ABSTRACT

Title of dissertation: Activity Representation from video Using Statistical Models on Shape Manifolds

Mohamed F. Abdelkader, Doctor of Philosophy, 2010

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

Activity recognition from video data is a key computer vision problem with applications in surveillance, elderly care, etc. This problem is associated with modeling a representative shape which contains significant information about the underlying activity. In this dissertation, we represent several approaches for view-invariant activity recognition via modeling shapes on various shape spaces and Riemannian manifolds.

The first two parts of this dissertation deal with activity modeling and recognition using tracks of landmark feature points. The motion trajectories of points extracted from objects involved in the activity are used to build deformation shape models for each activity, and these models are used for classification and detection of unusual activities. In the first part of the dissertation, these models are represented by the recovered 3D deformation basis shapes corresponding to the activity using a non-rigid structure from motion formulation. We use a theory for estimating the amount of deformation for these models from the visual data. We study the special case of ground plane activities in detail because of its importance in video surveillance applications.

In the second part of the dissertation, we propose to model the activity by learning

an affine invariant deformation subspace representation that captures the space of possible body poses associated with the activity. These subspaces can be viewed as points on a Grassmann manifold. We propose several statistical classification models on Grassmann manifold that capture the statistical variations of the shape data while following the intrinsic Riemannian geometry of these manifolds.

The last part of this dissertation addresses the problem of recognizing human gestures from silhouette images. We represent a human gesture as a temporal sequence of human poses, each characterized by a contour of the associated human silhouette. The shape of a contour is viewed as a point on the shape space of closed curves and, hence, each gesture is characterized and modeled as a trajectory on this shape space. We utilize the Riemannian geometry of this space to propose a template-based and a graphical-based approaches for modeling these trajectories. The two models are designed in such a way to account for the different invariance requirements in gesture recognition, and also capture the statistical variations associated with the contour data.

Activity Representation from Video Using Statistical Models on Shape Manifolds

by

Mohamed F. Abdelkader

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Larry Davis
Professor Min Wu
Professor David Jacob
Dr. Wael Abd-Almageed

To my Parents, wife, and daughter.

Acknowledgments

I would like to express my deepest gratitude first to God for guiding me to my goal and then to all the people who contributed to make this dissertation a reality.

First and foremost, I would like to thank my dissertation advisor, Prof. Rama Chellappa for his wise guidance and mentoring during the course of my graduate school. His dedication to his research and his students will always be a great source of inspiration and motivation for me through out the rest of my career. I will always be proud that I have his name on my dissertation and papers.

Through out my graduate experience, I was fortunate to have a chance to work with some great researchers who have enriched my experience and contributed a lot to this dissertation. My collaboration with Prof Amit Roy-Chowdhury, and Prof Anuj Srivastava has shaped up a lot of my research directions and ideas. The outcomes of this collaboration occupy two chapters of this dissertation and several publications. A special gratitude to Dr. Wael Abd-Almageed for working closely with me on my research ideas and for long hours of very fruitful discussions both within and outside the scope of this dissertation.

I want to thank all my fellow former and current graduate students at the computer vision laboratory and Prof Chellappa research group for making my everyday life a pleasant experience. A special mention to Aswin Sankaranarayanan, Ashok Veeraraghavan, Mahesh Ramachandran, James Sherman, Vishal Patel, Dikpal Reddy, Pavan Turaga and Nazre Batool for being great office mates and for hours of fruitful discussions.

I would also like to acknowledge help and support from several staff members who

make it possible for us to find our ways through the administrative jungle - Janice Perrone(CfAR), Maria Hoo (ECE), the UMIACS staff (Arlene Schenk, Yerty Valenzuela, Edna Walker) and the UMIACS computing staff.

A special thanks to all of my dear friends and their families in the masjid Al-Tauba community for making my and my family stay in College Park enjoyable and full of great memories. A special mention to Karim, Kommal, Ahmed, Amr, and Tarek and the families of Walaa, Hossam, Hussein, Hatem, and Islam.

I would like to express my sincere appreciation to my family. My parents, for teaching me everything that I rely on in my life, for helping me decide my way and pursuing it with passion, for supporting me along the road and being always there for me, and finally for flying five thousands miles just to be next to me during my dissertation defense. My brother and sisters, for being there all the time and specially for helping me to go through the tough times. My wife Alaa and my daughter Shahd, for taking care of me and providing me the support and atmosphere to succeed, for giving me many sweet moments, and for relieving all my stress with their precious smiles. My life would have never been the same without you.

