behaviors and well represented contents of behavior, so it would be a universal feature value for general behavior. In general, the combination of inter-limb movements can be preserved until the action changes. Therefore, observing changes of inter-limb correlations, they segmented motion capture data into temporal fragment of action units. Hunter et al. [28] proposed a system to determine the segment boundaries in a broad range of actions and then to discriminate different action-types. They predicted sub-events using a set of basic movement features for a wide range of actions in which a human model interacted with objects. In addition, they created an accessible tool to track human actions for use in a wide range of machine vision and cognitive science applications.

1.2.2 Survey on Atomic Action Recognition

In the action recognition issue, existing methods can be categorized into two classes, i.e., 3-D based or 2-D based, depending on the type of human body model adopted [40].

3-D Based Methods

Ohya [45] proposed computer vision based methods for analyzing human behaviors: estimating postures and recognizing interactions between a human body and object. He developed a heuristic based method and a non-heuristic method for estimating postures in 3D from multiple camera images. The heuristic based method analyzes the contour of a human silhouette so that significant points of a human body can be located in each image, while the non-heuristic method utilizes a curve function for analyzing contours without using heuristic rules. Finally, they used the function-based contour analysis and motion vector-based analysis for recognizing the interactions so that the system could judge whether the human body interacted with the object. In [9], Boulay et al. presented a new approach for recognizing human postures in video sequences. They first used projections of moving pixels on a reference axis and learned 2-D posture appearances through PCA. Then, they employed a 3-D model of the posture to make the projection-based method independent of the camera's position.

Dockstader et al. [19] proposed a new model-based approach toward the 3-D tracking and extraction of gait and human motion. They suggested a structural model of the human body that leveraged the simplicity and robustness of a 3-D bounding volume and the elegance and accuracy of a highly parameterized stick model. The hierarchical structural model is accompanied by hard and soft

kinematic constraints. In [66], Werghi proposed a method for recognizing human body postures from 3D scanner data by adopting a model-based approach. To find a representation for a high power of discrimination between posture classes, he developed a new type of 3D shape descriptors, namely wavelet transform coefficients (WC). These features can be seen as an extension to 3D of 2D wavelet shape descriptors developed by [56]. Finally, he compared the WC with other 3D shape descriptors within a Bayesian classification framework.

2-D Based Methods

Using 3-D human body model, one can deal with more complex human actions. However, due to the need of developing low-cost systems, complex computations and expensive 3-D solutions are not considered for real-time applications. As a result, a number of researchers have proposed their analyses of human action based on 2-D postures. For examples, Haritaoglu et al. [26] proposed the W^4 system, a real time visual surveillance system for detecting and tracking multiple people and monitoring their activities in an outdoor environment. They computed the vertical and horizontal projections of a 2-D silhouette image to determine the global posture of a subject (standing, sitting, bending, or lying). In [8], Bobick and Davis proposed a new view-based approach for the representation and recognition of human movement. First, they stacked a set of consecutive frames to build a 2-D temporal template that characterizes human motion by using motion energy images (MEI) and motion history images (MHI). Moment-based

features were then extracted from the MEI and MHI and used for action recognition based on template matching.

Rahman and Ishikawa [50] proposed an automatic human action representation and recognition technique. In their scheme, a tuned eigenspace technique for automatic human posture and/or motion recognition that successfully overcome the appearance-change problem due to human wearing dresses and body shapes was proposed. In the first stage tuning, they employed image pre-processing by Gaussian and Sobel edge filter for reducing a dress effect. In the second stage tuning, they proposed a mean eigenspace produced by taking the mean of similar postures for avoiding the preceding problem. Finally, the obtained tuned eigenspace was used for recognition of unfamiliar postures and actions. In [37], Lv and Nevatia presented an example based single view action recognition system and demonstrated it on a challenging test set consisting of 15 action classes. They modeled each action as a series of synthetic 2D human pose rendered from a wide range of viewpoints. First, silhouette matching between the input frames and the key poses was performed using an enhanced Pyramid Match Kernel algorithm. And then, the best matched sequence of actions was tracked using the Viterbi algorithm. Li et al. [34] presented an automatic analysis of complex individual actions in diving video, and the aim was to provide biometric measurements and visual tools for coaching assistant and performance improving. They used 2D articulated human body model fitting and shape analysis techniques to obtain the main body joint angles of the athlete. Finally,

they presented two visual analyzing tools for individual sports game training: motion panorama and overlay composition.

In [38], Meng et al. proposed a human action recognition system for embedded computer vision applications. They addressed the limitations of the well known MHI and proposed a new hierarchical motion history histogram (HMHH) feature to represent the motion information. HMHH not only provides rich motion information, but also remains computationally inexpensive. Finally, they extracted a low dimension feature vector from the combination of MHI and HMHH and then used the feature vector for the support vector machine (SVM) classifiers. Hsieh et al. [27] presented a novel posture classification system for analyzing human movements directly from video sequences. In their schemes, each sequence of movements was converted into a posture sequence. They triangulated the posture into triangular meshes, and then extracted two features: the skeleton feature and the centroid context feature. The first feature was used as a coarse representation of the subject, while the second was used to derive a finer description. They generated a set of key postures from a movement sequence based on these two features such that the movement sequence was represented by a symbol string. Therefore, matching two arbitrary action sequences became a symbol string matching problem.

1.3 Overview of the Proposed Methods

Most of the approaches mentioned above are supervised learning-based. However, since the atomic actions are unknown beforehand, a large number of manually labeled training examples must be collected when using a supervised learning approach. Therefore, unsupervised learning approaches are always preferable for human action analysis. In this dissertation, we propose two video processing techniques for human action analysis. First, to automatically analyze a long and unsegmented human action sequence, we propose an unsupervised analysis of human action scheme based on manifold learning. Second, to learn segmented atomic action sequences, we propose a learning atomic human actions scheme using variable-length Markov models (VLMMs). A brief overview of the proposed methods is given as follows.

Unsupervised Analysis of Human Action Based on Manifold Learning

In this work, we propose a framework for unsupervised analysis of long and unsegmented human action sequences based on manifold learning. First, a pairwise human posture distance matrix, based on a modified shape context matching technique, is derived from a training action sequence. Then, the isometric feature mapping (Isomap) algorithm is applied to construct a low-dimensional structure from the distance matrix. Consequently, the training

action sequence is mapped into a manifold trajectory in the Isomap space. To identify the break points between any two successive atomic action trajectories, we represent the manifold trajectory in the Isomap space as a time series of low-dimensional points. A temporal segmentation technique is then applied to segment the time series into sub-series, each of which corresponds to an atomic action. Next, the dynamic time warping (DTW) approach is used to cluster atomic action sequences. Finally, we use the clustering results to learn and classify atomic actions according to the nearest neighbor rule. If the distance between the input sequence and the nearest mean sequence is greater than a threshold, it is regarded as an unknown atomic action.

Learning Atomic Human Actions Using Variable-Length Markov Models

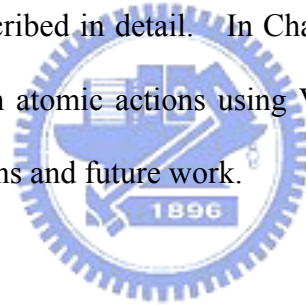


In this work, we propose a framework for learning and recognizing segmented atomic human action sequences using VLMMs. The framework is comprised of two modules: a posture labeling module, and a VLMM atomic action learning and recognition module. First, a posture template selection algorithm, based on the modified shape context matching technique, is developed. The selected posture templates form a codebook that is used to convert input posture sequences into discrete symbol sequences for subsequent processing. Then, the VLMM technique is applied to learn the training symbol sequences of atomic actions. Finally, the constructed VLMMs are transformed into hidden Markov models (HMMs) for recognizing input atomic actions. This approach combines the

advantages of the excellent learning function of a VLMM and the fault-tolerant recognition ability of an HMM.

1.4 Dissertation Organization

The remainder of this dissertation is organized as follows. In Chapter 2, we introduce the prerequisite materials of this dissertation, i.e., shape context, manifold learning, and variable-length Markov model. In Chapter 3, the proposed framework for unsupervised analysis of human action based on manifold learning is described in detail. In Chapter 4, we propose a framework for understanding human atomic actions using VLMMs. Finally, in Chapter 5, we present our conclusions and future work.





Chapter 2

Background Knowledge on Shape Context, Manifold Learning, and Variable-Length Markov Model



2.1 Shape Context

Shape context, proposed by Belongie et al. [5], is a shape descriptor, and it can be used for measuring shape similarity and recovering point correspondences. Therefore, shape context is usually applied to shape matching and object recognition. In the shape context theory, a shape is represented by a discrete set of sampled points, $P = \{p_1, p_2, \dots, p_n\}$. For each point $p_i \in P$, a coarse histogram h_i of the relative coordinates of the remaining $n-1$ points is computed as follows:

$$h_i(k) = \#\{p_j \neq p_i : (p_j - p_i) \in \text{bin}(k)\} \quad (2.1)$$

The histogram is defined to be the shape context of p_i . To make the descriptor more sensitive to positions of nearby sample points than to those of points farther away, the bins used in the histogram are uniform in a log-polar space. An example of shape context computation and matching is shown in Figure 2.1.

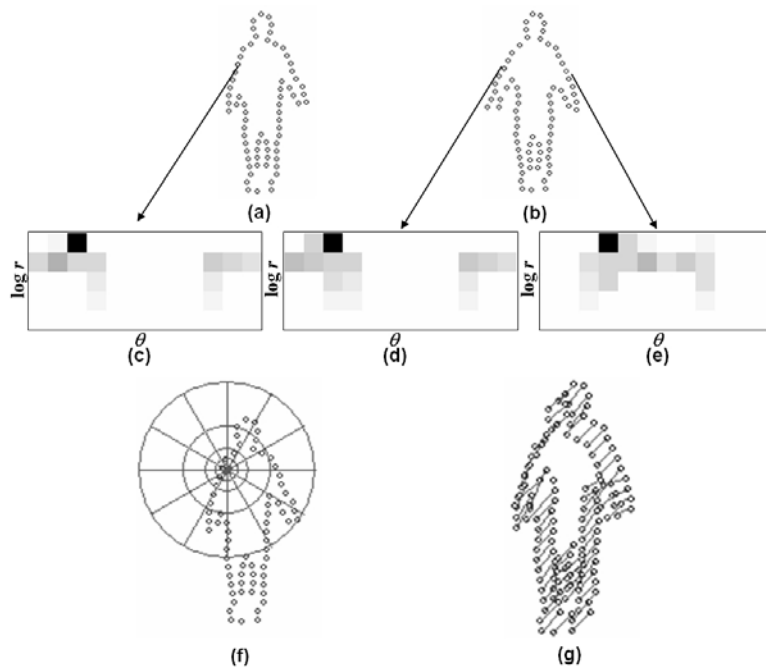


Figure 2.1. Shape context computation and matching: (a) and (b) show the sampled points of two shapes; and (c)-(e) are the local shape contexts corresponding to different reference points. A diagram of the log-polar space is shown in (f), while (g) shows the correspondence between points computed using a bipartite graph matching method.

Assume that p_i and q_j are points of the first and second shapes, respectively. The shape context approach defines the cost of matching the two points as follows:

$$C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}, \quad (2.2)$$

where $h_i(k)$ and $h_j(k)$ denote the K -bin normalized histograms of p_i and q_j , respectively. The cost $C(p_i, q_j)$ for matching points can include an additional term based on the local appearance similarity at points p_i and q_j . This is particularly useful when the shapes are derived from gray-level images instead of line drawings.

Given the set of costs $C(p_i, q_j)$ between all pairs of points p_i and q_j , shape matching is accomplished by minimizing the following total matching cost:

$$H(\pi) = \sum_i C(p_i, q_{\pi(i)}), \quad (2.3)$$

where π is a permutation of $1, 2, \dots, n$. Due to the constraint of one-to-one matching, shape matching can be considered as an assignment problem that can be solved by a bipartite graph matching method. A bipartite graph is a graph $G = (V = \{p_i\} \cup \{q_j\}, E)$, where $\{p_i\}$ and $\{q_j\}$ are two disjoint sets of vertices, and E is a set of edges connecting vertices from $\{p_i\}$ to $\{q_j\}$. The matching of a bipartite graph is to assign the edge connection. There are many matching

algorithms for bipartite graphs described in [4]. Here, the resulting correspondence points are denoted by $\{(p_i, q_{\pi(i)})\}, i = 1, 2, \dots, n\}$ or $\{(q_i, p_{\pi(i)})\}, i = 1, 2, \dots, m\}$, where n and m are the numbers of sample points on shapes P and Q , respectively. Therefore, the shape context distance between two shapes, P and Q , can be computed as follows:

$$D_{sc}(P, Q) = \frac{1}{n} \sum_i C(p_i, q_{\pi(i)}) + \frac{1}{m} \sum_j C(q_j, p_{\pi(j)}). \quad (2.4)$$

2.2 Manifold Learning

Manifold learning is a popular approach for nonlinear dimensionality reduction [36]. The purpose of dimensionality reduction is to map a high-dimensional data set into a low-dimensional space, while preserving most of the instinct structure in the data set. This is very important because many classifiers perform poorly in a high-dimensional space given a small number of training samples. Due to the prevalence of high-dimensional data, dimensionality reduction techniques have been popularly applied to many applications such as pattern recognition, data analysis, and machine learning. Most dimensionality reduction methods are linear, meaning that the extracted features are linear functions of the input features. Classical linear dimensionality reduction methods include the principal component analysis (PCA) [31, 60] and multidimensional scaling (MDS) [15]. Although the linear methods are easy to understand and are very simple to

implement, the linearity assumption does not lead to good results in many real-world applications. As a result, the design of nonlinear mapping methods is derived in a general setting.

Nonlinear mapping algorithms have been proposed recently based on the assumption that the data lie on a manifold. Thus, dimensionality reduction can be achieved by constructing a mapping that respects certain properties of the manifold. Popular manifold learning algorithms include the Isomap algorithm [58], the locally linear embedding (LLE) algorithm [54], and the Laplacian eigenmaps (LE) algorithm [6]. Each manifold learning algorithm tries to preserve a different geometrical property of the underlying manifold. Local approaches such as LLE and LE aim to preserve the proximity relationship among the data, while global approaches like Isomap aim to preserve the metrics at all scales. Thus, the global approaches give a more faithful embedding [36]. An example of dimensionality reduction is shown in Figure 2.2. Figure 2.2 (a) shows a 3-D data set, “Swiss Roll”, and the 2-D embedding manifolds recovered by using PCA, MDS, Isomap, LLE, and LE algorithms are shown in Figures 2.2 (b)-(f), respectively. Since we apply the Isomap algorithm to our first work, in what follows we introduce the Isomap algorithm in more details.

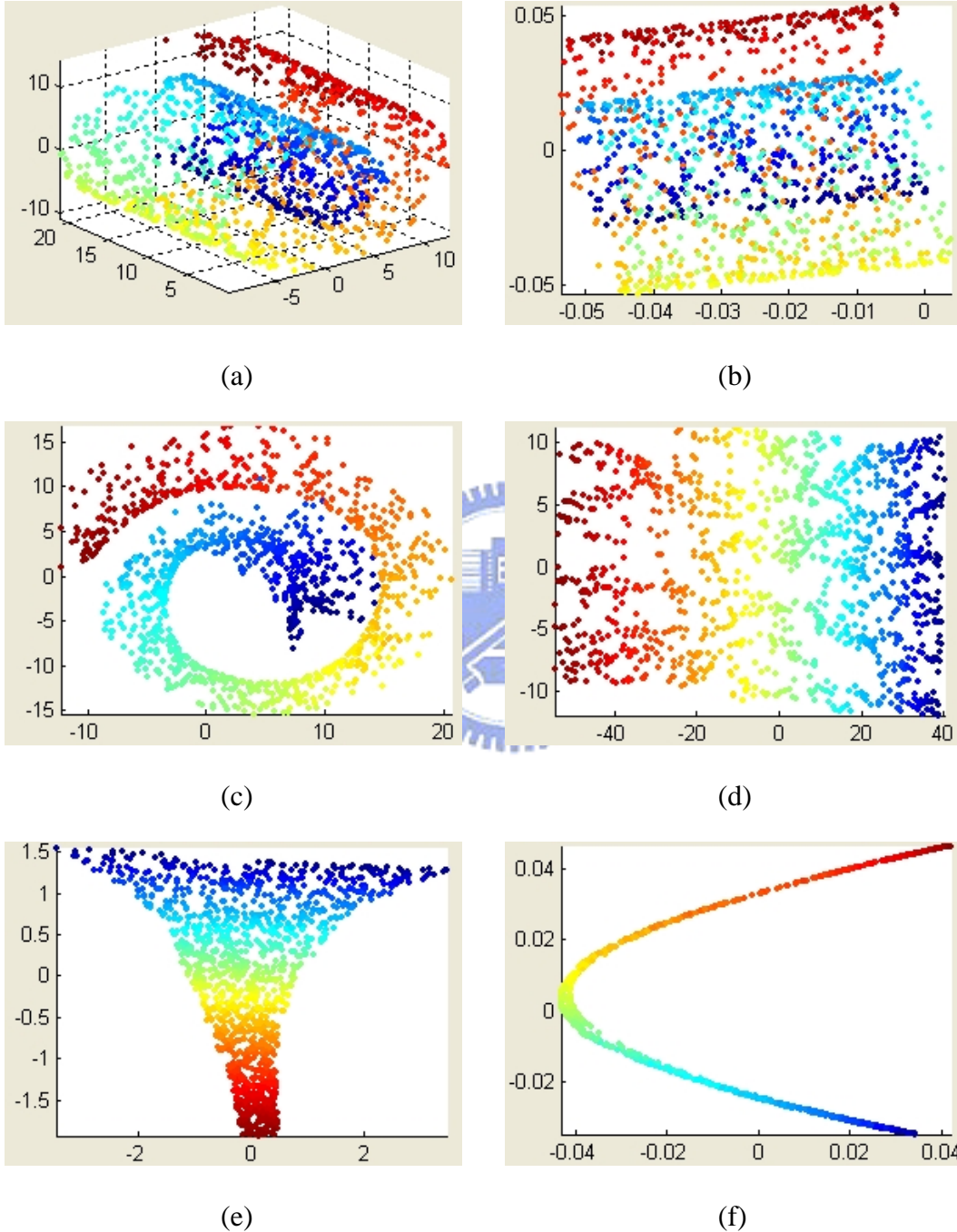


Figure 2.2. The 2-dimensional embedding manifolds of "Swiss Roll" computed with five different dimensionality reduction techniques: (a) Original 3-D data set, (b) PCA, (c) MDS, (d) Isomap, (e) LLE, and (f) LE.