I don't think that I can find enough words to acknowledge the rule of my parents, wife, and daughter in this dissertation. So, I dedicate this dissertation to them.

Table of Contents

List of Figures

Chapter 1

Introduction

Video understanding has been one of the ultimate goals of the research efforts in computer vision and pattern recognition domains for a long period of time. The ability to describe what is happening in the video is an important problem for so many applications like surveillance systems, human computer interaction, and multimedia analysis and indexing. Activity representation serves as a key component in the path to achieve this goal. In this dissertation, we address the problem of modeling and recognition of human activities in video. We propose several statistical models on various shape spaces and Riemannian manifolds to achieve a robust representation of various individual activities.

Existing literature in human movement visual analysis (see reviews [4, 39, 51, 73]) uses the terms gesture, and action interchangeably to refer to sequences of human poses corresponding to different activities. However, there are slight variations among these terms. Gesture recognition from video refers to the problem of modeling and recognizing full body gestures performed by an individual in the form of a sequence of body poses and captured by a video camera. These gestures are typically used to communicate certain control commands and requests to a machine equipped with vision capabilities. This scenario usually arises in applications such as human computer interaction and robotics. On the other hand, action recognition refer to the more general case of modeling and recognition of different human actions- such as walking, running, jumping, etc, performed under

different scenarios and conditions. This problem is more prominent in applications such as smart surveillance and media indexing. The main difference between the two problems is that in the former the human subject can be more cooperating, which reduces the need for building view-invariance into the models. The models proposed in this dissertation can be used for either of the two problems. This is demonstrated by using different experimental data sets for gesture and action recognition. We will use the term activity to refer to both gesture and simple action throughout the dissertation.

In many situations, the problem of activity modeling is associated with modeling a representative shape which contains significant information about the underlying activity. This can range from the shape of the silhouette of a person performing an action to the trajectory of the person or a part of his body while preforming the action. The human visual system has a unique ability to recover the underlying activity form the sparse shape data. This ability was first investigated by Johansson [54] and he denoted it the term "biological motion perception".

In this dissertation we hypothesize that modeling the dynamic shape variation associated with activities can play a crucial rule in classifying different activities. However, several challenges arise while using shape models for activity recognition. One of these challenges is the need to account for different variations in activity performance and acquisition. Invariance to camera view-point, execution rate, and anthropometric variations stands as a critical requirement in activity recognition models.

In all of these situations, several factors suggest the use of shape theory [60, 29] to provides powerful methods for representing these shapes. One of these factors is the geometric nature of the shape features, where ideas from differential geometry such as

quotient spaces and equivalence classes can provide efficient tools to computationally characterize the different variation factors among different shapes. These spaces are also characterized by a rich mathematical and statistical theory in the area of differential geometry and shape spaces.

1.1 Motivation

In this dissertation we will present several approaches for modeling human activities by building models on the Riemannian shape manifolds. We will address different shape spaces tailored to applications.

1.1.1 Activity representation Using 3D Shape Models

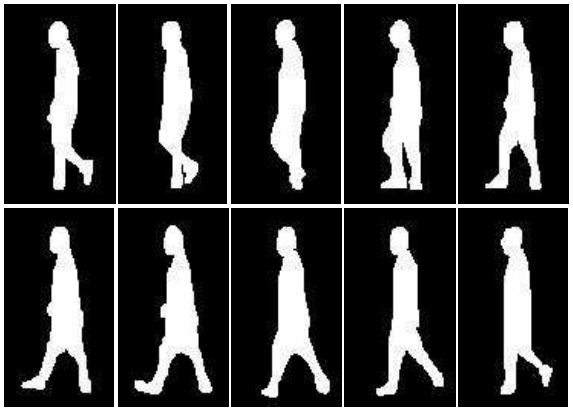
In applications with relatively high quality videos, landmark feature points can be located and tracked with high accuracy. We propose a framework for recognizing activities by first extracting the trajectories of the various points taking part in the activity, followed by a non-rigid 3D shape model fitted to the trajectories. It is based on the empirical observation that many activities have an associated structure and a dynamical model.