2.2.1 Isomap Algorithm

The Isomap algorithm tries to find a low-dimensional Euclidean space that best preserves the geodesic distances between any two data points in the original high-dimensional space [58]. Since the manifold learning approach assumes that the data set have a low-dimensional structure, it is more appropriate to measure the distance between any two data points by their geodesic distance along the curve of the low-dimensional structure, rather than the Euclidean distance in the high-dimensional space. Therefore, the Isomap algorithm is to estimate the geodesic distances by the shortest paths in the neighborhood graph derived from connecting neighboring points. The algorithm comprises the following three steps:

1. *Construct neighborhood graph*: A weighted graph is constructed by connecting each point to its neighborhoods, and the weight of each edge is equal to the distance between the two points. The neighborhoods of each point can be determined using either the k nearest neighbor rule or points situated within a hyper-sphere of radius ϵ .
2. *Compute the pairwise geodesic distances*: The pairwise geodesic distance between any two nodes of the neighborhood graph is estimated by computing the shortest path between them on the graph.
3. *Construct a d -dimensional embedding*: The classic MDS algorithm [15] is applied to construct a d -dimensional embedding of the data.

Note that the difference between MDS and Isomap is that the Isomap uses the geodesic distance whereas MDS does not.

An important issue with the Isomap algorithm is how to determine the dimension d of the Isomap space. The residual variance, R_d , defined in the following equation is used to evaluate the error of dimensionality reduction

$$R_d = 1 - r^2(\mathbf{G}, \mathbf{D}_d), \quad (2.5)$$

where \mathbf{G} denotes the geodesic distance matrix; \mathbf{D}_d denotes the Euclidean distance matrix in the d -dimensional space; and $r(\mathbf{G}, \mathbf{D}_d)$ is the correlation coefficient of \mathbf{G} and \mathbf{D}_d . The value of d is determined using a trial and error approach to reduce the residual variance. Another important issue is how to construct a d -dimensional embedding of the data based on the MDS algorithm, in what follows we introduce the MDS algorithm in more detail.

Multidimensional Scaling

The objective of MDS [15] is to find the Euclidean distance reconstruction that best preserves the inter-point distances. Given a distance matrix $\mathbf{G} = [g_{ij}] \in \mathfrak{R}^{n \times n}$, where g_{ij} is the distance between points i and j , MDS constructs a set of n points in the d -dimensional Euclidean space such that inter-point distances are close to those in \mathbf{G} . Let $\mathbf{x}_i = (x_{i1}, \dots, x_{id})^T$ denote the

coordinates of the i th point in Isomap's Euclidean space. The Euclidean distance between the i th and j th points can be computed as follows:

$$d_{ij}^2 = (\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_i + \mathbf{x}_j^T \mathbf{x}_j - 2\mathbf{x}_i^T \mathbf{x}_j. \quad (2.6)$$

To overcome the indeterminacy of the solution due to arbitrary translation, the following zero-mean assumption is imposed:

$$\sum_{i=1}^n \mathbf{x}_i = \mathbf{0}. \quad (2.7)$$

From Equations (2.6) and (2.7), the inner product between \mathbf{x}_i and \mathbf{x}_j can be derived as follows:

$$b_{ij} = \mathbf{x}_i^T \mathbf{x}_j = -\frac{1}{2} \left(d_{ij}^2 - \frac{1}{n} \sum_{i=1}^n d_{ij}^2 - \frac{1}{n} \sum_{j=1}^n d_{ij}^2 + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 \right). \quad (2.8)$$

Let $\mathbf{D} = [d_{ij}]$ denote the distance matrix computed in the Isomap space. Since the Isomap space is determined such that \mathbf{D} is close to \mathbf{G} , the inner product matrix $\mathbf{B} = [b_{ij}]$ can be obtained by

$$\mathbf{B} = -\frac{1}{2} \mathbf{H} \mathbf{G} \mathbf{H}, \quad (2.9)$$

where $\mathbf{H} = \mathbf{I} - \frac{1}{n} \mathbf{1} \mathbf{1}^T$ is the centering matrix with $\mathbf{1} = [1, 1, \dots, 1]^T$, a vector of n ones. Let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T$ be the $n \times d$ matrix of the unknown coordinates of

the n points in the Isomap space. Then, the inner product matrix can be expressed as $\mathbf{B} = \mathbf{X}\mathbf{X}^T$. To compute \mathbf{X} from \mathbf{B} , we decompose \mathbf{B} into $\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$, is the diagonal matrix of eigenvalues and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$ is the matrix of corresponding eigenvectors.

The coordinate matrix \mathbf{X} can be calculated as follows:

$$\mathbf{X} = \mathbf{V}' \mathbf{\Lambda}^{\frac{1}{2}}, \quad (2.10)$$

where $\mathbf{\Lambda}^{\frac{1}{2}} = \text{diag}(\lambda_1^{\frac{1}{2}}, \lambda_2^{\frac{1}{2}}, \dots, \lambda_d^{\frac{1}{2}})$ and $\mathbf{V}' = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d]$.

2.3 Variable-Length Markov Model



A VLMM technique is usually applied to deal with a class of random processes in which the amount of memory is variable, in contrast to an n th-order Markov model for which the amount of memory is fixed. The advantage over a fixed memory Markov model is the ability to locally optimize the amount of memory required for prediction. Therefore, the VLMM technique is frequently applied to language modeling problems [25, 52] because of its powerful ability to encode temporal dependencies.

As shown in Figure 2.3, a VLMM can be regarded as a probabilistic finite state automaton (PFSA) $\Lambda = (S, V, \tau, \gamma, \pi)$ [52], where

- S denotes a finite set of model states, each of which is uniquely labeled by a

symbol string representing the memory of a conditional transition of the VLMM,

- V denotes a finite observation alphabet,
- $\tau: S \times V \rightarrow S$ is a state transition function such that $\tau(s_j, v) \rightarrow s_{j+1}$,
- $\gamma: S \times V \rightarrow [0,1]$ represents the output probability function with $\forall s \in S, \sum_{v \in V} \gamma(s, v) = 1$, and
- $\pi: S \rightarrow [0,1]$ is the probability function of the initial state satisfying $\sum_{s \in S} \pi(s) = 1$.

In the following subsections, we consider the VLMM learning in Section 2.3.1 followed by the VLMM recognition in Section 2.3.2.

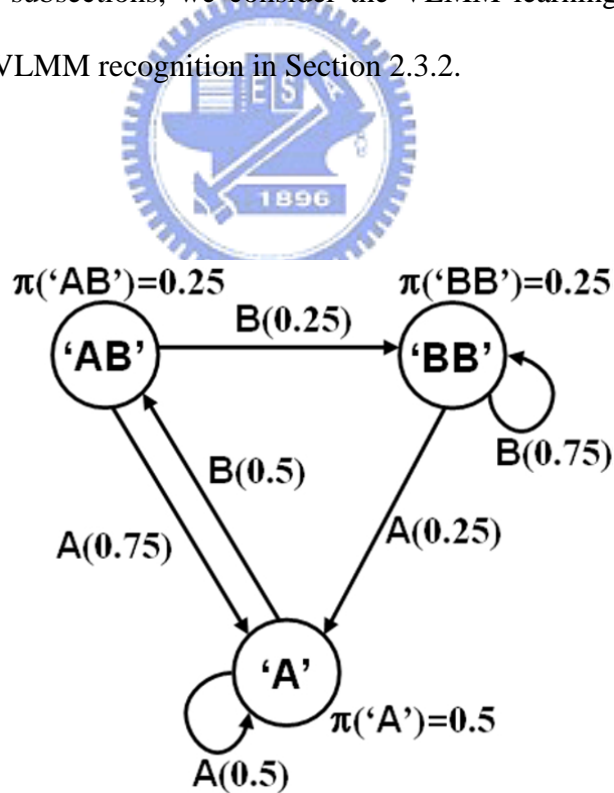


Figure 2.3. An example of a VLMM

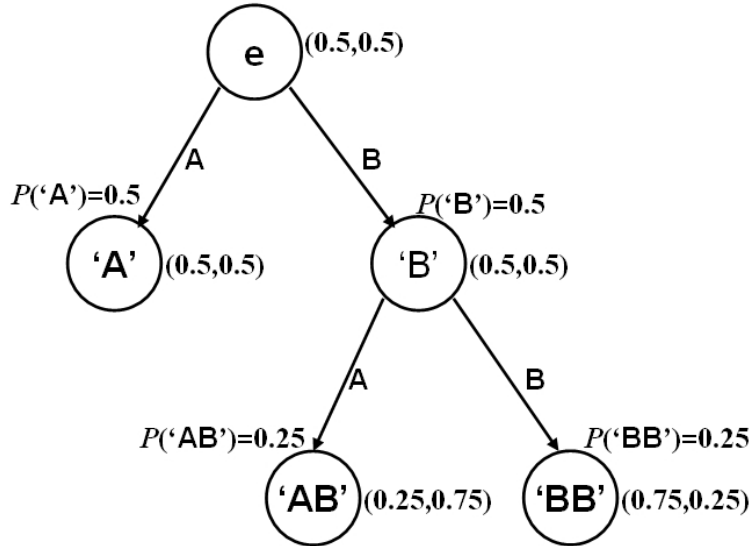


Figure 2.4. The PST for constructing the VLMM shown in Figure 2.3



2.3.1 VLMM Learning

The topology and the parameters of a VLMM can be learned from training sequences by optimizing the amount of memory required to predict the next symbol. Usually, the first step of training a VLMM involves constructing a prediction suffix tree (PST) [52]. A PST contains the information of the prefix of a symbol learned from the training data. Therefore, this prefix/suffix relationship helps to determine the amount of memory required to predict the next symbol. After the PST is constructed from the training sequences, the PST is converted to a PFSA representing the trained VLMM. Figure 2.4 depicts the PST constructed from a training sequence for converting the VLMM shown in

Figure 2.3.

Except for the root node, each node of the PST represents a non-empty symbol string, and each parent node represents the longest suffix tree of its child nodes. In addition, $P(v|s)$ is the output probability distribution of the next symbol v of each node s that satisfies $\sum_{v \in V} P(v|s) = 1$. The output and prior probabilities can be derived from the training symbol sequences as follows

$$P(v|s) = \frac{N(sv)}{N(s)}, \quad (2.11)$$

$$P(s) = \frac{N(s)}{N_0}, \quad (2.12)$$

where $N(s)$ is the number of occurrences of string s in the training symbol sequences, and N_0 denotes the size of the training symbol sequences.

To optimize the amount of memory required to predict the next symbol, it is necessary to determine when the PST growing process should be terminated. Assume that s is a node with the output probability $P(v|s)$, and $v's$ is its child node with the output probability $P(v|v's)$. We choose a termination criterion in order to avoid degrading the prediction performance of the reconstructed VLMM. Note that if the child node's output probability $P(v|v's)$ used to predict the next symbol, v , is significantly better than the output probability $P(v|s)$ of the parent node, the child node is a deemed better predictor than the parent node; therefore,

the PST should be grown to include the new child node. However, if the inclusion of a new child node does not improve the prediction performance significantly, the new child node should be discarded. Usually, the weighted Kullback-Leibler (KL) divergence [10] is applied to measure the statistical difference between the probabilities $P(v|v's)$ and $P(v|s)$ as follows:

$$\Delta H(v's, s) = P(v's) \sum_v P(v|v's) \log \frac{P(v|v's)}{P(v|s)}. \quad (2.13)$$

If $\Delta H(v's, s)$ is greater than a given threshold, the node $v's$ is added to the tree. In addition to the KL divergence criterion, a maximal-depth constraint of the PST is imposed to further limit the PST's size.

After the PST has been constructed, it must be transformed into a PFSA. First, the leaf nodes of the PST are defined as the states of the PFSA, and the latter's initial probability function is defined according to the probabilities of leaf nodes. Then, the transition function can be defined according to the symbol string combined from leaf nodes and their prediction symbols. The output probability function is defined based on the output probability distribution of the next symbol of each leaf node. Finally, the PFSA can be derived from the PST. For example, Figure 2.3 shows the PFSA derived from a PST shown in Figure 2.4.

2.3.2 VLMM Recognition

After a VLMM has been trained, it is used to predict the next input symbol according to a variable number of previously input symbols. In general, a VLMM decomposes the probability of a string of symbols, $O = o_1o_2\dots o_T$, into the product of conditional probabilities as follows:

$$P(O | \Lambda) = \prod_{j=1}^T P(o_j | o_{j-d_j} \dots o_{j-1}, \Lambda), \quad (2.14)$$

where o_j is the j -th symbol in the string and d_j is the amount of memory required to predict the symbol o_j .

The goal of VLMM recognition is to find the VLMM that best interprets the observed string of symbols, $O = o_1o_2\dots o_T$, in terms of the highest probability.

Therefore, the recognition result can be determined as model i^* as follows:

$$i^* = \arg \max_i P(O | \Lambda_i). \quad (2.15)$$



Chapter 3

Unsupervised Analysis of Human Action Based on Manifold Learning

In this chapter, we describe the proposed framework for unsupervised analysis of long and unsegmented human action sequences based on manifold learning. First, we give an introduction about this research topic. The proposed approach is then described. Next, we detail the experiment results. Finally, conclusions are given.

3.1 Introduction

In general, unsupervised learning is more difficult than supervised learning, so the number of published unsupervised learning methods is much smaller than that of supervised ones. Wang et al. [65] proposed an unsupervised approach for analyzing human gestures. They segmented the sequences of a human motion into atomic components and trained an HMM for each atomic component. Then,

they applied a hierarchical clustering approach to cluster the segmented components using the distances between the HMMs. Based on the clustering result, each atomic action can be converted into a discrete symbol. Finally, they extracted behavior lexicons from discrete symbols using the COMPRESSIVE algorithm [43]. Zhong et al. [72] proposed an unsupervised technique for detecting unusual events in a large video set. First, the features of each frame in the video set were extracted and classified into prototypes using the k-means algorithm. Second, the video sequences were divided into equal length segments. Third, a segment-prototype co-occurrence matrix was computed so that the segments could be clustered using the document-keyword clustering method proposed in [18]. Finally, unusual video segments were identified by finding clusters far away from the others. Turaga et al. proposed a vocabulary model for dynamic scenes and presented algorithms for unsupervised learning of the vocabulary from long video sequences [59]. They first segmented a video sequence into action elements, each of which was modeled as a linear time invariant (LTI) dynamical system. Next, they clustered those segments to discover distinct action elements using the distances between the LTI systems [13]. Then each segment was assigned a discrete symbol, and persistent activities in the symbol sequence were identified by using n-gram statistics.

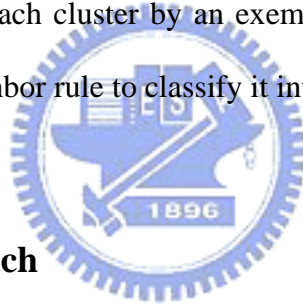
The above-mentioned approaches show that a general unsupervised system for human action analysis usually involves three stages: temporal segmentation, atomic action clustering, and atomic action learning and classification. Since the

human shape can be modeled as an articulated object with a high degree of freedom, the dimensions of a human shape descriptor are usually very large. Under these circumstances, the computation required for temporal segmentation, clustering and classification in a high dimensional feature space is not intuitive and may be very time consuming. Theoretically, a continuous human action sequence can be viewed as the variation of human postures lying on a low-dimensional manifold embedded in a high-dimensional space, which can be learned effectively from a set of training data [36]. In this chapter, we propose a framework for unsupervised analysis of human action based on manifold learning.

The goal of manifold learning, discussed in Section 2.2, is to discover a low-dimensional structure from a set of high-dimensional data. In recent years, some researchers have applied manifold learning algorithms to different tasks in the field of human action analysis, e.g., 3D body pose recovery [20], human tracking [41, 51], and human action recognition [12, 62]. The human action recognition methods proposed in [12, 62] are similar to the proposed approach, but they adopt a supervised learning method for human action recognition and they do not address the problems of temporal segmentation.

The main contribution of this study is that we propose a framework for unsupervised analysis of human action based on the Isomap algorithm. First, we propose a convex-hull shape contexts (CSC) descriptor to represent a human posture. Since the Isomap algorithm can preserve the CSC distance between any two postures of a training sequence and give a more faithful embedding,

mentioned in Section 2.2, we compute a CSC-based distance matrix and apply the Isomap algorithm to construct a low-dimensional structure from it. As a result, the training action sequence is mapped into a manifold trajectory in the Isomap space during the training process. To separate an action sequence into atomic actions precisely, the break points between any two consecutive atomic actions must be identified. To do this, we represent a manifold trajectory as a time series of low-dimensional points, and use a temporal segmentation technique to segment the manifold trajectory into atomic actions correctly. Next, we apply a DTW algorithm to perform atomic action sequence clustering. Finally, we use the clustered results to represent each cluster by an exemplar. For an input atomic action, we use the nearest neighbor rule to classify it into the correct category.



3.2 The Proposed Approach

Figure 3.1 shows the flowchart of the proposed method. The proposed approach comprises five stages: Posture representation and matching, Isomap learning of human action, temporal segmentation, atomic action clustering, and atomic action learning and classification, which we describe in the following four subsections, respectively.

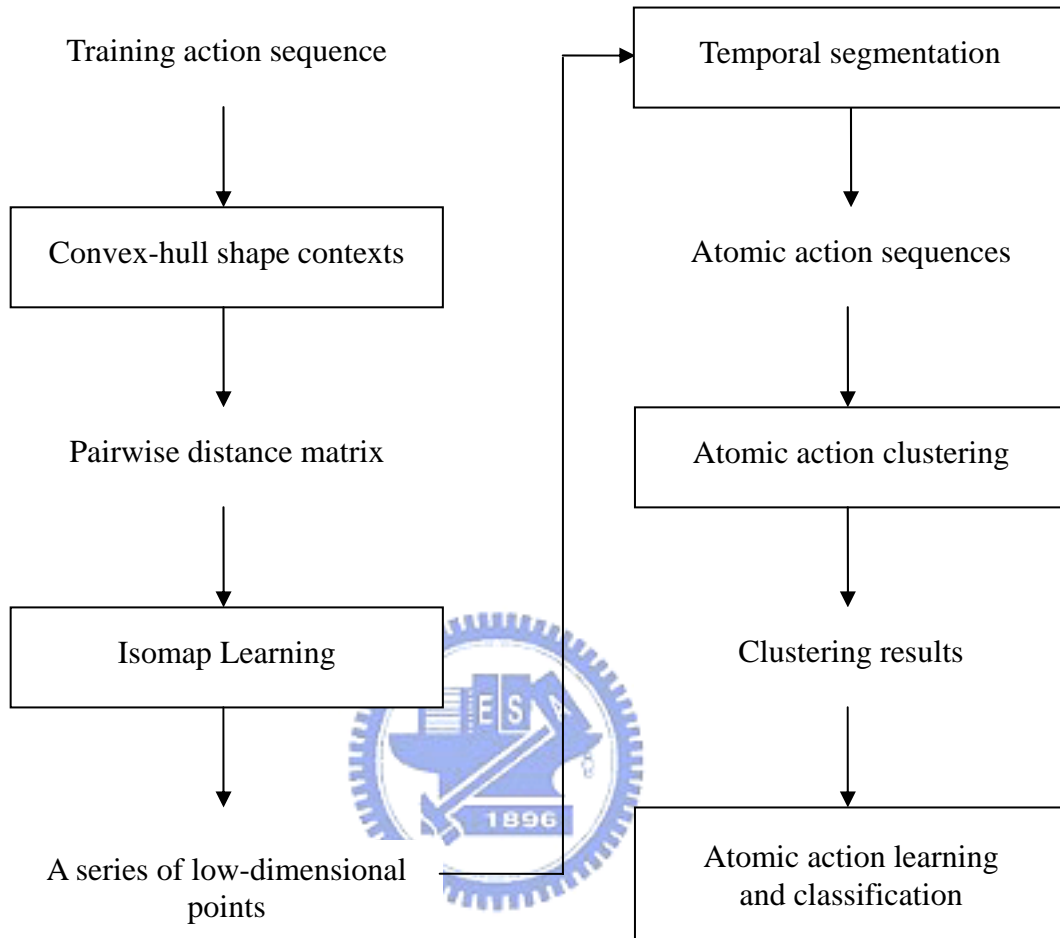


Figure 3.1. The flowchart of the proposed method

3.2.1 Posture Representation and Matching

Human action usually consists of a series of discrete human postures, as shown in Figure 3.2. Therefore, a human posture can be represented by a silhouette image, and a shape matching process can be used to assess the difference between two postures. For simplicity, it is assumed that the input video sequence has been

processed to obtain the human silhouette sequence, i.e., the action sequence. To construct a low-dimensional structure of human action from a training action sequence, the human posture must be represented effectively in the high-dimensional space. Therefore in this work, we modify the shape context technique, discussed in Section 2.1, to represent the human posture and deal with the posture matching problem. This modified method is aimed to improve the efficiency of posture matching with the prerequisite of not sacrificing too much the matching accuracy.

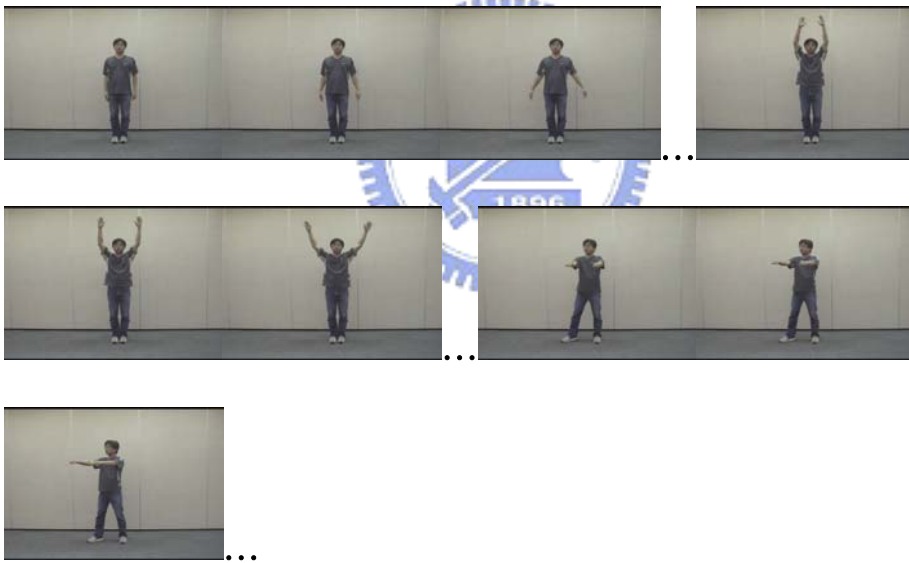


Figure 3.2. Human action consists of a series of discrete human postures

Although the shape context matching algorithm usually provides satisfactory results, the computational cost of applying it to a large database of human postures is so high that is not feasible. To reduce the computation time, we only compute the local shape contexts at certain critical reference points, which should

be easily and efficiently computable, robust against segmentation error, and critical to defining the shape of the silhouette. Note that the last requirement is very important because it helps preserve the informative local shape context. In this dissertation, the critical reference points are selected as the vertices of the convex hull of a human silhouette. Matching based on this modified shape context technique can be accomplished by minimizing a modified version of Equation (2.3) as follows:

$$H'(\pi) = \sum_{p \in A} C(p, q_{\pi(p)}), \quad (3.1)$$

where A is the set of convex hull vertices and H' is the adapted total matching cost. However, reducing the number of local shape contexts to be matched will also increase the influence of false matching results. To minimize the false matching rate, the ordering constraint of the vertices has to be imposed. However, since traditional bipartite graph matching algorithms [4] do not consider the order of all sample points, they are not suitable for our algorithm. Therefore, dynamic programming is adopted in the shape matching process. Suppose a shape P includes a set of convex hull vertices, A , and another shape Q includes a set of convex hull vertices, B . The CSC distance can be calculated as follows:

$$D_{\text{csc}}(P, Q) = \frac{1}{|A|} \sum_{p \in A} C(p, q_{\pi(p)}) + \frac{1}{|B|} \sum_{q \in B} C(q, p_{\pi(q)}). \quad (3.2)$$

An example of CSC matching is shown in Figure 3.3. There are three important reasons why convex-hull shape contexts can deal with the posture shape

matching problem effectively.

1. Since the number of convex hull vertices is significantly smaller than the number of whole shape points, the computation cost can be reduced substantially.
2. Convex hull vertices usually include the tips of human body parts; hence they can preserve more salient information about the human shape, as shown in Figure 3.2(a).
3. Even if some body parts are missed by human detection methods, the remaining convex hull vertices can still be applied to shape matching due to the robustness of computing the convex hull vertices, as shown in Figure 3.3.

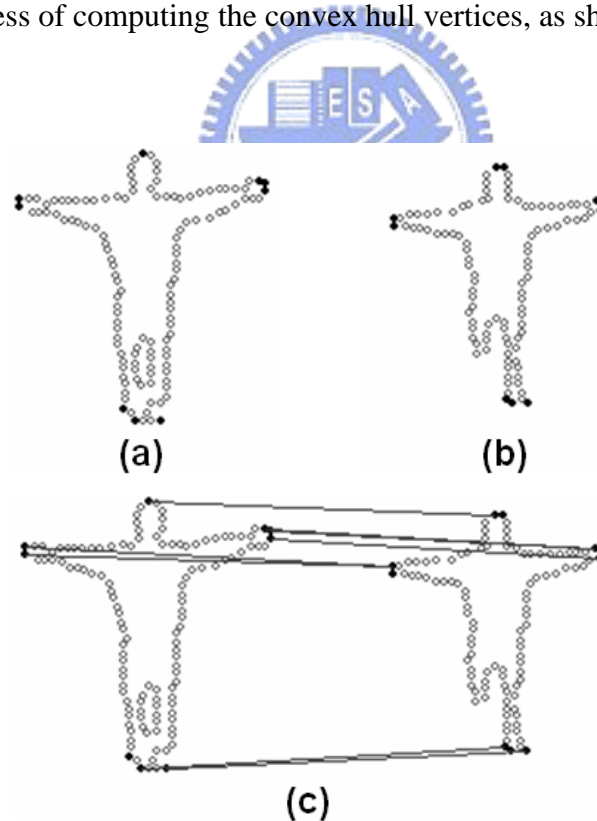


Figure 3.3. Convex hull-shape contexts matching: (a) and (b) show the convex hull vertices of two shapes; (c) shows the correspondence between the convex hull vertices determined using shape matching.

3.2.2 Isomap Learning of Human Action

When each silhouette in the training action sequence is represented as a CSC descriptor, a pairwise shape distance matrix can be calculated based on the shape matching. The computed distance matrix is used to construct an Isomap using the method described in Section 2.2.1. As a result, each human silhouette is transformed into a low-dimensional point in the Isomap space. Figure 3.4 shows the residual variance of the Isomap on the training data computed with different values of d , from which the number of dimensions of the Isomap space can be selected as four. Figure 3.5 shows the constructed 4-D Isomap space.

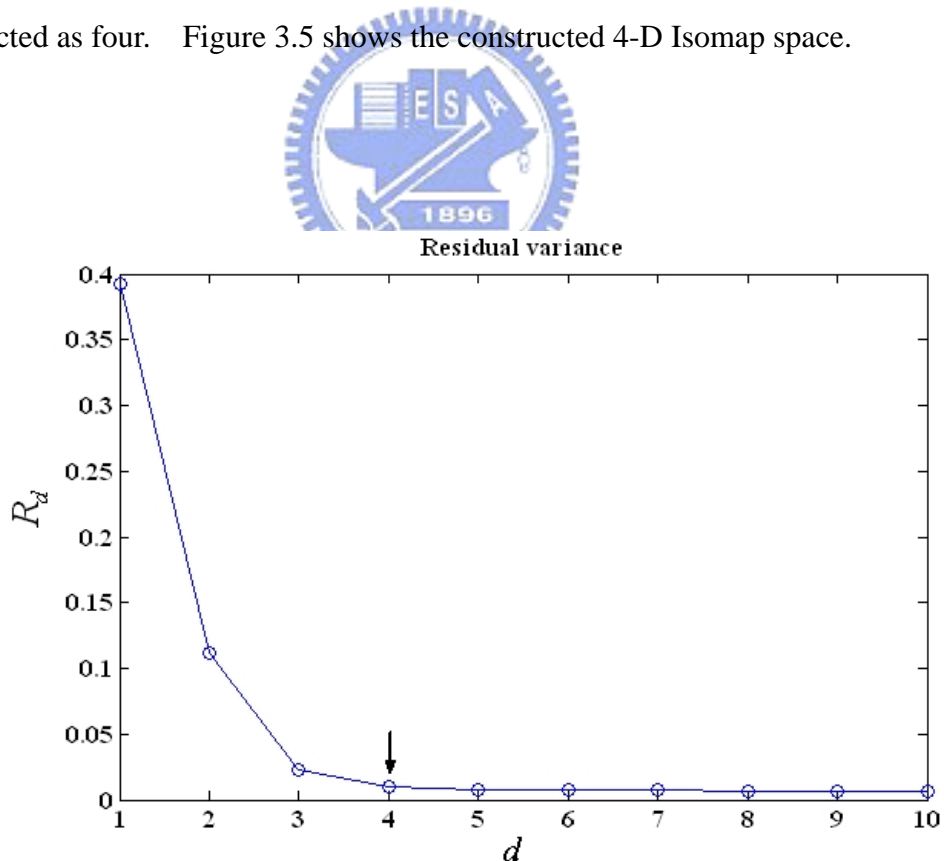


Figure 3.4. The residual variance of Isomap on the training data

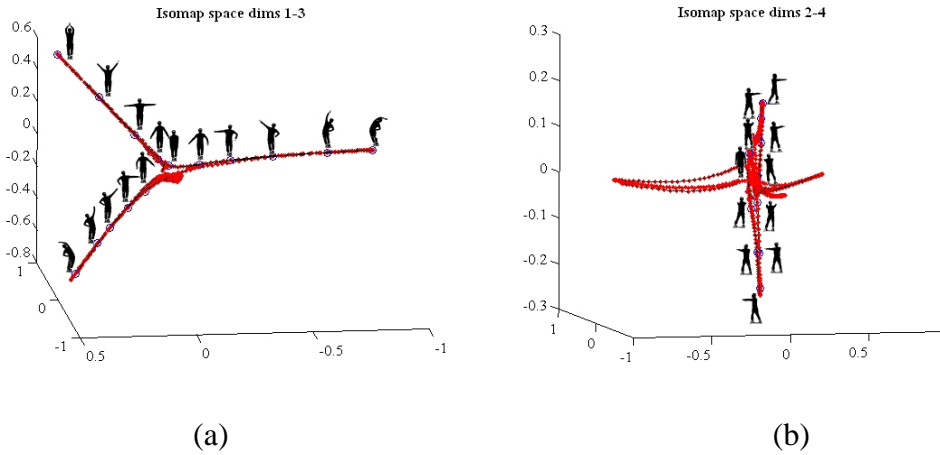


Figure 3.5. The constructed 4-D Isomap space: the manifold trajectory of the action sequence projected on to (a) the first three dimensions (dims. 1-3), and (b) the last three dimensions (dims. 2-4).

3.2.3 Temporal Segmentation

The purpose of temporal segmentation is to identify suitable break points to partition a continuous action sequence into atomic actions. In Figure 3.5, it is obvious that different atomic actions that can be distinguished using the CSC descriptors will have different trajectories. Therefore, the segmentation process involves identifying the break points between any two successive atomic action trajectories. To deal with this problem, we first represent the manifold trajectory as a time series of d -D data points and then calculate the magnitude (i.e., the two norm) of each point, as shown in Figure 3.6. In general, a human motion slows down at the boundary of an atomic action. Therefore, the local minima and the local maxima of the magnitude series can be regarded as candidate break points. Furthermore, since humans usually return to a rest posture after completing an

atomic action, we define the points that indicate low-speed actions and the postures adjacent to the rest posture as the break points of atomic actions. Note that, since rest postures appear in nearly almost all atomic actions, they are usually the most common postures mapped in the neighborhood of the origin of the Isomap space due to the zero-mean assumption formulated in Equation (2.7). Therefore, we only use the local minima as break points to derive atomic action sequences. In the magnitude series shown in Figure 3.6, there are eleven local minima, which divide the action trajectory into ten atomic actions.

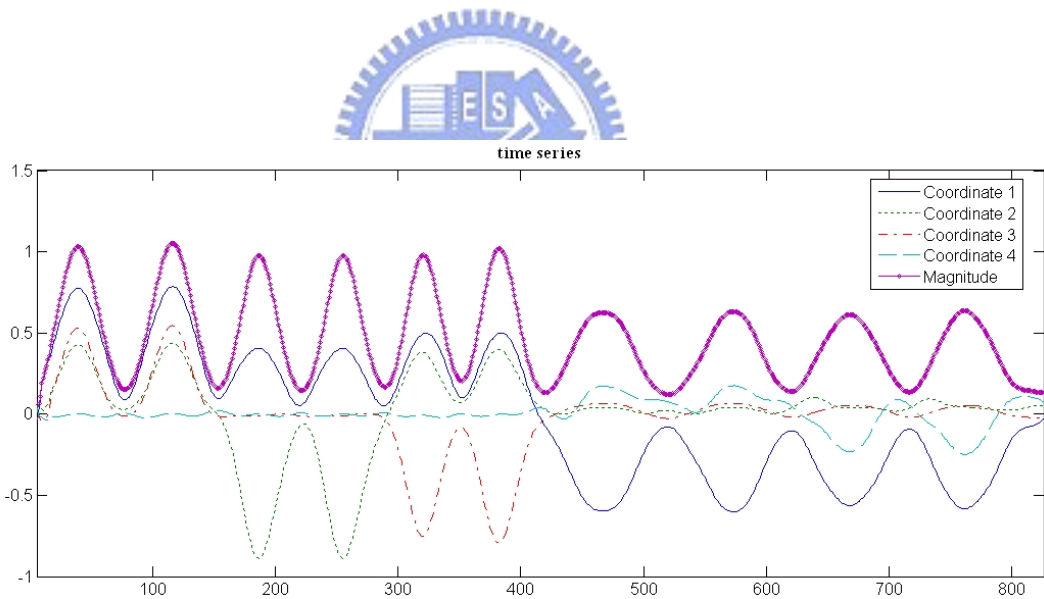


Figure 3.6. The time series of data points and corresponding magnitudes after Gaussian smoothing

3.2.4 Atomic Action Clustering

After segmenting the training action sequence, the segmented actions are clustered to identify and model each atomic action. Since the duration of each segmented action sequence is different, the DTW algorithm [48] is used to cluster the segmented action sequences. DTW aligns and compares two sequences by finding an optimal warping path between them. For example, suppose we have two sequences: $A = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m)$ and $B = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$, where \mathbf{a}_i and \mathbf{b}_j are d -D vectors in the Isomap space. A warping path $W = (w(1), w(2), \dots, w(m))$ is used to align A with B , where $j = w(i)$ means that \mathbf{a}_i is aligned with \mathbf{b}_j . The warping path is computed with the following three constraints: $1 = w(1)$, $n = w(m)$, and $w(k+1) \geq w(k)$. The distance between A and B along the warping path W can then be calculated as follows:

$$D_W(A, B) = \frac{1}{m} \sum_{i=1}^m \|\mathbf{a}_i - \mathbf{b}_{w(i)}\|. \quad (3.3)$$

The objective of the DTW is to find the warping path that minimizes the distance $D_W(A, B)$. Therefore, the DTW matching score between A and B can be calculated by

$$DTW(A, B) = \min_W \{D_W(A, B)\}. \quad (3.4)$$

Since the definition of DTW matching score is not symmetric, we define the DTW

action distance between two segmented action sequences by

$$AD(A, B) = \frac{1}{2}(DTW(A, B) + DTW(B, A)). \quad (3.5)$$

After calculating the pairwise action distances between all segmented actions, we group similar atomic actions into one cluster using the hierarchical clustering algorithm [29].