Consider, as an example, the set of images of a walking person in Figure 1.1(a) (obtained from the USF database for the Gait Challenge problem [81]). The binary representation clearly shows the change in shape of the body for one complete walk cycle. The person in this figure is free to move his/her hands and feet any way he/she likes. However, this random movement does not constitute the activity of walking. For humans to perceive and appreciate the walk, the different parts of the body have to move in a certain

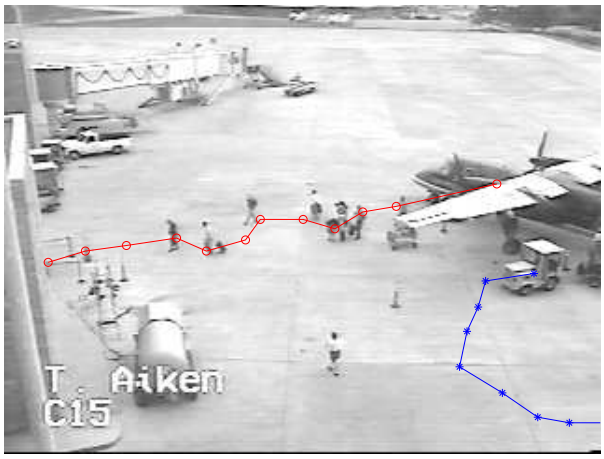
synchronized manner. In mathematical terms, this is equivalent to modeling the walk by the deformations in the shape of the body of the person. Similar observations can be made for other activities performed by a single human, e.g. dancing, jogging, sitting, etc.

An analogous example can be provided for an activity involving a group of people. Consider people getting off a plane and walking to the terminal, where there is no jet-bridge to constrain the path of the passengers (Figure 1.1(b)). Every person after disembarking, is free to move as he/she likes. However, this does not constitute the activity of people getting off a plane and heading to the terminal. The activity here is comprised of people walking along a path that leads to the terminal. Again, we see that the activity can be modeled by the shape of the trajectories taken by the passengers. Using deformable shape models is a higher level abstraction of the individual trajectories and provides a method of analyzing all the points of interest together, thus modeling their interactions in a very elegant way.

Not only is the activity represented by a deformable shape sequence, the amount of deformation is different for different activities. For example, it is reasonable to say that the shape of the human body while dancing is usually more deformable than during walking, which is more deformable than when standing still. Since it is possible for the human observer to roughly infer the degree of deformability based on the contents of the video sequence, the information about how deformable a shape is must be contained in the sequence itself. We will use this intuitive notion to quantify the deformability of a shape sequence from a set of tracked points on the object. In our activity representation model, a deformable shape is represented as a linear combination of rigid basis shapes [107]. The deformability index provides a theoretical framework for estimating the required number



(a)



(b)

Figure 1.1: Two examples of activities: (a) the binary silhouette of a walking person, and (b) people disembarking from an airplane. It is clear that both of these activities can be represented by a deformable shape models by using the body contour in (a) and the passenger/vehicle motion paths in (b).

of basis shapes.

1.1.2 Affine-invariant Activity Recognition on the Grassmann Manifold of Deformation Subspaces

In the second part of this dissertation, we propose to use statistical models on the Grassmann manifold for affine invariant action recognition. Given the 2D locations of different landmark points on the body, captured from arbitrary camera viewpoints, we compute an affine-invariant subspace representation of this action. This subspace representation captures the underlying deformation modes associated with the action under a linear basis deformation model similar to the model usually used in non-rigid structure from motion literature (NRSFM). We apply statistical classification methods on the Grassmann manifold for action recognition.

This approach has the advantage that it does not need a complete recovery of the

actual 3D shape of the subject while performing the action, which eliminates the need to solve for the metric upgrade in the NRSFM formulation. It also builds affine-invariance into the learnt representation, which makes the model invariant to both viewpoint and body type variations. An invariance to the temporal rate of execution of the action is also inherent in this model, since the classification is performed on the view-invariant subspace of modes of deformation rather than the actual temporal trajectory within that subspace.

1.1.3 Gesture and Action Recognition via Modeling Trajectories on Riemannian manifolds

The major challenge of using landmark tracking data is that it requires highly accurate low-level processing tasks such as tracking of interest points. This turns out to be very hard to achieve in certain recognition scenarios because of fast articulation, self occlusion, and different resolution levels that are encountered in different applications.

In order to overcome this limitation, silhouette-based approaches have been receiving increasing attention recently [13, 119, 123]. These approaches focus on the use of the shape of the binary silhouette of the human body as a feature for gesture recognition. They rely on the observation that most human gestures can be recognized using only the shape of the outer contour of the body. The most important advantage of these features is that they are easy to extract from the raw video frames using object localization and background subtraction algorithms, which are low-level processing tasks and relatively high accuracy can be achieved in these tasks under different conditions.