3.2.5 Atomic Action Learning and Classification

In this step, the mean trajectory of each cluster is used as an exemplar to represent the cluster. The time warping paths computed by DTW are used to normalize the duration of each segmented action sequence in a cluster in order to calculate the mean trajectory. Meanwhile, the exemplars of the atomic actions are utilized to classify a new input action based on the nearest-neighbor approach using the DTW distance defined in Equation (3.5). To recognize a new input action sequence, we need to map the new sequence into the Isomap space. Since the Isomap space is only constructed for the training data, to project new test points into the Isomap space, both the neighborhood graph and the geodesic distance must both be recomputed. Then, the MDS approach can be applied to generate a new Isomap space. However, reconstructing the Isomap with new data is very time consuming, especially when the size of the new input sequence is large. To resolve this problem, Law and Jain [33] proposed an incremental Isomap learning

method, which avoids spending time on “batch” Isomap construction. Specifically, the method only updates the neighbor graph and the geodesic distance for partial points related to the new sample to identify the coordinates of that sample, after which it updates the coordinates of all the points. Although the approach can reduce a certain amount of computation time, it is still a time-consuming process. In this work, we propose another way to deal with the problem. We find that it is not necessary to build a new Isomap space for atomic action recognition unless a new action model is added. Therefore, as proposed in [33], we simply estimate the coordinates of a new sample to project the new data points into the existing Isomap space.

Suppose that the coordinates of the training data points are $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ and the coordinate vector of a new data point is denoted by \mathbf{x}_{n+1} . Then, the summation over $i = 1, \dots, n+1$ and $j = 1, \dots, n+1$ for d_{ij} in Equation (2.6) leads to

$$\mathbf{x}_{n+1}^T \mathbf{x}_{n+1} = \frac{1}{n} \sum_{i=1}^n d_{i(n+1)}^2 - \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^T \mathbf{x}_i. \quad (3.6)$$

The inner product of \mathbf{x}_{n+1} and \mathbf{x}_j can also be calculated using Equation (2.6) as follows:

$$\mathbf{x}_{n+1}^T \mathbf{x}_j = -\frac{1}{2} (d_{(n+1)j}^2 - \mathbf{x}_{n+1}^T \mathbf{x}_{n+1} - \mathbf{x}_j^T \mathbf{x}_j). \quad (3.7)$$

Then, from Equations (2.6), (3.6) and (3.7), the value of $\mathbf{x}_{n+1}^T \mathbf{x}_j$ can be calculated

by

$$\mathbf{x}_{n+1}^T \mathbf{x}_j = -\frac{1}{2}(d_{(n+1)j}^2 - \frac{1}{n} \sum_{i=1}^n d_{i(n+1)}^2 - \frac{1}{n} \sum_{i=1}^n d_{ij}^2 + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2) = f_j. \quad (3.8)$$

Therefore,

$$[\mathbf{x}_1, \dots, \mathbf{x}_n]^T \mathbf{x}_{n+1} = \mathbf{f} = [f_1, \dots, f_n]^T. \quad (3.9)$$

Substituting Equation (2.10) into Equation (3.9), we have

$$(\sqrt{\lambda_1} \mathbf{v}_1, \dots, \sqrt{\lambda_d} \mathbf{v}_d)^T \mathbf{x}_{n+1} = \mathbf{f}. \quad (3.10)$$

Then, the least-square solution of \mathbf{x}_{n+1} can be derived as follows:

$$\mathbf{x}_{n+1} = \left(\frac{1}{\sqrt{\lambda_1}} \mathbf{v}_1^T \mathbf{f}, \dots, \frac{1}{\sqrt{\lambda_d}} \mathbf{v}_d^T \mathbf{f} \right)^T. \quad (3.11)$$

After each human posture in the test action sequence has been projected into the Isomap space, the test sequence can be classified into a cluster using the nearest-neighbor approach based on the DTW distance. We use a threshold to judge whether an unknown action sequence belongs to one of the learned clusters. If the action distance between the unknown action sequence and the nearest mean sequence is greater than the threshold value, the unknown action is regarded as a new action. The threshold is set assigned as $\mu_k + 2\sigma_k$, where μ_k and σ_k are, respectively, the mean and the standard deviation of the distances between the

training action sequences and the exemplar of the k th cluster.

Although not building a new Isomap for atomic action recognition saves a great deal of computation time, by Equation (3.8), we still have to calculate the CSC distances between a new human posture and all of the training postures. However, computing the CSC distances is inefficient when the number of training postures is large. To resolve this problem, we propose the following accelerated mapping approach. First, based on the k-means algorithm, we replace the training postures with a set of key data points selected from the training set in the Isomap space. Then, based on Equations (3.6)-(3.9), we use the key data points to estimate the coordinate vector of a new data point. Because the equations are derived based on the zero-mean assumption about training data points, we have to translate the coordinates of the key data points as follows:

$$\mathbf{x}'_{a_i} = \mathbf{x}_{a_i} - \bar{\mathbf{x}}_a, \quad (3.12)$$

where $\mathbf{x}_{a_i}, i=1, \dots, k$ are the key data points and $\bar{\mathbf{x}}_a = \frac{1}{k} \sum_{i=1}^k \mathbf{x}_{a_i}$. Therefore, the inner product of \mathbf{x}'_{n+1} and all of the key data points can be calculated by applying Equation (3.9):

$$\left[\mathbf{x}'_{a_1}, \dots, \mathbf{x}'_{a_k} \right]^T \mathbf{x}'_{n+1} = \mathbf{X}'_a{}^T \mathbf{x}'_{n+1} = \mathbf{f}'_a = \left[f'_{a_1}, \dots, f'_{a_k} \right]^T. \quad (3.13)$$

Then, the least-square solution of \mathbf{x}'_{n+1} can be derived as follows:

$$\mathbf{x}'_{n+1} = (\mathbf{X}_a^T \mathbf{X}_a)^{-1} \mathbf{X}_a^T \mathbf{f}_a. \quad (3.14)$$

Finally, the coordinate vector of a new data point \mathbf{x}_{n+1} can be calculated by $\mathbf{x}'_{n+1} + \bar{\mathbf{x}}_a$. Note that $(\mathbf{X}_a^T \mathbf{X}_a)^{-1} \mathbf{X}_a^T$ in Equation (3.14) can be computed beforehand; thus, estimating the coordinates of a new posture in the embedded Euclidian space is very efficient.

3.3 Experiments

We conducted a series of experiments to evaluate the effectiveness of the proposed method. The data used in the experiments included one training sequence and three test sequences performed by two human subjects. The training data contained 25 atomic action sequences comprised of 1983 frames that belonged to five different classes of actions performed by subject 1. Some typical image frames of each atomic action class are shown in Figure 3.7. Using the Isomap algorithm, a 4-D Isomap space was constructed from the training data, as shown in Figure 3.8. Next, we represented the manifold trajectory in Figure 3.8 as a time series of data points, and then applied temporal segmentation to the time series. The results of temporal segmentation are shown in Figure 3.9. Using atomic action clustering, the segmented atomic actions were correctly grouped into five clusters, and five exemplar mean trajectories (see Figure 3.10) were computed to represent the obtained clusters.






Action class 1	
Action class 2	
Action class 3	
Action class 4	
Action class 5	

Figure 3.7. The five classes of atomic actions used for training

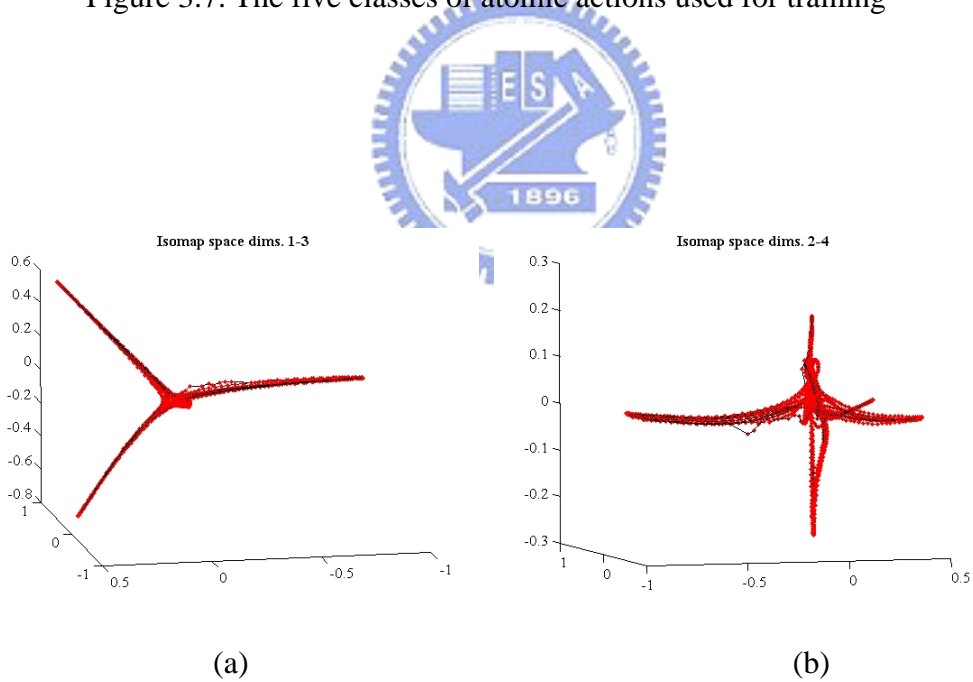


Figure 3.8. The Isomap space constructed from the training data: the 4-D manifold trajectory projected on to (a) the first three dimensions (dims. 1-3), and (b) the last three dimensions (dims. 2-4).

3.3 Experiments

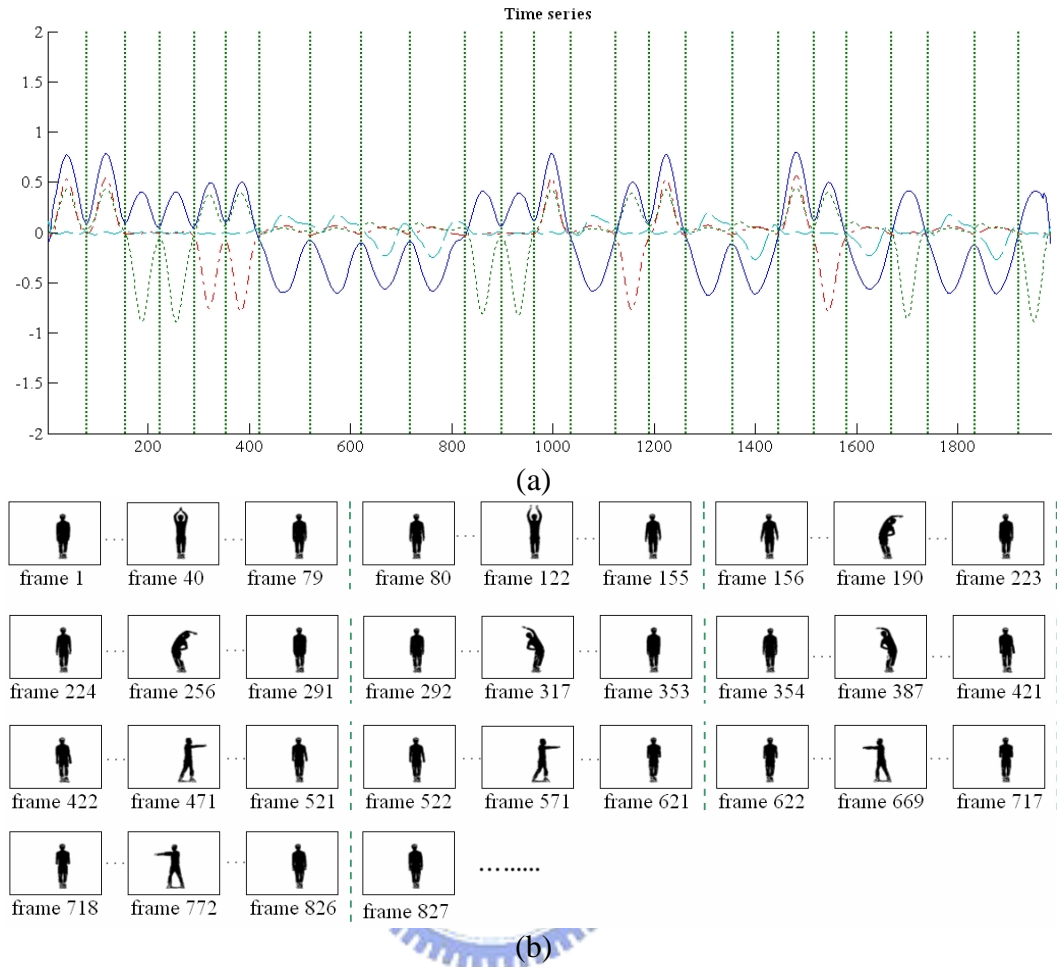


Figure 3.9. The results of temporal segmentation of (a) the time series, and (b) the human posture sequence

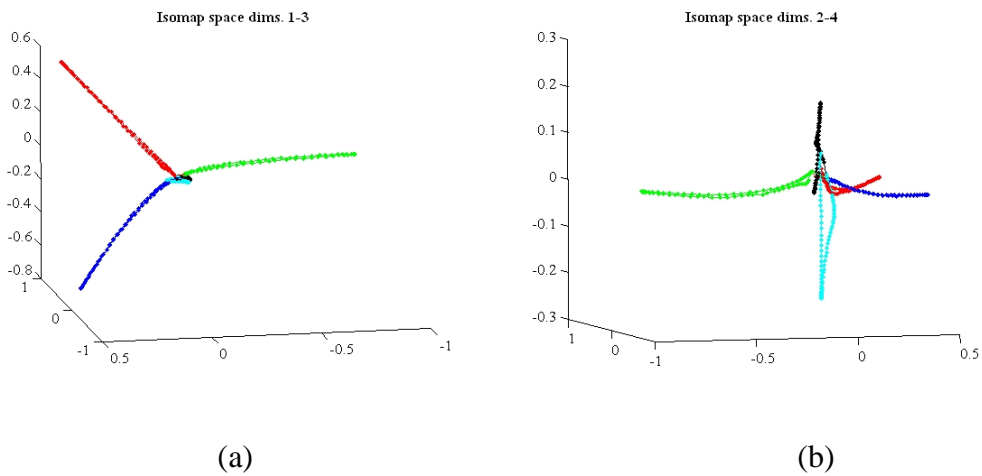


Figure 3.10. Five mean trajectories representing the five classes of atomic actions are plotted in different colors.

Next, we used three test action sequences to evaluate the performance of the proposed temporal segmentation method and the action classifier. The projection of the test data into the Isomap space was accomplished by using the method proposed in [33]. In the first two test sequences, all the atomic actions performed by the two subjects belonged to the five learned action clusters. The first test sequence was performed by subject 1, who provided 22 atomic actions. Figure 3.11 shows the atomic action trajectories constructed by mapping the new test data into the Isomap space using the method described in Section 3.2.5. In this experiment, all atomic actions were correctly segmented and classified. The second test sequence, which contained 46 atomic actions, was obtained from subject 2. In the constructed atomic action trajectories, shown in Figure 3.12, all the atomic actions were also correctly segmented and classified. The third test sequence was obtained by asking subject 1 to perform new actions that were different from all the trained atomic actions. The constructed atomic action trajectories and five mean trajectories are shown in Figure 3.13. Using the proposed action classification method, these atomic actions were all successfully classified as unknown actions.

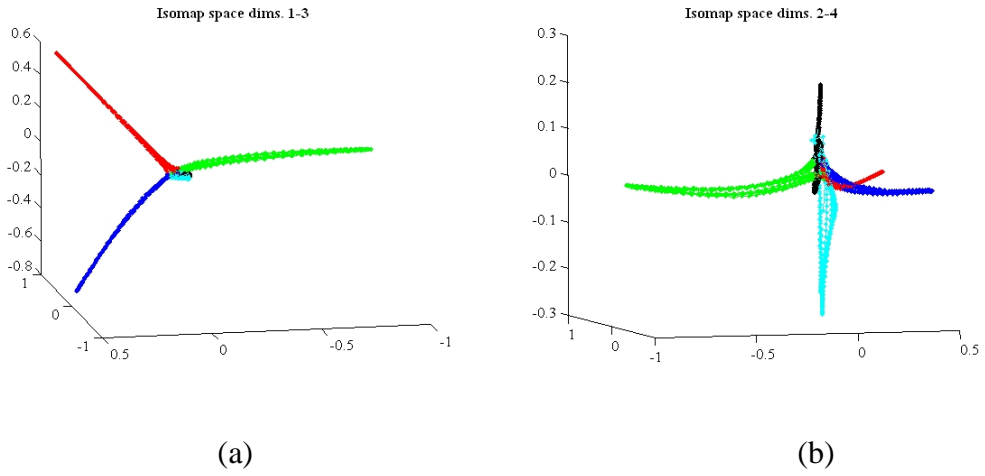


Figure 3.11. The atomic action trajectories constructed from test data sequence 1 and the classification results

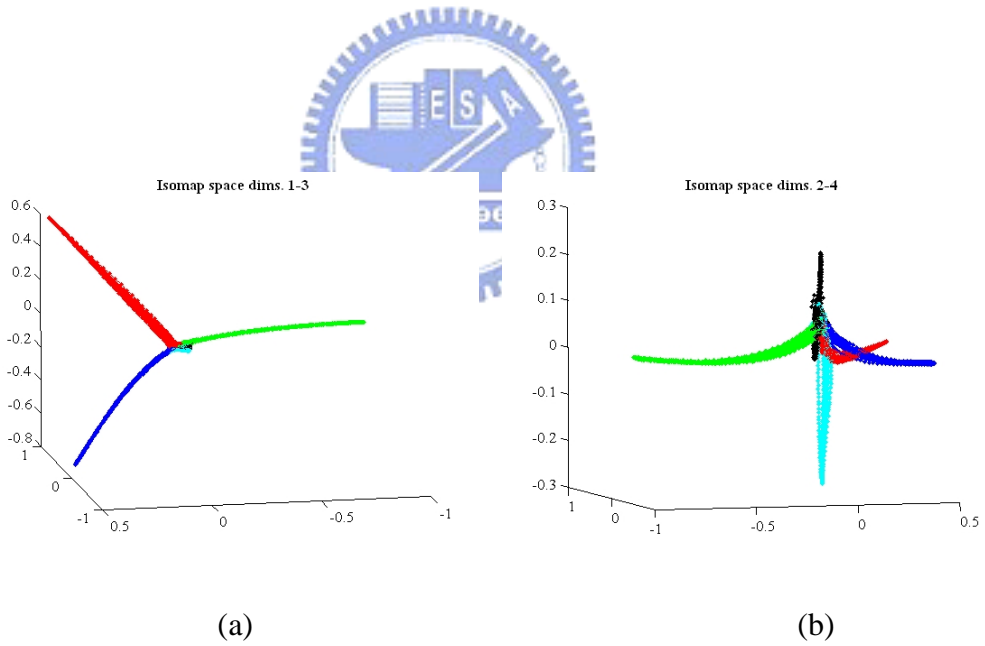


Figure 3.12. The atomic action trajectories constructed from test data sequence 2 and the classification results

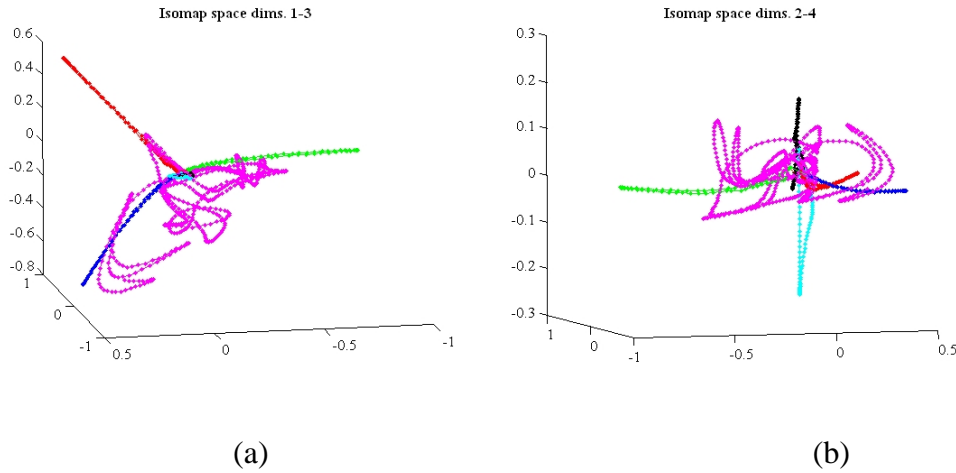


Figure 3.13. The atomic action trajectories constructed from test data sequence 3 superimposed on to the five learned exemplar trajectories.

In the final experiment, we evaluated the performance of the proposed action classifier based on the accelerated mapping approach. First, we tested the accuracy of the proposed mapping method by remapping the training sequence into the Isomap space using Equation (3.14) and evaluating the reconstruction error of the new method. The reconstruction error is defined to as the average distance between the reconstructed and the original Isomaps. Figure 3.14 shows a set of selected key data points and the reconstructed Isomap. From the figure it is obvious that the reconstructed Isomap is very similar to the original Isomap shown in Figure 3.8. Clearly, the number of selected key data points is a crucial issue because the number of points has a substantial influence on both the recognition rate and the computation time. Selecting too many key data points will result in low computational efficiency, while selecting too few key data points will lead to inaccurate results. To determine the appropriate number of key data

points, we calculate the reconstruction error using different percentages of selected key points over the total number of data points. Figure 3.15 shows that when the percentage of selected key data points falls within the range 1.2% to 1.4%, the reconstruction error and the number of key data points would both be sufficiently small. Therefore, we use this range as the threshold for the selecting appropriate number of key points. In this experiment, the number of key data points was set to 25 (about 1.26% of the total number of 1,983 data points). Compared to the original approach in which we have to evaluate 1,983 CSC distances, the computational complexity is dramatically reduced to the evaluation of only 25 CSC distances. This is approximately 79 times faster than the original approach. Based on the accelerated mapping method, we tested the recognition rate using the three test action sequences. Figures 3.16-3.18 show the reconstructed atomic action trajectories derived by mapping the new test data into the simplified Isomap space. It is obvious that the proposed method does not degrade the recognition performance at all. The results show that the proposed method is fast and accurate.

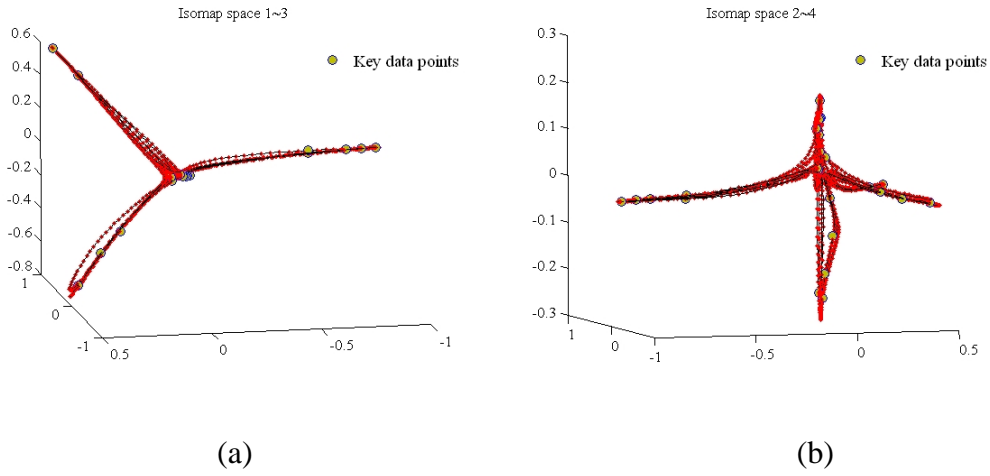


Figure 3.14. The selected key data points and the reconstructed Isomap space

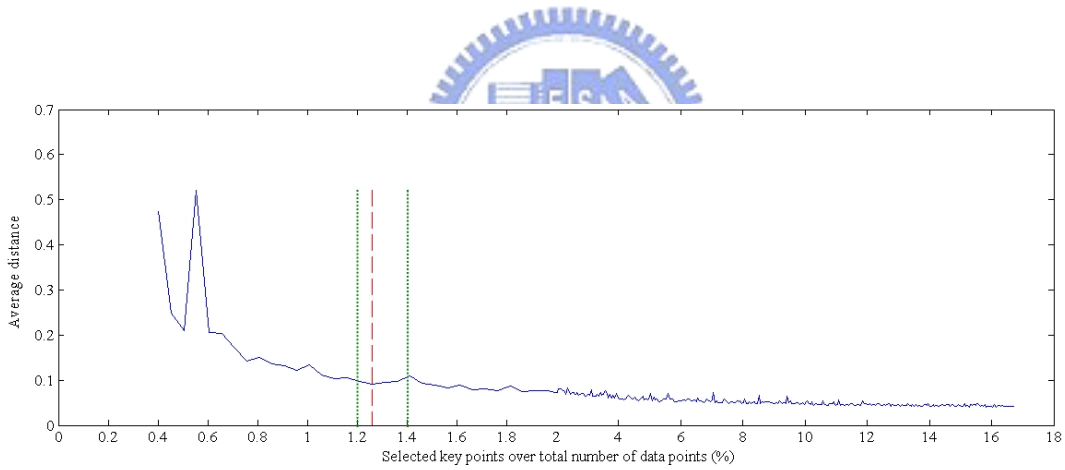


Figure 3.15. The average distance between the original Isomap and the reconstructed Isomap using different percentages of selected key points over the total number of data points

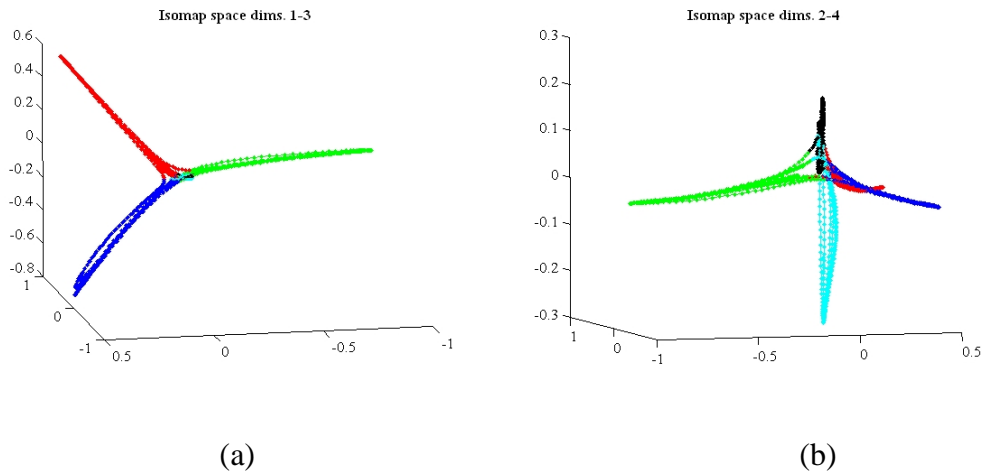


Figure 3.16. The reconstructed atomic action trajectories derived from test data sequence 1 and the classification results based on the simplified action classification approach.

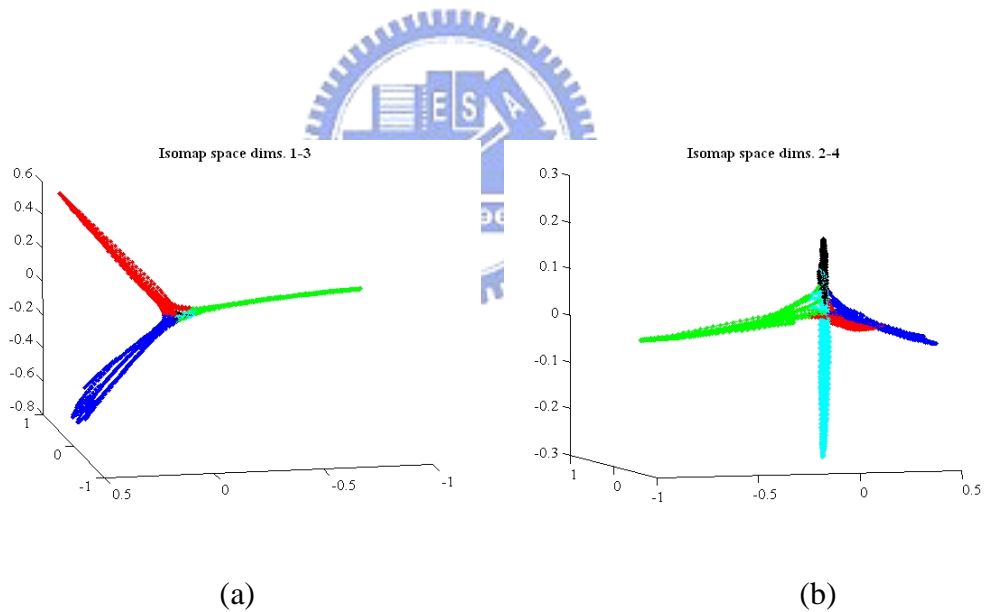


Figure 3.17. The reconstructed atomic action trajectories derived from test data sequence 2 and the classification results based on the simplified action classification approach.

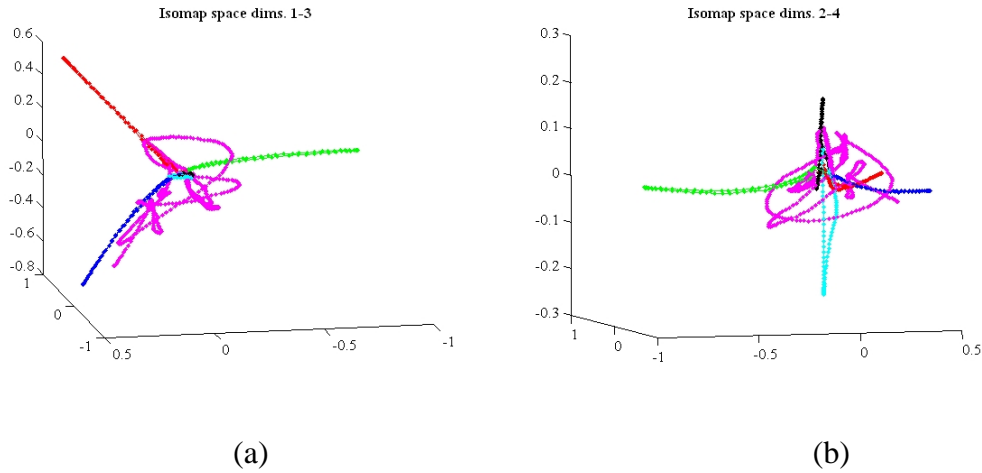


Figure 3.18. The reconstructed atomic action trajectories derived from test data sequence 3 based on the simplified action classification approach superimposed on to the five learned exemplar trajectories.

3.4 Concluding Remarks



In this chapter, we have proposed a framework for unsupervised analysis of long and unsegmented human action sequences based on Isomap learning. The framework comprises four modules: an Isomap learning module, a temporal segmentation module, an atomic action clustering module, and an atomic action learning and classification module. First, we calculate a pairwise CSC distance matrix from the training action sequence, and then apply the Isomap algorithm to construct a low-dimensional structure from the distance matrix. Next, the data points in the Isomap space are represented as a time series of low-dimensional points, and a temporal segmentation process is used to segment this sequence into atomic actions. A DTW approach is then applied to cluster the atomic actions. Finally, the clustering results are used to learn and classify atomic actions. In

3.4 Concluding Remarks

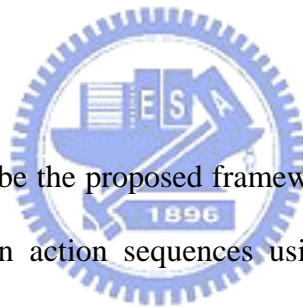
addition, to speed up the mapping from a new input posture into the Isomap space, we propose an efficient method that is approximately 79 times faster than the original approach. Our experiment results demonstrate the efficacy of the proposed framework.





Chapter 4

Learning Atomic Human Actions Using Variable-Length Markov Models



In this chapter, we describe the proposed framework for learning and recognizing segmented atomic human action sequences using VLMM. First, we give an introduction about this research topic. The proposed approach is then described. Next, we detail the experiment results. Finally, we present our conclusions.

4.1 Introduction

Since the human body is an articulated object with many degrees of freedom, inferring a body posture from a single 2-D image is usually an ill-posed problem. Providing a sequence of images might help to solve the ambiguity of action recognition. However, to integrate the information extracted from the images, it

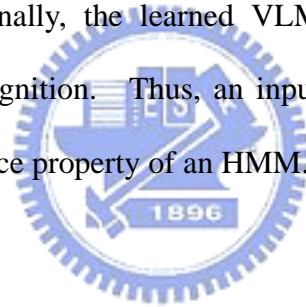
is essential to find a model that can effectively formulate the spatial-temporal characteristics of human actions. Note that if a continuous human posture can be quantized into a sequence of discrete postures, each one can be regarded as a letter of a specific language. Consequently, an atomic action composed of a short sequence of discrete postures can be regarded as a verb of that language. Sentences and paragraphs that describe human action can then be constructed, and the semantic description of a human action can be determined by a language modeling approach.

In a natural language, the most informative word of a sentence is usually its verb. Because an atomic action acts the verb of a sentence in a natural language, it is vital to recognize each atomic action in order to transform an input video sequence into semantic-level descriptions. In this work, we focus on the problem of automatic action recognition by using a language modeling approach to bridge the semantic gap between an atomic action sequence and a verb.

Language modeling [30, 53], a powerful tool for dealing with temporal ordering problems, has been applied in many fields, such as speech recognition [30, 32], handwriting recognition [47, 61], and information retrieval [16, 71]. In this chapter, we consider its application to the analysis of human action. A number of approaches have been proposed thus far. For example, Bobick and Ivanov [7] and Ogale et al. [44] used context-free grammars to model human actions, while Park et al. employed hierarchical finite state automata to recognize human behavior [46]. In [69, 70], HMMs were applied to human action

recognition. The HMM technique is useful for both human action recognition and human action sequence synthesis. Galata et al. utilized VLMMs to characterize human actions [22], and showed that VLMMs trained with motion-capture data or silhouette images can be used to synthesize human action animations. Existing language modeling approaches for action analysis can be categorized into two classes: deterministic algorithms [7, 44, 46] and stochastic algorithms [22, 69, 70]. Since the latter have higher degrees of freedom than the former, they are suitable for a wider range of applications. Currently, the HMM is the most popular stochastic algorithm for language modeling because of its versatility and mathematical simplicity. However, since the states of an HMM are not observable, encoding high-order temporal dependencies with this model is a challenging task. There is no systematic way to determine the topology of an HMM or even the number of its states. Moreover, the training process only guarantees a local optimal solution; thus, the training result is very sensitive to the initial values of the parameters. On the other hand, since the states of a VLMM are observable, its parameters can be estimated easily given sufficient training data. Consequently, a VLMM can capture both long-term and short-term dependencies efficiently because the amount of memory required for prediction is optimized during the training process. However, thus far, the VLMM technique has not been applied to human action recognition directly because of two limitations: 1) it cannot handle the dynamic time warping problem, and 2) it lacks a model for handling the noise observation.

In this research, we propose a hybrid framework of VLMM and HMM that retains the models advantages, while avoiding their drawbacks. The framework is comprised of two modules: a posture labeling module, and a VLMM atomic action learning and recognition module. First, a posture template selection algorithm is developed based on the CSC technique, discussed in Section 3.2.1. The selected posture templates constitute a codebook, which is used to convert input posture sequences into discrete symbol sequences for subsequent processing. Then, the VLMM technique is applied to learn the symbol sequences that correspond to atomic actions. This avoids the problem of learning the parameters of an HMM. Finally, the learned VLMMs are transformed into HMMs for atomic action recognition. Thus, an input posture sequence can be classified with the fault tolerance property of an HMM.



4.2 The Proposed Method for Atomic Action Recognition

The proposed method comprises two phases: 1) posture labeling, which converts a continuous human action into a discrete symbol sequence; and 2) application of the VLMM technique to learn the constructed symbol sequences and recognize the input posture sequences. The two phases are described below.