In the third part of this dissertation, we explore the use of shape analysis on closed curve manifolds for human actions and gesture recognition. Our approach falls into the category of the silhouette-based approaches described earlier. Each silhouette is represented by a planar closed curve corresponding to the contour of this silhouette, and we are interested in evolving shapes of these curves during actions and gestures. We use a recent approach proposed for shape analysis [56, 57, 101], that uses differential geometric tools on the shape spaces of closed curves. While there are several ways to analyze shapes of closed curves, an elastic analysis of the parameterized curves is particularly appropriate in this application. This is because: (1) the elastic matching of curves allows nonlinear registration and improved matching of features (e.g. body parts) across silhouettes, (2) this method uses a square-root representation under which the elastic metric reduces to the standard $\mathbb{L}^2$ metric and thus simplifies the analysis, and (3) under this metric the re-parameterizations of curves do not change Riemannian distances between them and thus help remove the parametrization variability from the analysis. Furthermore, such geometric approaches are useful because they allow us to perform intrinsic statistical analysis tasks, such as shape modeling and clustering, on such Riemannian spaces [100].

Using a square-root representation of contours, each human gesture is transferred into a sequence of points on the shape space of closed curves. Thus, the problem of action recognition becomes a problem of modeling and comparing dynamical trajectories on the shape space. We propose two different approaches to model these trajectories.

In the first approach, we propose a template-based approach to learn a unique template trajectory representing each gesture. One of the main challenges in template-based methods is to account for variations in temporal execution rate. To deal with this problem,

we use a modified version of the Dynamic Time Warping (DTW) algorithm to learn the warping functions between the different realizations of each gesture. We use the geodesic distances on the shape space to match different points on the trajectories in order to learn the warping functions. An iterative approach is then used to learn a mean trajectory on the shape space and to compute the temporal warping functions.

In the second approach, we utilize the geometry of the shape space more efficiently in order to cope with the different variations within each gesture caused by changes in execution style, body shape and noise. Each gesture is modeled as a Markov model to represent the transition among different clusters on the shape space of closed curve. We learn these models by decoupling the problem into two stages. In the first stage, we cluster the individual silhouette shapes using the Affinity Propagation (AP) clustering technique [38], and build statistical model of variation within each cluster. In the second stage, a hidden Markov Model (HMM) is used to learn the transition between different clusters for each gesture.

1.2 Contributions

This dissertation makes the following contributions:

1. **Activity representation using 3D shape Models:** We propose an approach for activity representation and recognition based on extracting the 3D shapes generated by the activity. We use the 3D deformable shape model for characterizing the objects corresponding to each activity. We study the special case of ground plane activity trajectory in more detail as an important application because of its impor-

tance in surveillance scenarios.

2. **Affine Invariant Activity Recognition on Grassmann manifold of Deformation Subspaces:** We model the problem of affine-invariant activity recognition as a classification problem on the space of deformation subspaces of landmark data. We study and propose several statistical models on the Grassmann manifold to solve this activity classification problem. We propose a multi-scale Bayesian formulation to deal with the variable degree of deformation for different actions.

3. **Gesture and Action Recognition via Modeling Trajectories on Riemannian manifolds:** We pose the problem of gesture and action recognition as one of classifying the trajectories on a Riemannian shape space of closed curves. We study the differential geometry of this space, and propose a template-based model and a Markovian graphical model for modeling the time series data of points on this manifold. These models were designed to fully adhere to the geometry of the manifold and to model the statistical variation of the data. Finally, we provide a comprehensive set of experimental analysis of the proposed models on two different datasets for gesture and action recognition.

1.3 Organization

This rest of this dissertation is organized as follows. Chapter 2 reviews some of the literature related work and introduces some of the needed basis mathematical background in Riemannian geometry. Chapter 3 discusses the use of 3D shape models for activity recognition. Chapter 4 represents activity recognition via modeling deformability spaces

using statistical models on the Grassmann manifold. Chapter 5 deals with silhouette-based activity recognition via modeling trajectories on the Riemannian manifold of closed curves. Finally, we discuss future directions of research in chapter 6.

Chapter 2

Related Work

In this chapter we review some of the literature related to several aspects of this dissertation. We will also review some basic mathematical results from Riemannian geometry that are used in the discussion in later chapters.

2.1 Literature Review

The problem of action and activity modeling from visual data has received a lot of attention in the literature in the last few decades, resulting in many publications scattered among different communities. Several survey papers [3, 4, 39, 72, 73] have tried to group and analyze the existing body of work. We refer the reader to these review papers for a complete overview of the related work in the field.