4.2.1 Posture Labeling

To convert a human action into a sequence of discrete symbols, a codebook of posture templates must be created as an alphabet to describe each posture. Although the codebook should be as complete as possible, it is important to minimize redundancy. Therefore, a posture is only included in the codebook if it cannot be approximated by existing codewords, each of which represents a human posture. In this work, a human posture is represented by a silhouette image, and a shape matching process is used to assess the difference between two shapes. Figure 4.1 shows the block diagram of the proposed posture labeling process. First, a low-level image processing technique is applied to extract the silhouette of a human body from each input image. Then, the codebook of posture templates computed from the training images is used to convert the extracted silhouettes into symbol sequences. Shape matching and posture template selection are the most important procedures in the posture labeling process. Shape matching has been described in Section 3.2.1, and posture template selection is discussed in the following.

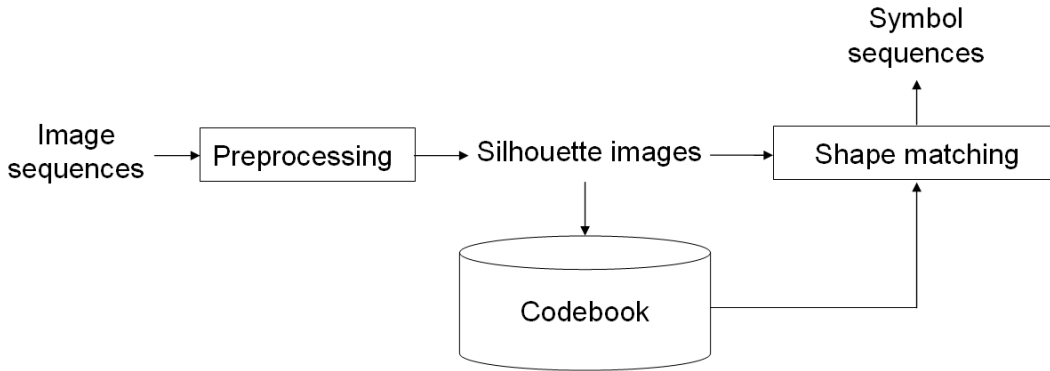


Figure 4.1. Block diagram of the proposed posture labeling process

Posture Template Selection

Posture template selection is used to construct a codebook of posture templates from training silhouette sequences. If the training atomic action sequences are segmented from a long human action sequence based on Isomap learning, we can use the key data points described in Section 3.2.5 as the codebook. Otherwise, for general segmented atomic action sequences, we propose an automatic posture template selection algorithm (see Algorithm 4.1), based on the CSC technique. In the posture template selection method, the cost of matching two shapes (see Equation (3.2)), is denoted by $D_{\text{csc}}(b_i, a_j)$. We only need to empirically determine one threshold parameter τ_c in our posture template selection method. This parameter determines whether a new training sample should be incorporated into the codebook. The selection of τ_c is not unique for all cases. Because incoming action sequences may contain any kind of action, the selection of τ_c is

basically an ill-posed problem in mathematics. Therefore, we cannot determine a universal τ_c to fit in all cases. In fact, the selection of τ_c is not a major concern in this work because our objective is to establish an automatic posture template selection scheme.

Algorithm 4.1: Posture Template Selection

Codebook of posture templates: $A = \{a_1, a_2, \dots, a_M\}$

Training sequence: $T = \{t_1, t_2, \dots, t_N\}$

for each $t \in T$ do {

if ($A = \phi$ or $\min_{a \in A} D_{\text{csc}}(t, a) > \tau_c$) {

$A \leftarrow A \cup \{t\}$

$M \leftarrow M + 1$

}

}

4.2.2 Human Action Sequence Learning and Recognition

Atomic Action Learning

Using the codebook of posture templates, an input sequence of postures $\{b_1, b_2, \dots, b_n\}$ can be converted into a symbol sequence $\{a_{q(1)}, \dots, a_{q(n)}\}$, where $q(i) = \arg \min_{j \in \{1, 2, \dots, M\}} D_{\text{esc}}(b_i, a_j)$. Thus, atomic action VLMMs can be trained by the method outlined in Section 2.3.1. These VLMMs are actually different order Markov chains. For simplicity, we transform all the high order Markov chains into first-order Markov chains by augmenting the state space. For example, the probability of a d_i -th order Markov chain with state space S is given by

$$P(X_i = r_i \mid X_{i-d_i} = r_{i-d_i}, X_{i-d_i+1} = r_{i-d_i+1}, \dots, X_{i-1} = r_{i-1}), \quad (4.1)$$

where X_i is a state in S . To transform the d_i -th order Markov chain into a first-order Markov chain, a new state space is constructed such that both $Y_{i-1} = (X_{i-d_i}, \dots, X_{i-1})$ and $Y_i = (X_{i-d_{i+1}-1}, \dots, X_i)$ are included in the new state space. As a result, the high order Markov chain can be formulated as the following first-order Markov chain [24]

$$\begin{aligned} P(X_i = r_i \mid X_{i-d_i} = r_{i-d_i}, X_{i-d_i+1} = r_{i-d_i+1}, \dots, X_{i-1} = r_{i-1}) \\ = P(Y_i = (r_{i-d_{i+1}-1} \dots r_i) \mid Y_{i-1} = (r_{i-d_i} \dots r_{i-1})). \end{aligned} \quad (4.2)$$

Hereafter, we assume that every VLMM has been transformed into a first-order Markov model.

Atomic Action Recognition

After the VLMMs are trained from the training sequence, the VLMM recognition technique, mentioned in Section 2.3.2, can be applied to atomic action recognition. This VLMM recognition technique works well for natural language processing. However, since natural language processing and human action analysis are inherently different, two problems must be solved before the VLMM technique can be applied to atomic action recognition. First, the VLMM technique cannot handle the dynamic time warping problem; hence VLMMs cannot recognize atomic actions when they are performed at different speeds. Second, the VLMM technique does not include a model for noise observation, so the system is less tolerant of image preprocessing errors.

First, note that the speed of the action affects the number of repeated symbols in the constructed symbol sequence: a slower action produces more repeated symbols. To eliminate this speed-dependent factor, the input symbol sequence is preprocessed to merge repeated symbols. VLMMs corresponding to different atomic actions are trained with preprocessed symbol sequences similar to the method proposed by Galata et al. [22]. However, this approach is only valid when the observed noise is negligible, which is an impractical assumption. The

recognition rate of the constructed VLMMs is low because image preprocessing errors may identify repeated postures as different symbols. To incorporate a noise observation model, the VLMMs trained with unrepeated sequences must be modified to recognize input sequences with repeated symbols. Let a_{ij} denote the state transition probability from state i to state j . Initially, $a_{ii}^{old} = 0$ because the training data contains no repeated symbols. The self-transition probability is updated by $a_{ii}^{new} = P(v_i | v_i) + \delta$, where $P(v_i | v_i) = \frac{N(v_i v_i)}{N(v_i)}$ computed with the original training sequences and δ is a small positive number to prevent the over-fitting problem [49]. Note that if the self-transition probability is zero, then an action sequence that contains repetition will result in a zero probability such that the system will not perform normally when faced with slower action sequences. To overcome this limitation, we add the small positive number δ to the self-transition probability. This parameter can be determined using the cross-validation method. The other transition probability must also be updated as $a_{ij}^{new} = a_{ij}^{old} (1 - a_{ii}^{new})$. For example, if the input training symbol sequence is “AAABBAAACCAAABB,” the preprocessed training symbol sequence becomes “ABACAB.” The VLMM constructed with the original input training sequence is shown in Figure 4.2(a); while the original VLMM and modified VLMM constructed with the preprocessed training sequence are shown in Figures 4.2(b) and 4.2(c), respectively.

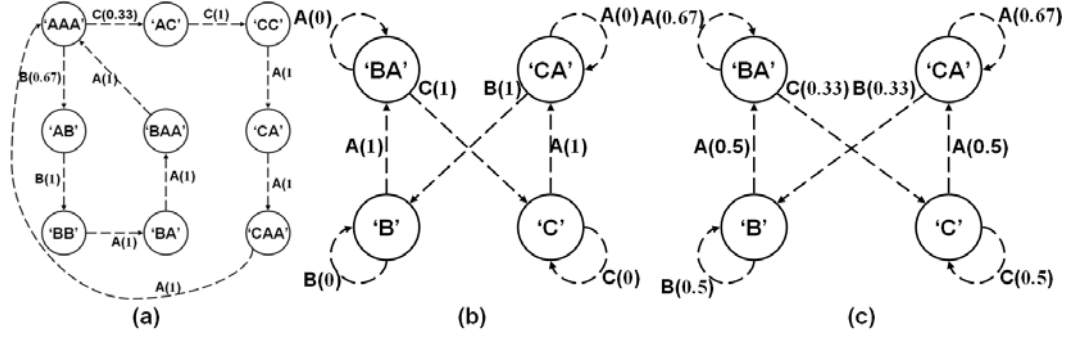


Figure 4.2. (a) The VLMM constructed with the original input training sequence; (b) the original VLMM constructed with the preprocessed training sequence; (c) the modified VLMM, which includes the possibility of self-transition.

Next, a noise observation model is introduced to convert a VLMM into an HMM. Note that the output of a VLMM determines its state transition and vice versa because the state of a VLMM is observable. In general, the possible output is restricted to several discrete symbols. However, due to the noise caused by image preprocessing, the symbol sequence corresponding to an atomic action includes some randomness. Such randomness will cause the action sequence not recognizable by the VLMMs. Therefore, we propose to modify the symbol observation model as described in the following. Suppose that the output symbol of a VLMM is q_t at time t , and its posture template retrieved from the codebook is a_{q_t} . If the VLMM is the right model, the extracted silhouette image o_t will not deviate too much from its corresponding posture template a_{q_t} provided that the segmentation result does not contain any major errors. Due to noise observation, the silhouette image o_t is a random variable, and so is the

CSC distance $D_{\text{csc}}(o_t, a_{q_t})$. It is possible to learn the distribution of the CSC distance, $D_{\text{csc}}(o_t, a_{q_t})$, using the training data. An example is shown in Figure

4.3. In this example, it is clear that a Gaussian distribution can be applied to

model the CSC distance, i.e. $P(o_t | q_t, \Lambda) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{D_{\text{csc}}(o_t, a_{q_t})}{2\sigma^2}}$. The standard deviation

σ of this distribution is estimated using the maximum-likelihood technique.

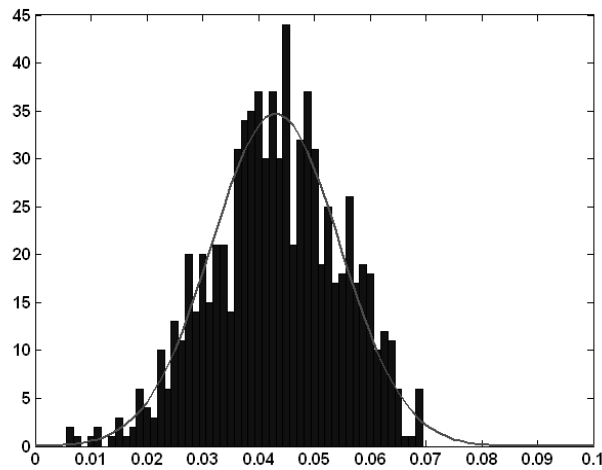


Figure 4.3. The distribution of observation error, obtained using the training data.

Note that the VLMM has now been converted into a first-order Markov chain. If the VLMM's observation model is detached from the symbol of a state, then the VLMM becomes a standard HMM. The probability of the observed silhouette image sequence, $O = o_1 o_2 \dots o_T$, for a given model Λ can be evaluated by the HMM forward/backward procedure with proper scaling [49]. Finally, category i^* obtained with the following equation is deemed to be the recognition result:

$$i^* = \arg \max_i \log[P(O | \Lambda_i)]. \quad (4.3)$$

4.3 Experiments

We conducted a series of experiments to evaluate the effectiveness of the proposed method. A powerful, scalable recognition system would only use the data extracted from one person for training but would still be capable of recognizing data collected from other people. Accordingly, the training data used in our experiments was a real video sequence comprised of approximately 900 frames. The training data contained ten categories of action sequences that were performed by a single person. Some typical image frames are shown in Figure 4.4. Using the posture template selection algorithm, a codebook of 95 posture templates (see Figure 4.5), was constructed from the training data. The data was then used to build ten VLMMs, each of which was associated with one of the atomic actions shown in Figure 4.4.

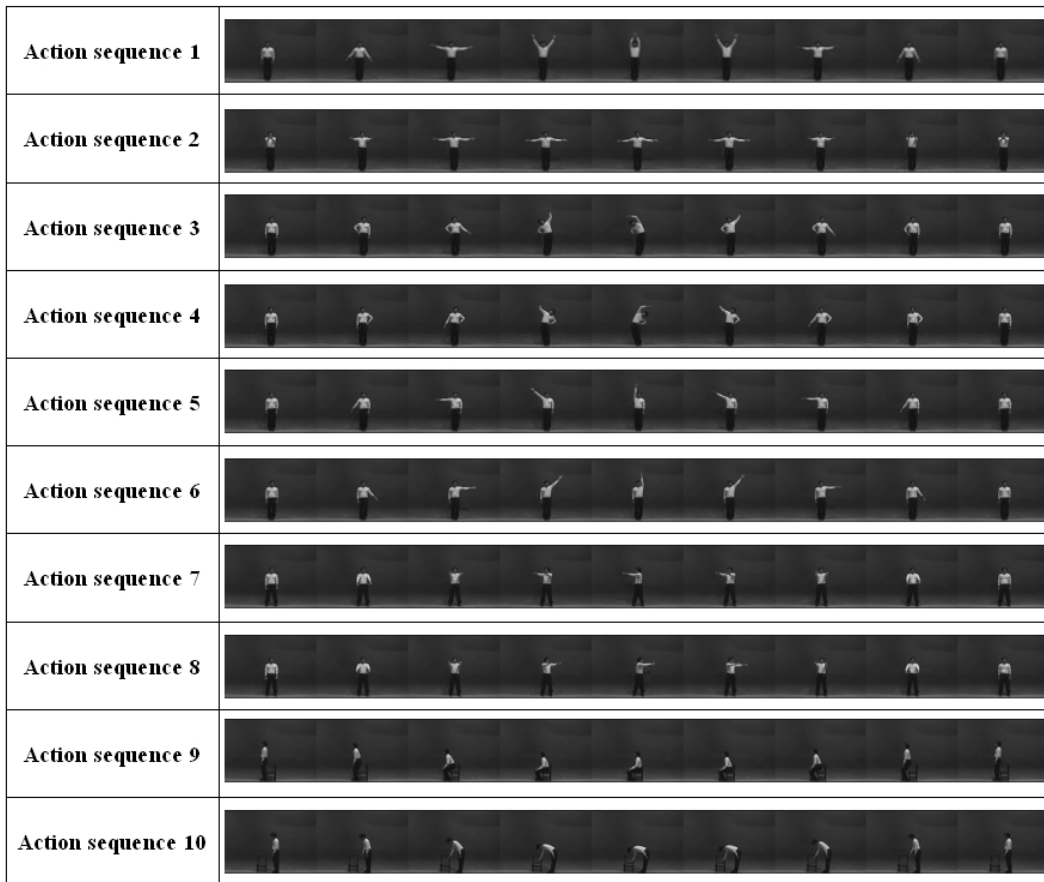


Figure 4.4. The ten categories of atomic actions used for training

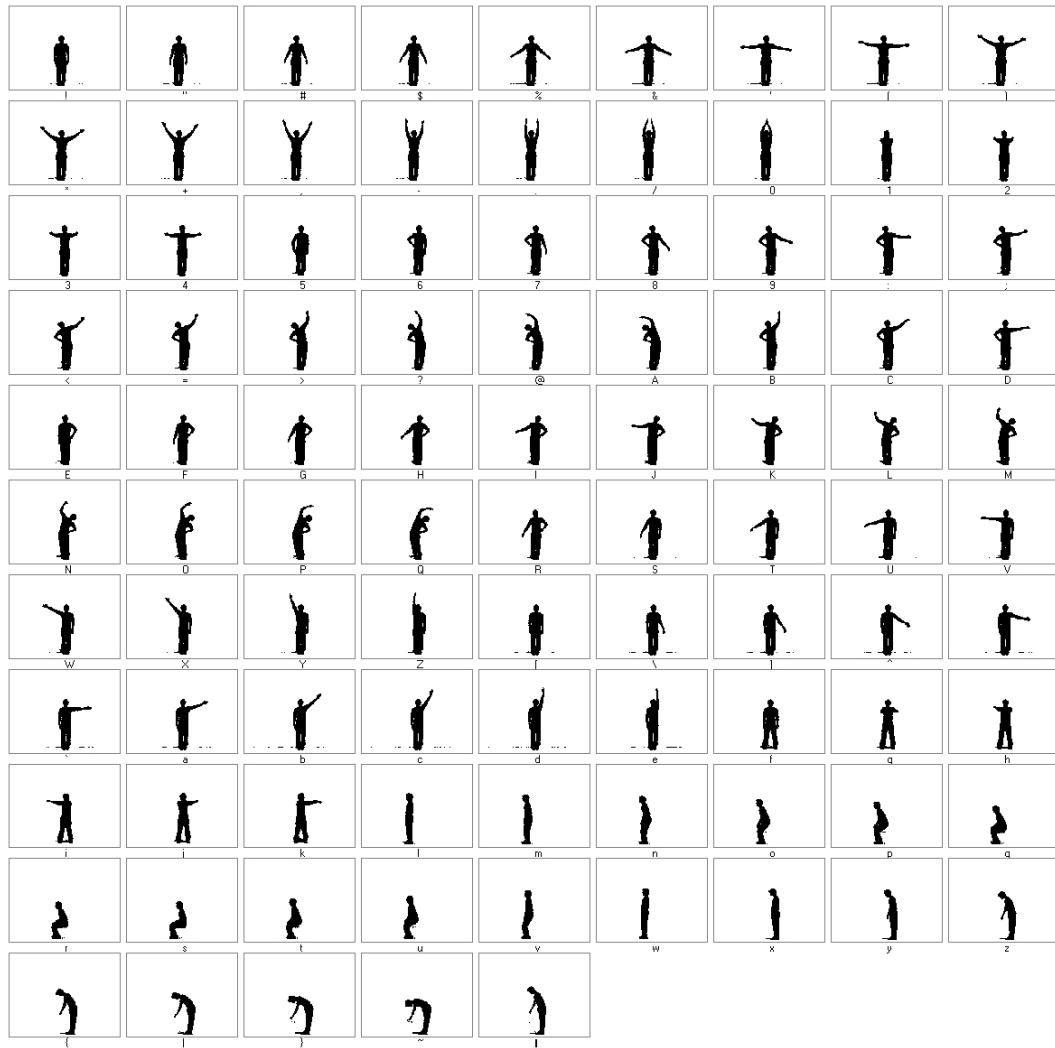


Figure 4.5. Posture templates extracted from the training data

The average log-likelihood of the training error computed with the training data is shown in Table 4.1. The results indicate that the proposed action recognition method can deal with the problem of human action recognition effectively. Next, a test video was used to assess the effectiveness of the proposed method. The test data was obtained from the same human subject. Each atomic action was repeated four times, yielding a total of 40 test samples (4

positive samples and 36 negative samples) for evaluating the performance of the learnt VLMMs. The proposed method achieved a 100% recognition rate for all the test sequences. To further verify the recognition results, we tested the similarity of any two VLMMs obtained in the experiment. First, we generated 10,000 action sequences for each of the 10 VLMMs, which yielded a total of 100,000 action sequences. Out of the 100,000 action sequences, only 74 sequences were incorrectly recognized and all the errors were on actions 7 and 8 because these two sequences contained many similar postures and thus could be mixed up easily (refer to Figure 4.4). This result is consistent with the data shown in Table 4.1: the log-likelihood of actions 7 and 8 computed using VLMMs 8 and 7 were relatively high. This result confirms that the data shown in Table 4.1 is valid. Furthermore, we have also estimated the p-values [73] for each action model. The posture templates shown in Figure 4.5 were used to generate 10,000 random action sequences using a sample-with-replacement process. The histograms of the log-likelihood of the random sequences and the positive sequences for an action model are shown in Figure 4.6. Since these two histograms do not overlap at all, it is reasonable to infer that the p-value of the action model is very low. To estimate the p-value, we approximate the distributions of the log-likelihood by Gaussian distributions (see Figure 4.6). Therefore, the p-value can be easily computed. The maximum p-value of the ten models is smaller than 0.0001, which confirms that the results are statistically significant.