One of the earliest efforts was the survey paper by Aggarwal and Cai [3]. In this paper, the authors present a survey of the work on three major areas related to human action and activities. These areas are segmentation of human body parts, human tracking across video sequence and finally activity recognition. In a recent review, Aggarwal and Park [4] concentrate on the high-level processing tasks involved in action and activity modeling. They discuss different aspects such as the level of detail required, the different human models, recognition approaches and high-level recognition schemes. Gavrila [39] presents a survey that focuses mainly on whole body and hand motion actions. He clas-

sifies the approaches into 2D and 3D approaches, and gives a nice survey of the various application domains for human motion analysis. Moeslund and others published two survey papers [72, 73] that cover the work on motion capture analysis before the year 2000 and from 2000-2006 respectively. They use a consistent functional taxonomy that divides the work into the areas of initialization, tracking, pose estimation and finally recognition. The reader is referred to these review papers for a complete overview of the related work in the field. Below, we give a brief review for some approaches in three different areas that are of most relevance to this dissertation.

2.1.1 Activity Representation and Recognition

Most of the early work on activity representation comes from the field of Artificial Intelligence (AI) [110, 75]. More recent work comes from the fields of image understanding and visual surveillance, employing formalisms like hidden Markov models (HMMs), logic programming and stochastic grammars [74, 64, 28, 19, 25, 88, 122, 102, 128]. A method for visual surveillance using a "forest of sensors" was proposed in [44]. Many uncertainty-reasoning models have been actively pursued in the AI and image understanding literature, including belief networks [79, 53, 86], Dempster-Shafer theory [94], dynamic Bayesian networks [6, 52] and Bayesian inference [48].

Video data in its raw format consists of a sequence of pixel intensity values organized in a spatial and temporal order. This results in a massive amount of data and a very high dimensional data vectors. However, these data vectors are highly correlated as they reflect a small degree of freedom that controls the generation of the underlying action and

activity. For this reason, an important processing step in activity recognition is the feature extraction stage, which aims at extracting a more concise representation for the video data in order to use it in modeling and recognizing actions and activities in video.

Among the different features used for activity recognition, silhouette-based methods have received the most attention in the literature. Wang and Suter recently proposed two approaches [124, 123] for modeling motion subspaces associated with human actions represented by silhouettes. In the first approach [124], they represent the silhouettes using block based features, where each block is represented by the normalized value of the intensity values within the block. They then use the kernel PCA (KPCA) technique to learn a nonlinear subspace of the action data. The dynamics of the action is then modeled using a discriminative factorial conditional random field model (FCRF). In the second approach [123], they use features based on a distance transform (DT) and use local preserving projections (LPP) to project the feature data onto a lower dimensional manifold. They use a simpler model for recognition based on comparing the trajectories on the embedding space. In both approaches, the main assumption was that the features will have an inherently low-dimensional structure, and that data-driven nonlinear dimensionality techniques will successfully capture this structure.

Another related class of approaches for silhouette-based action recognition is called exemplar-based modeling [32, 127], where the different features corresponding to gestures are clustered into a set of exemplars or key poses, and the dynamics of the action are learnt as a set of transitions between these exemplars. A recent example of these approaches is [68], where each gesture is represented using a sequence of shape-motion prototypes. The training data is used to build a prototype tree in the joint shape and

motion space via hierarchical k-means clustering. A k-NN classifier is used to classify gestures based on the frame to prototype distance of the test sequence. Exemplar-based hidden Markov models [37, 108] were used in [32] to model body gestures using Chamfer distance as a matching measure between different poses. The same tool was used in [127] for view-invariant modeling of actions using 3D exemplars. Our technique for learning the graphical model for trajectories presented in Section 5.3, comes under this category, if we consider the different clusters as exemplar points on the shape space. However, we use the Riemannian structure of the manifold to build statistical models around these exemplars. This enables us to perform the statistical inference task more efficiently than using only the exemplars.

2.1.2 Riemannian Manifolds in Computer Vision

The geometric nature of many vision problems leads to features and exemplars that lies on a certain curved space that has a Riemannian structure, leading to the applicability of a wide variety of tools from differential geometry and Riemannian manifold analysis. These tools have been used in several vision problems. For example, a covariance descriptor was proposed in [113] for region-based shape description. This descriptor lies on the space of symmetric positive definite matrices. The Riemannian geometry of this space was later used to build models for object detection and tracking [114]. In [115], a Riemannian classification approach was used for pedestrian detection on this manifold. They used a modified version of the Logitboost approach, where the weak learners are trained directly on that manifold.