Table 4.1. The results of atomic action recognition using the training data

VLMM Log Likelihood Action	1	2	3	4	5	6	7	8	9	10
1	-5.707	-63.78	-73.66	-81.2	-91.82	-91.12	-240	-211.1	-206.3	-239.9
2	-27.82	-5.944	-83.2	-67.84	-109.1	-107.7	-167.7	-156.1	-358.1	-259.5
3	-49.42	-76.62	-5.39	-64.79	-65.27	-42.27	-110.1	-100.6	-158.6	-162.6
4	-52.99	-83.22	-75.9	-5.524	-50.32	-80.6	-108.7	-115.6	-157.8	-177.5
5	-71.96	-81.93	-66.6	-46.16	-5.603	-89.7	-111.1	-119.1	-125.5	-119
6	-79.9	-100	-39.61	-87.91	-94.46	-5.559	-142.6	-126	-178.6	-254.2
7	-122.7	-75.95	-91.43	-110.7	-85.54	-96.43	-5.764	-9.797	-150.2	-150
8	-117.4	-87.62	-104.6	-103.9	-117.1	-81.35	-24.05	-5.884	-135.7	-159.5
9	-152.6	-149.3	-171.6	-131.4	-134.4	-124.3	-141.3	-140.9	-5.134	-111.8
10	-185.4	-198.1	-161.3	-166.7	-128.6	-224.1	-189.3	-192	-206.6	-5.453

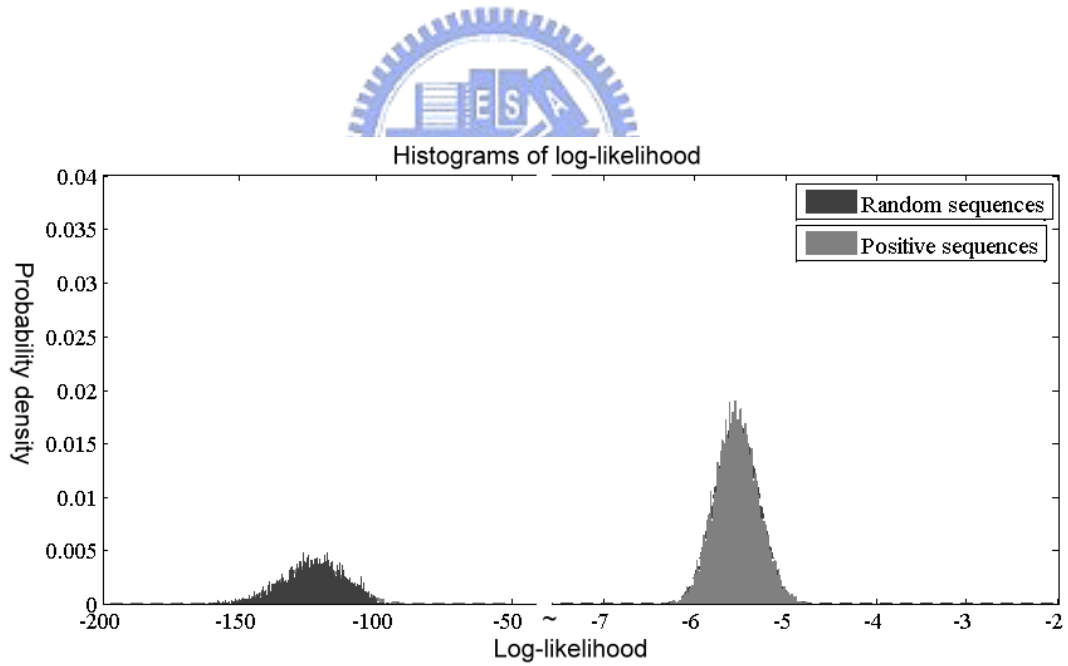


Figure 4.6. The histograms of the log-likelihood of the random sequences and the positive sequences for an action model

In the third experiment, test videos of nine different human subjects (see Figure 4.7) were used to evaluate the performance of the proposed method. Each person repeated each action five times, so we had five sequences for each action and each human subject, which yielded a total of 450 action sequences. For comparison, we also tested the performance of the HMM method in this experiment. Since the ten atomic actions used in the experiments were acyclic, only the left-right HMMs were considered in this experiment. Because the initial parameters and the number of HMM states would affect recognition results, the HMM implementation was evaluated using a variety of HMMs, each of which had a different number of hidden states. Furthermore, the HMM were trained ten times and the average results were used to reduce the effect of the initial random parameters. Table 4.2 compares our method's recognition rate with that of the HMM method, for test data from nine different human subjects. Our method clearly outperforms the HMM method, no matter how many states were selected. In Table 4.2, the shaded cells denote the best recognition results of the HMM approach for a particular action. It is clear that the selection of the number of states is a critical issue for the HMM method. Note that the number of HMM states that could be set for deriving the best performance was varying in different actions which makes the selection of the number of states even more difficult. In contrast to the difficulty in determining the topology of an HMM, our method is simple and effective because the topology of a VLMM can be determined automatically with a robust algorithm. Note that the recognition rates for action 1 were the worst across all actions. Figure 4.8(a) shows some typical input

postures for a human subject performing action 1. The retrieved, corresponding closest posture templates in the database are shown in Figure 4.8(b). When comparing the corresponding posture templates shown in Figure 4.8(b) with the training posture sequences shown in Figure 4.4, it is clear that the posture templates and the training postures of action 1, in this case, are not well matched. Due to the segmentation error of the lower arms areas, the input postures were incorrectly related to posture templates of different actions. For example, the retrieved posture templates shown in Figure 4.8(b), from left to right, were extracted from training data of actions 1, 4, 2, 2, 2, 1, 2, 2, 2, 4, and 1, respectively. Since the proposed method is silhouette-based, when the same postures of two individuals appear to be drastically different (due to dissimilar physical characteristics, motion styles, or improper segmentation), observation errors would bias the recognition result. In particular, if most of the input postures are with high observation error, the context information is not sufficient for accurate performance.



Figure 4.7. Nine test human subjects

Table 4.2. Comparison of our method's recognition rate with that of the HMM computed with the test data obtained from nine different human subjects

Actions Recognition rate(%) Methods	1	2	3	4	5	6	7	8	9	10
Our method	88.89	97.78	100	100	100	100	97.78	100	100	97.78
HMM (5 states)	88.22	82.00	93.78	87.78	88.22	90.89	96.89	99.78	90.44	97.56
HMM (10 states)	87.56	78.89	93.11	92.00	97.78	77.78	96.22	98.44	87.33	97.78
HMM (15 states)	88.89	81.33	93.11	93.11	92.89	66.44	97.33	99.33	98.89	97.33
HMM (20 states)	88.89	80.00	93.33	92.22	95.56	77.56	97.11	98.44	90.89	97.56
HMM (25 states)	88.89	81.56	93.56	92.67	93.56	60.89	95.56	100	98.89	97.56
HMM (30 states)	88.89	81.33	93.78	93.56	94.00	57.78	95.56	99.78	85.78	97.33

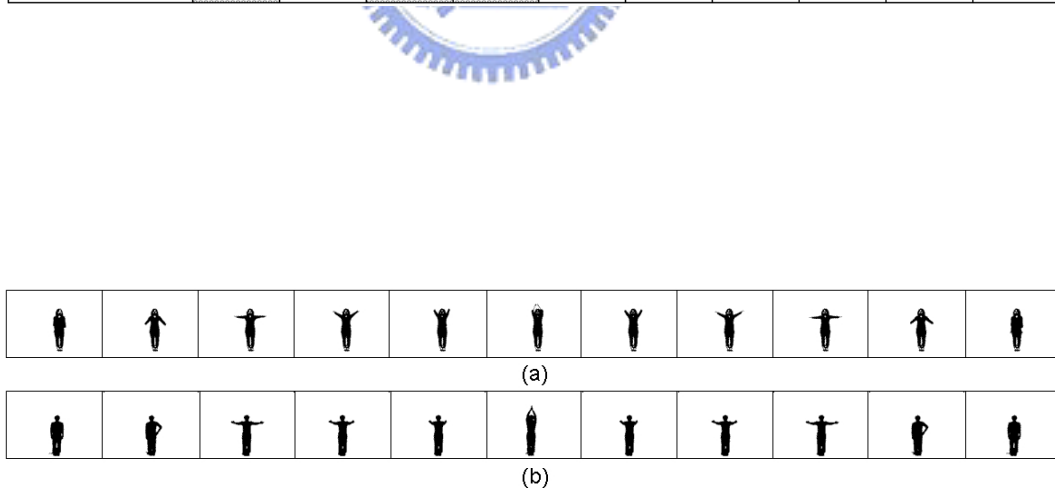


Figure 4.8. Some typical postures of a human subject exercising action 1: (a) the input posture sequence; (b) the corresponding minimum-CSC-distance posture templates.

In order to show that the selection of the parameter τ_c in the posture template selection process was not a major concern, we calculated the recognition rates for different τ_c . Figure 4.9 shows the recognition rates with respect to different τ_c , and it demonstrates that the change of τ_c only has little influence to the recognition results.

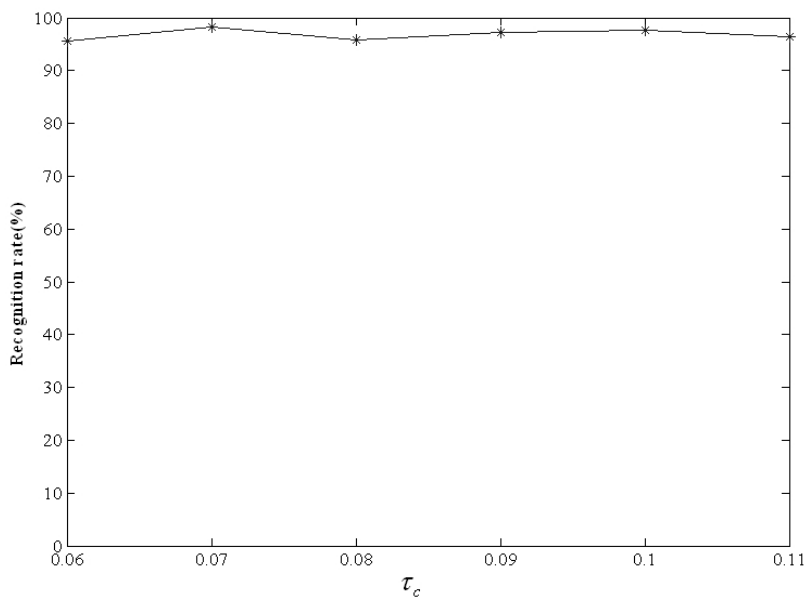


Figure 4.9. Recognition rates with respect to different τ_c

In the fourth experiment, to evaluate the scalability of the proposed algorithm, we used a new, publicly-available database [3, 63]. This database consists of 90 low-resolution (180×144) action sequences from nine different people, each performing ten natural actions. These actions include: bending (bend), jumping jacks (jack), jumping forward on two legs (jump), jumping in place on two legs

(pjump), running (run), galloping sideways (side), skipping (skip), walking (walk), waving one hand (wave1), and waving two hands (wave2). Sample images of each type of action sequence are shown in Figure 4.10. In [63], a sequence of human silhouettes derived from each action sequence was converted into two representations, namely average motion energy (AME) and mean motion shape (MMS). Subsequently, a nearest neighbor classifier (NN) was used for recognition, and the leave-one-out cross-validation rule was adopted to compute the recognition rate. Recognition results for these two representations, shown in the top two rows of Table 4.3, are compared against our method.

In order to compare our method with the two competing methods in a fairer fashion, we also applied the leave-one-out rule to our method. In this case, eight sets of data grabbed from eight distinct human subjects were used to train the VLMMs, resulting in eight VLMMs for each action. Finally, the category with the maximum likelihood was deemed to be the recognition result. Results using this methodology are shown in the last row of Table 4.3. It is clear that our method outperforms the other two methods for this public database.

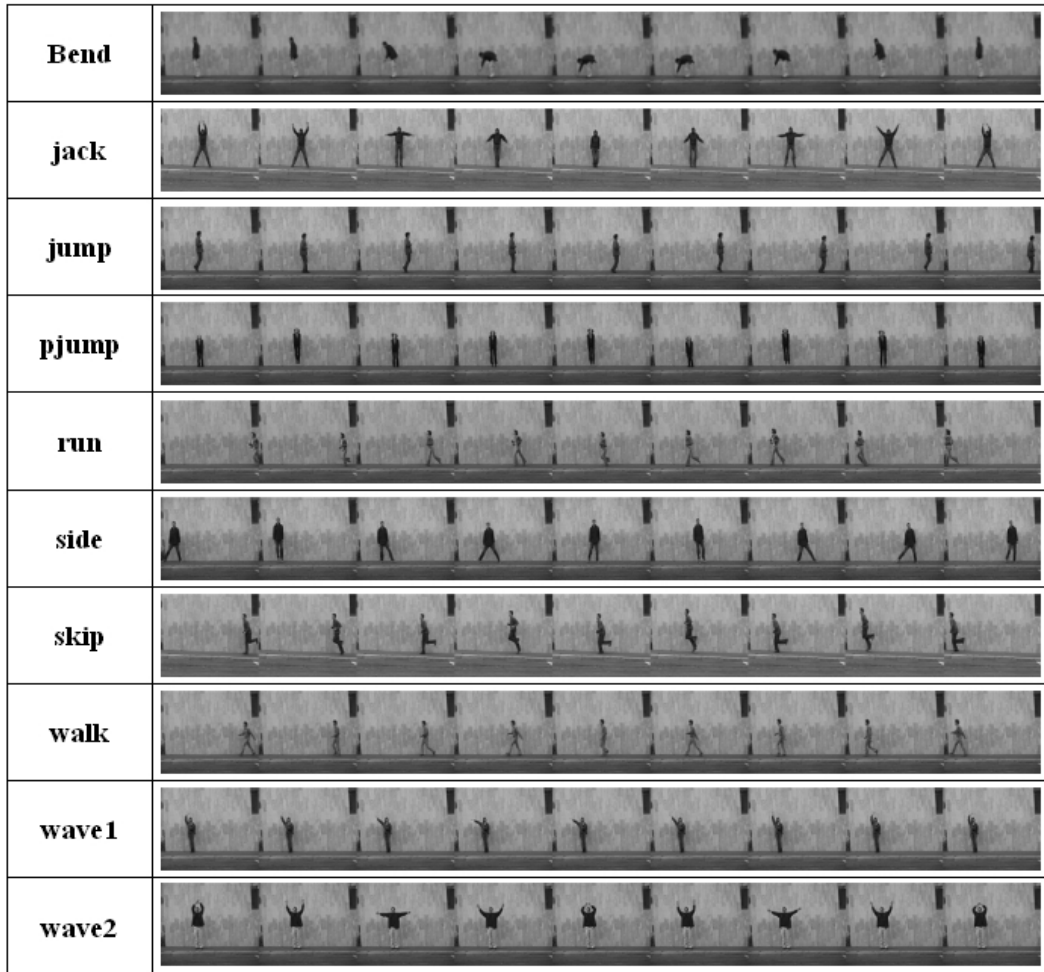


Figure 4.10. Sample images in the public action database

Table 4.3. Comparison of our method's recognition rate with that of the AME plus NN method and the MMS plus NN method for the public database

Actions Recognition rate(%) Methods	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2	total
AME plus NN [63]	100	100	88.89	100	100	100	88.89	100	88.89	100	96.67
MMS plus NN [63]	100	100	88.89	77.78	100	88.89	100	88.89	88.89	100	93.33
Our method	100	100	100	100	100	100	100	100	100	100	100

4.4 Concluding Remarks

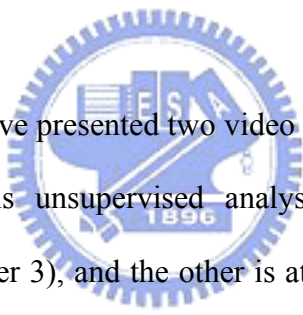
We have proposed a framework for understanding human atomic actions using VLMMs. The framework comprises two modules: a posture labeling module, and a VLMM atomic action learning and recognition module. We have developed a simple and efficient posture template selection algorithm based on the modified shape context matching method. A codebook of posture templates is created to convert the input posture sequences into discrete symbols so that the language modeling approach can be applied. The VLMM technique is then used to learn human action sequences. To handle the dynamic time warping problem and the lack of noise observation model problem of applying the VLMM technique to action analysis, we have also developed a systematic method to convert the learned VLMMs into HMMs. The contribution of our approach is that the topology of the HMMs can be automatically determined and the recognition accuracy is better than the traditional HMM approach. Experiment results demonstrate the efficacy of the proposed method.



Chapter 5

Conclusions and Future Work

5.1 Conclusions



In this dissertation, we have presented two video processing techniques for human action analysis. One is unsupervised analysis of human action based on manifold learning (Chapter 3), and the other is atomic human action learning and recognition using VLMMs (Chapter 4).

In Chapter 3, we have proposed a framework for unsupervised analysis of long and unsegmented human action sequences based on Isomap learning. The framework comprises five modules: an posture representation and matching module, an Isomap learning module, a temporal segmentation module, an atomic action clustering module, and an atomic action learning and classification module. First, we calculate a pairwise CSC distance matrix from the training action sequence, and then apply the Isomap algorithm to construct a low-dimensional structure from the distance matrix. Next, the data points in the Isomap space are

represented as a time series of low-dimensional points, and a temporal segmentation process is used to segment this sequence into atomic actions. A DTW approach is then applied to cluster the atomic actions. Finally, the clustering results are used to learn and classify atomic actions. In addition, to speed up the mapping from a new input posture into the Isomap space, we propose an efficient method that is approximately 79 times faster than the original approach. Our experiment results demonstrate the efficacy of the proposed framework.

In Chapter 4, we have proposed a framework for understanding human atomic actions using VLMMs. The framework comprises two modules: a posture labeling module, and a VLMM atomic action learning and recognition module. We have developed a simple and efficient posture template selection algorithm based on the modified shape context matching method. A codebook of posture templates is created to convert the input posture sequences into discrete symbols so that the language modeling approach can be applied. The VLMM technique is then used to learn human action sequences. To handle the dynamic time warping problem and the lack of noise observation model problem of applying the VLMM technique to action analysis, we have also developed a systematic method to convert the learned VLMMs into HMMs. The contribution of our approach is that the topology of the HMMs can be automatically determined and the recognition accuracy is better than the traditional HMM approach. Experiment results demonstrate the efficacy of the proposed method.

5.2 Future work

Since the CSC descriptor for human posture is not a view-invariant representation, we can not deal with same atomic actions with different views. Therefore, we shall handle this problem to make our system more scalable in the future. Moreover, high-level semantic description for human action using natural language will be another subject for our future work.





Bibliography

- [1] J. K. Aggarwal and Q. Cai, "Human motion analysis: a review," *Computer Vision and Image Understanding*, Vol. 73, No. 3, pp. 428- 440, 1999.
- [2] A. Ali and J. K. Aggarwal, "Segmentation and recognition of continuous human activity," *Proceedings of IEEE Workshop on Detection and Recognition of Events in Video*, pp. 28- 35, 2001.
- [3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Barsi, "Actions as space-time shapes," *Proceedings of the IEEE International Conference on Computer Vision*, Vol. 2, pp. 1395- 1402, 2005.
- [4] H.A. Baler Saip and C.L. Lucchesi, "Matching algorithm for bipartite graph," *Technical Report DCC-03/93, Departamento de Cincia da Computao, Universidade Estadual de Campinas*, 1993.
- [5] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 24, pp. 509- 522, 2002.

- [6] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, Vol. 15, No. 6, pp. 1373-1396, 2003.
- [7] A. F. Bobick and Y. A. Ivanov, "Action recognition using probabilistic parsing," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 192- 202, Santa Barbara, California, 1998.
- [8] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 3, pp. 257-267, 2001.
- [9] B. Boulay, F. Bremond, and M. Thonnat, "Human posture recognition in video sequence," *Proceedings IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 23- 29, 2003.
- [10] M. Broniatowski, "Estimation of the Kullback-Leibler Divergence," *Mathematical Methods of Statistics*, 2003.
- [11] D. Y. Chen, S. W. Shih, and H. Y. Mark Liao, "Atomic human action segmentation using a spatio-temporal probabilistic," *Proceedings of IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 327- 330, Pasadena, CA, USA, Dec. 2006.
- [12] T. J. Chin, L. Wang, K. Schindler, and D. Suter, "Extrapolating learned manifolds for human activity recognition," *Proceedings of the IEEE*

International Conference on Image Processing, Vol. 1, pp. 381- 384, 2007.

[13] K. D. Cock and B. D. Moor, "Subspace angles and distances between ARMA models," *Proceedings of the Mathematical Theory of Networks and Systems*, 2000.

[14] R. T. Collins, A. J. Lipton, and T. Kanade, "Introduction to the special section on video surveillance," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 745- 746, 2000.

[15] T. F. Cox and M.A.A Cox, *Multidimensional Scaling*. Chapman and Hall, 2001.

[16] W. B. Croft and J. Lafferty, *Language Modeling for Information Retrieval*, Kluwer Academic Publishers, Norwell, MA, 2003.

[17] N. P. Cuntoor and R. Chellappa, "Key frame-based activity representation using antieigenvalues," *ACCV 2006, LNCS 3852*, pp. 499- 508, 2006.

[18] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 269-274, 2001.

[19] S. L. Dockstader, M. J. Berg, and A. M. Tekalp, "Stochastic kinematic modeling and feature extraction for gait analysis," *IEEE Transactions on Image Processing*, Vol. 12, No. 8, pp. 962-976, 2003.

- [20] A. Elgammal and C. S. Lee, "Inferring 3D body pose from silhouettes using activity manifold learning," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 681-688, 2004.
- [21] W. Freeman, K. Tanaka, J. Ohta, and K. Kyuma, "Computer vision for computer games," *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 100-105, 1996.
- [22] A. Galata, N. Johnson, and D. Hogg, "Learning variable-length Markov models of behavior," *Computer Vision and Image Understanding*, Vol. 81, No. 3, pp. 398-413, 2001.
- [23] D. M. Gavrila, "The visual analysis of human movement: a survey," *Computer Vision and Image Understanding*, Vol. 73, No. 1, pp. 82-98, 1999.
- [24] P. Guttorp, *Stochastic Modeling of Scientific Data*, London: Chapman and Hall/CRC, 1995.
- [25] I. Guyon and F. Pereira, "Design of a linguistic postprocessor using variable memory length Markov models," *Proceedings of International Conference on Document Analysis and Recognition*, pp. 454-457, Montréal, Canada, 1995.
- [26] 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, Vol. 22, No. 8, 2000.

[27] J. W. Hsieh, Y. T. Hsu, H. Y. Mark Liao and C. C. Chen, "Video-based human movement analysis and its application to surveillance systems," *IEEE Transactions on Multimedia*, Vol. 10, pp. 372- 384, 2008.

[28] J. E. Hunter, D. M. Wilkes, D. T. Levin, C. Heaton, and M. M. Saylor, "Autonomous segmentation of human action for behaviour analysis," *Proceedings of IEEE International Conference on Development and Learning*, Monterey, California, Aug. 2008.

[29] A. K. Jain, M. N. Murthy, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, Vol. 31, pp. 264- 323, 1999.

[30] F. Jelinek, *Statistical Methods for Speech Recognition*, Cambridge, Mass.: MIT Press, 1998.

[31] I. T. Jolliffe, *Principal Component Analysis*. Springer, 1989.

[32] R. Kuhn and R. De Mori, "A cache-based natural language model for speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 6, pp. 570- 583, 1990.

[33] Martin H. C. Law and Anil K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 3, 2006.

- [34] H. Li, S. Lin, Y. Zhang, and K. Tao, "Automatic video-based analysis of athlete action," *Proceedings of IEEE International Conference on Image Analysis and Processing*, pp. 205- 210, Modena, Italy, Sep. 2007.
- [35] Y. Li, S. Ma, and H. Lu, "Human posture recognition using multi-scale morphological method and Kalman motion estimation," *Proceedings of IEEE International Conference on Pattern Recognition*, pp. 175-177, 1998.
- [36] T. Lin and H. Zha, "Riemannian manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 5, 2008.
- [37] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1- 8, Minneapolis, Minnesota, June 2007.
- [38] H. Meng, N. Pears, and C. Bailey, "A human action recognition system for embedded computer vision application," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, June 2007.
- [39] H. Miyamori and S. Iisaku, "Video annotation for content-based retrieval using human behavior analysis and domain knowledge," *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 320- 325, Grenoble, France, 2000.

- [40] T. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, Vol. 81, No. 3, pp. 231- 268, 2001.
- [41] V. I. Morariu and O. I. Camps, "Modeling correspondences for multi-camera tracking using nonlinear manifold learning and target dynamics," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 545- 552, 2006.
- [42] T. Nakata, "Temporal segmentation and recognition of body motion data based on inter-limb correlation analysis," *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007.
- [43] C. G. Nevill-Manning and I. H. Witten, "On-line and off-line heuristics for inferring hierarchies of repetitions in sequence," *Proceedings of the IEEE*, Vol. 88, No. 11, 2000.
- [44] A. S. Ogale, A. Karapurkar, and Y. Aloimonos, "View-invariant modeling and recognition of human actions using grammars," *Workshop on Dynamical Vision at ICCV*, Beijing, China, 2005.
- [45] J. Ohya, "Analysis of human behaviors by computer vision based approaches," *Proceedings of IEEE International Conference on Multimedia and Expo*, Vol. 1, pp. 913- 916, Lusanne, Switzerland, Aug. 2002.
- [46] J. Park, S. Park, and J. K. Aggarwal, "Model-based human motion tracking

- and behavior recognition using hierarchical finite state automata,” *Proceedings of International Conference on computational Science and Its Applications*, pp. 311- 320, Assisi, Italy, 2004.
- [47] R. Plamondon and S. N. Srihari, “Online and off-line handwriting recognition: a comprehensive survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, pp. 63- 84, 2000.
- [48] Y. Qiao and M. Yasuhara, “Affine invariant dynamic time warping and its applications to online rotated handwriting recognition,” *Proceedings of the IEEE International Conference on Pattern Recognition*, Vol. 2, pp. 905- 908, 2006.
- [49] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, Vol. 77, No. 2, 1989.
- [50] M. M. Rahman and S. Ishikawa, “Robust appearance-based human action recognition,” *Proceedings of IEEE International Conference on Pattern Recognition*, Vol. 3, pp. 165- 168, Cambridge, UK, Aug. 2004.
- [51] N. Rane and S. Birchfield, “Isomap tracking with particle filtering,” *Proceedings of the IEEE International Conference on Image Processing*, Vol. 2, pp. 513- 516, 2007.
- [52] D. Ron, Y. Singer, and N. Tishby, “The power of amnesia,” *Advances in*

- Neural Information Processing Systems*, pp. 176- 183, Morgan Kauffmann, New York, 1994.
- [53] R. Rosenfeld, “Two decades of statistical language modeling: where do we go from here,” *Proceedings of the IEEE*, Vol. 88, No. 8, pp. 1270- 1278, 2000.
- [54] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, Vol. 290, pp. 2323- 2326, 2000.
- [55] R. Sharma, V. I. Pavlović, and T. S. Huang, “Toward multimodal human-computer interface,” *Proceedings of the IEEE*, Vol. 86, No. 5, pp. 853- 869, 1998.
- [56] D. Shen and H. H. S. Ip, “Discriminative wavelet shape descriptors for recognition of 2-D patterns,” *Pattern Recognition*, Vol. 32, pp. 151- 165, 1999.
- [57] C. W. Su, H. Y. Mark Liao, H. R. Tyan, C. W. Lin, D. Y. Chen, and K. C. Fan, “Motion flow-based video retrieval,” *IEEE Transactions on Multimedia*, Vol. 9, No. 6, pp. 1193- 1201, 2007.
- [58] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, Vol. 290, pp. 2319- 2323, 2000.
- [59] P. K. Turaga, A. Veeraraghavan, and R. Chellappa, “From videos to verbs:

- mining videos for activities using a cascade of dynamical systems,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1- 8, 2007.
- [60] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71- 86, 1991.
- [61] A. Vinciarelli, S. Bengio, and H. Bunke, “Offline recognition of unconstrained handwritten texts using HMMs and statistical language models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 6, pp. 709- 720, 2004.
- [62] L. Wang and D. Suter, “Learning and matching of dynamic shape manifolds for human action recognition,” *IEEE Transactions on Image Processing*, Vol. 16, No. 6, 2007.
- [63] L. Wang and D. Suter, “Informative shape representations for human action recognition,” *Proceedings of the IEEE International Conference on Pattern Recognition*, Vol. 2, pp. 1266- 1269, 2006.
- [64] L. Wang, W. Hu, and T. Tan, “Recent developments in human motion analysis,” *Pattern Recognition*, Vol. 36, No. 3, pp. 585- 601, 2003.
- [65] T. S. Wang, H. Y. Shum, Y. Q. Xu, and N. N. Zheng, “Unsupervised analysis of human gestures,” *Proceedings of the IEEE Pacific-Rim Conference on Multimedia*, pp. 174- 181, 2001.

- [66] N. Werghi, "A discriminative 3D wavelet-based descriptors: application to the recognition of human body postures," *Pattern Recognition Letters*, Vol. 26, pp. 663- 677, 2005.
- [67] 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*, Vol. 19, No. 7, pp. 780- 785, 1997.
- [68] M. Yazdi, A. B. Albu, and R. Bergevin, "Morphological analysis of spatio-temporal patterns for the segmentation of cyclic human activities," *Proceedings of IEEE International Conference on Pattern Recognition*, Vol. 4, pp. 240- 243, Cambridge, UK, Aug. 2004.
- [69] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 379- 385, 1992.
- [70] J. Yang, Y. Xu, and C. S. Chen, "Human action learning via hidden Markov model," *IEEE Transactions on System, Man, and Cybernetics*, Vol. 27, No. 1, pp. 34- 44, 1997.
- [71] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Transactions on Information Systems*, Vol. 22, No. 2, pp. 179- 214, 2004.

- [72] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 819- 826, 2004.
- [73] http://en.wikipedia.org/wiki/Statistical_significance.



Publication List

A. Journal Paper

1. **Yu-Ming Liang**, Sheng-Wen Shih, Arthur Chun-Chieh Shih, Hong-Yuan Mark Liao, and Cheng-chung Lin, "Learning Atomic Human Actions Using Variable-Length Markov Models," *IEEE Transaction on Systems, Man, and Cybernetics, PartB: Cybernetics*, Vol. 39, No. 1, 2009.

B. Conference Papers

1. **Yu-Ming Liang**, Sheng-Wen Shih, Arthur Chun-Chieh Shih, Hong-Yuan Mark Liao, and Cheng-chung Lin, "Unsupervised Analysis of Human Behavior Based on Manifold Learning," *IEEE International Symposium on Circuits and Systems*, Taipei, Taiwan, May 2009.
2. **Yu-Ming Liang**, Sheng-Wen Shih, Arthur Chun-Chieh Shih, Hong-Yuan Mark Liao, and Cheng-chung Lin, "A Language Modeling Approach to Atomic Human Action Recognition," *IEEE Workshop on Multimedia Signal Processing*, pp. 288-291, Chania, Crete, Greece, Oct. 2007.
3. **Yu-Ming Liang**, Sheng-Wen Shih, Arthur Chun-Chieh Shih, and Hong-Yuan Mark Liao, "Understanding Human Behavior Using a Language Modeling Approach," *IEEE International Conference on Intelligent Information Hiding and Multimedia*, pp. 331-334, Pasadena, CA, USA, Dec. 2006.

C. Other Publications

C1. Journal Papers

1. W. L. Hsu, H. R. Tyan, **Y. M. Liang**, B. S. Jeng, and K. C. Fan, "Real-Time Vehicle Tracking on a Highway," *Journal of Information Science and Engineering*, Vol. 21, No. 4, pp. 733- 752, 2005.
2. **Y. M. Liang**, H. R. Tyan, S. L. Chang, H. Y. Mark Liao, and S. W. Chen, "Video Stabilization for a Camcorder Mounted on a Moving Vehicle," *IEEE Transaction on Vehicular Technology*, Vol. 53, No. 6, pp. 1636-

1648, 2004.

3. C. J. Pai, H. R. Tyan, **Y. M. Liang**, H. Y. Mark Liao, and S. W. Chen, "Pedestrian Detection and Tracking at Crossroads," *Pattern Recognition*, Vol.37, pp. 1025-1034, 2004.
4. **Y. M. Liang**, S. W. Chen, and H. Y. Mark Liao, "Image Stabilization for ITS Applications," *Journal of Taiwan Normal University, Mathematics, Science and Technology*, Vol. 47, No.1, pp. 87-100, 2002.

C2. Conference Papers

1. **Y. M. Liang**, S. L. Chang, H. Y. M. Liao, and S. W. Chen, "Video Stabilization for an In-Vehicle Vision System," *Proc. of 8th Int'l Symposium on Technologies for the Next Generation Vehicles*, pp. 37-47, Chonnam National University, Gwangju, Korea, Oct. 2005.
2. **Y. M. Liang**, Arthur C. C. Shih, H. R. Tyan, and H. Y. Mark Liao, "Background Modeling Using Phase Space for Day and Night Video Surveillance Systems," *5th IEEE Pacific-Rim Conference on Multimedia*, Lecture Notes in Computer Science, Tokoy, Japan, 2004.
3. **Y. M. Liang**, H. R. Tyan, H. Y. Mark Liao, and S. W. Chen, "Stabilizing Image Sequences Taken by the Camcorder Mounted on a Moving Vehicle," *Proceedings of IEEE 6th International Conference on Intelligent Transportation Systems*, Shanghai, China, Vol. 1, pp. 90-95, 2003.
4. C. J. Pai, H. R. Tyan, **Y. M. Liang**, H. Y. Mark Liao, and S. W. Chen, "Pedestrian Detection and Tracking at Crossroads," *Proceedings of IEEE 10th International Conference on Image Processing*, Barcelona, Spain, Vol. 2, pp. 101-104, 2003.
5. **Y. M. Liang**, S. W. Chen, and H. Y. Mark Liao, "Image Stabilization for ITS Applications," *Proceeding of the 14th IPPR Conf. on CVGIP*, Pingtung, Taiwan, 2001.