Another special manifold, the Grassmann manifold, was used in [9] for affine invariant shape clustering. Statistical models on this manifold were used in [112] to build parametric and non-parametric distributions for several other vision applications like activity recognition, video-based face recognition. Trajectories on this manifold were later used in [111] to represent locally time-invariant dynamical models for action recognition.

2.1.3 Shape Spaces and Activity Recognition

Among the different geometric manifolds that are of interest in computer vision applications, shape spaces have received the most attention in the literature and have been used for shape detection, tracking and analysis that are invariant to different shape-preserving transformation groups. The work in this area can be classified into two main categories. The first category is based on landmark-based analysis, where the shape of the object is represented by a discrete set of landmarks extracted from the object contour [60, 29]. This formulation was used to represent shape spaces that are invariant under transformation groups such as similarity [59], affine [78], and projective transformation [41]. The limitation with this class of approaches is its dependence on automatic detection and tracking of landmark points. A simple landmark-based approach is Active Shape Model (ASM) [22], that assumes that shape belongs to the Euclidean space and uses principal component analysis (PCA) of the landmark location to model the shape variations within that space. Although this approach has found many applications because of its simplicity, it ignores the nonlinear geometry of the shape space which limits its representative power under different geometric transformations. The second category

of shape spaces can be linked to Grenander's formulation [42], where shapes are modeled as points on infinite-dimensional manifolds, and the variations between the shapes are modeled by the actions of Lie-groups (diffeomorphisms) on these manifolds. This class of approaches avoid the problems associated with landmark tracking. However it suffers from computational cost of such high-dimensional models.

Shape spaces have also been used for activity recognition in several computer vision publications. Kendall's statistical shape theory was used to model the interactions of a group of people and objects in [116], as well as the motion of individuals [118]. In both approaches, assuming the stationarity of the shape sequence of an activity, the authors project the sequence of shape points into the tangent plane at the mean shape. The dynamic of the projected points is then modeled using regular vector space approaches on this single tangent plane. In a more recent work [24], a non stationary model was proposed. It builds an auto regressive (AR) process on the shape space represented by a sequence of tangent planes. They use an alignment step to align the tangent planes at two consecutive time steps. The proposed model was used for tracking of the shape using a particle filter algorithm and for action synthesis. There was no experimental results for using this approach for action classification and recognition.

A method for representation of human activities based on space curves of joint angles and torso location and attitude was proposed in [20]. In [85], the authors proposed an activity recognition algorithm using dynamic instants and intervals as view-invariant features and the final matching of trajectories was conducted using a rank constraint on the 2D shapes. In [77], each human action was represented by a set of 3D curves which are quasi-invariant to the viewing direction. In [55] and [76] the motion trajectories of an

object are described as a sequence of flow vectors, and neural networks are used to learn the distribution of these sequences. In [21], a wavelet transform was used to decompose the raw trajectory into components of different scales, and the different subtrajectories are matched against a data base to recognize the activity. Although landmark-based shape space methods have proven successful in several vision problems, the performance usually relies on accurate detection and tracking of key landmark points. This turns out to be a very challenging task especially under rapid motion and self-occlusion, which are common in gesture recognition problems. For this reason, instead of using landmark-based shape representation and space, our approach models the shape of the contour at each frame using a closed curve representation.

Recently, there has been a growing interest in building mathematical representations and metrics for modeling the shapes of closed curves. We will employ one of the recent approaches [71], which uses differential geometry tools for analyzing the space of closed curves using elastic string models. Similar ideas have also been presented in [132, 132]. Using Riemannian analysis of such space, statistical models and efficient methods were built to perform tasks such as shape modeling and clustering [100]. Later, the same methods were combined using a square-root representation [56, 101] to construct a shape space of closed curves in $\mathbb{R}^2$ and to compute geodesics between 2D and 3D closed curves. Invariance to different shape preserving geometric transformation and re-parametrization were incorporated into the computation of geodesics between shapes and subsequently used for shape analysis of curves in [57].

2.2 Mathematical Background

2.2.1 Overview of Riemannian Geometry

We will briefly review the basics of Riemannian geometry [27]. A topological manifold is a set that is, among other things, locally Euclidean. This means that for an n-dimensional manifold $\mathcal{M}$, there exists a set of neighborhoods that cover $\mathcal{M}$ and that are homeomorphic to corresponding open sets in $\mathbb{R}^n$. If these local mappings are diffeomorphic, and the different neighborhoods are (maximally) smoothly compatible, then $\mathcal{M}$ is a differentiable manifold. Figure 2.1 shows an example of a two-dimensional manifold in $\mathbb{R}^3$. The (infinite-dimensional) Hilbert manifold is a set that is locally diffeomorphic to the (infinite-dimensional) Hilbert space. The tangent space $T_x\mathcal{M}$ at each point $x \in \mathcal{M}$ is the vector space that contains the velocity vectors of all the differentiable curves on $\mathcal{M}$ passing through x, at x. A Riemannian metric on a manifold $\mathcal{M}$ is an inner product $\langle ., . \rangle$ on the tangent space $T_x\mathcal{M}$ that varies smoothly with x. The norm of the vector $v \in T_x\mathcal{M}$ is given by $||v|| = \langle v, v \rangle^{(1/2)}$. If $\alpha : [0, 1] \rightarrow \mathcal{M}$ is a differentiable path in $\mathcal{M}$, then its length is given by $L[\alpha] = \int_0^1 ||\dot{\alpha}(t)|| \, dt$. The Riemannian distance between two points $x, y \in \mathcal{M}$, denoted as $d(x, y)$, is defined as the minimum length over all paths on the manifold between x and y. A geodesic path is a path that locally minimizes the length between points.

Given a tangent vector $\upsilon \in T_x\mathcal{M}$, there exists a locally unique geodesic, $\gamma_\upsilon(t)$, starting at x with υ as its initial velocity and traveling with constant speed. The Riemannian exponential map, $exp_x : T_x\mathcal{M} \rightarrow \mathcal{M}$ maps a tangent vector υ to a point on the manifold that is reached in unit time by the geodesic $\gamma_\upsilon(t)$. The inverse of exp_x is known as the logarithm map and is denoted by $log_x : \mathcal{M} \rightarrow T_x\mathcal{M}$. Thus, for a point y in the domain of

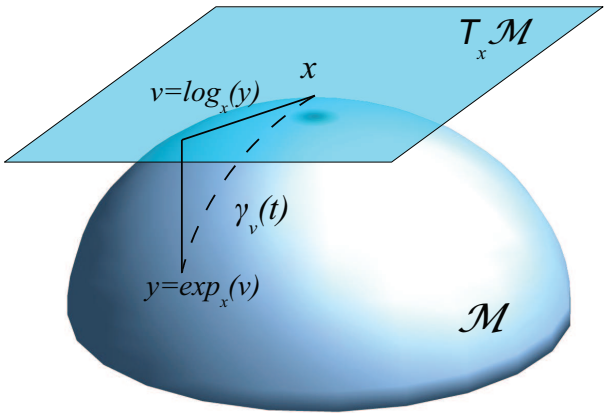


Figure 2.1: A two-dimensional manifold $\mathcal{M}$ in $\mathbb{R}^3$, $T_x\mathcal{M}$ the tangent plane at $x \in \mathcal{M}$, and the exp and log maps relating $x, y \in \mathcal{M}$.

log_x, the geodesic distance between x and y is given by

$$d(x, y) = \|log_x(y)\| . \tag{2.1}$$

As explained in [80, 66], the exponential map realizes a chart called the *the exponential chart*. In this chart, geodesics starting from x are straight lines, and the distance from the development point are preserved. This chart is considered the "most linear" chart of the manifold with respect to the point x.

In order to build efficient and robust models of trajectories on Riemannian manifolds, we need to use some tools for statistical analysis of points on Riemannian manifolds. We start with calculating the intrinsic mean and principal components for a set of points $x_1, \cdots, x_n \in \mathcal{M}$ that lie on a sufficiently small neighborhood on $\mathcal{M}$.

The intrinsic mean or Karcher mean μ is defined as a local minimizer in $\mathcal{M}$ of the sum-of-squared Riemannian distances to each point. Thus, this intrinsic mean is given by

$$\mu = argmin_{x \in \mathcal{M}} \sum_{i=1}^{N} d(x, x_i)^2 \tag{2.2}$$

where $d(.,.)$ is the Riemannian geodesic distance defined in (2.1). A gradient descent

approach was proposed in [80, 66] to solve this minimization problem.

Building high order statistical models directly on Riemannian manifolds is rather difficult to perform. This is mainly due to the non-linearity of these manifolds. So a common approach is to build the statistical model on the tangent plane of the manifold at some reference point, which is a vector space and hence more conventional statistics can be applied. Usually, the reference point is chosen to be the mean point of the set of samples of the distribution. Under the assumption that the data points lie in a small neighborhood about the mean point μ, it was shown in [34] that solving for the principal geodesic components, the equivalent of principal component vectors in Euclidean space, boils down to performing PCA for the tangent vectors $log_{\mu}(x_i) \in T_{\mu}\mathcal{M}$. This method is called principal geodesic analysis (PGA).

Chapter 3

Activity representation Using 3D Shape Models

3.1 Introduction

In many situations, the problem of activity modeling is associated with modeling a representative shape which contains significant information about the underlying activity. This can range from the shape of the silhouette of a person performing an action to the trajectory of the person or a part of his body. However, these shapes are often hard to model because of their deformability and variations under different camera viewing directions.

In all of these situations, shape theory provides powerful methods for representing these shapes [60, 29]. The work in this area is divided between 2D and 3D deformable shape representations. The 2D shape models focus on comparing the similarity of two or more 2D shapes [29, 22, 70, 93, 97]. Two-dimensional representations are usually computationally efficient and there exists a rich mathematical theory using which appropriate algorithms could be designed. Three-dimensional models have received much attention in the past few years. In addition to the higher accuracy provided by these methods, they have the advantage that they can potentially handle variations in camera viewpoints.

3D approaches have been applied in several applications like face recognition [135]. For example in [87], the authors proposed a 3D morphable face model, where each face can be represented as a linear combination of basis shapes and textures. However, the

use of 3D shapes for activity recognition has been much less studied. In many of the 3D approaches, a 2D shape is represented by a finite-dimensional linear combination of 3D basis shapes and a camera projection model relating the 3D and 2D representations [107, 14, 129, 15]. This method has been applied primarily to deformable object modeling and tracking. In [96], actions under different variability factors were modeled as a linear combination of spatio-temporal basis actions. The recognition in this case was performed using the angles between the action subspaces and without explicitly recovering the 3D shape. However, this approach needs sufficient video sequences of the actions under different viewing directions and other forms of variability to learn the space of each action.

In this chapter, we propose an approach for activity representation and recognition based on 3D shapes generated by the activity. We use the 3D deformable shape model for characterizing the objects corresponding to each activity. The underlying hypothesis is that an activity can be represented by deformable shape models that capture the 3D configuration and dynamics of the set of points taking part in the activity. This approach is suitable for representing different activities as shown by experiments in Section 3.4. This idea has also been used for 2D shape-based representation in [119, 117]. We also use a method for estimating the amount of deformation of a shape sequence by deriving a "deformability index" (DI) [91]. Estimation of the DI is non-iterative, does not require selecting an arbitrary threshold and can be done before estimating the 3D structure, which means we can use it as an input to the 3D non-rigid model estimation process. We study the special case of ground plane activity trajectories in more detail because of their importance in surveillance scenarios. The 3D shapes in this special scenario are constrained by the ground-plane which reduces the problem to a 2D shape representation. Our method

in this case has the ability to match the trajectories across different camera view points (which would not be possible using 2D shape modeling methods), and the ability to estimate the number of activities using the DI formulation. Preliminary versions of this work appeared in [90, 89] and a more detailed analysis of the concept of measuring the deformability was presented in [91].

We have tested our approach on different experimental data sets. First we validate the method of DI estimation using motion capture data as well as videos of different human activities. The results show that the DI is in accordance with our intuitive judgment and corroborates certain hypotheses prevailing in human movement analysis studies. Subsequently, we present the results of applying our algorithm to two different applications: view-invariant human activity recognition using 3D models (high-resolution imaging), and detecting anomalies in ground plane surveillance scenario (low-resolution imaging).

This Chapter is organized as follows. Section 3.2 describes the shape-based activity modeling approach along with the special case of ground plane motion trajectories. Section 3.3 presents the method for estimating the DI for a shape sequence. Detailed experimental results are presented in Section 3.4, before providing a chapter summary in Section 3.5.

3.2 Shape Based Activity Models

We propose a framework for recognizing activities by first extracting the trajectories of the various points taking part in the activity, followed by a non-rigid 3D shape model fitted to the trajectories. It is based on the empirical observation that many activities have

an associated structure and an underlying dynamical model.

3.2.1 Estimation of Deformable Shape Models

We hypothesize that each shape sequence can be represented by a linear combination of 3D basis shapes. Mathematically, if we consider the trajectories of P points representing the shape (e.g. landmark points), then the overall configuration of the P points is represented as a linear combination of the basis shapes S_i as

$$S = \sum_{i=1}^{K} l_i S_i, \qquad S, S_i \in \Re^{3\times P}, l \in \Re. \tag{3.1}$$

where l_i represent the weight associated with the basis shape S_i.

The choice of K is determined by quantifying the deformability of the shape sequence and will be studied in detail in Section 3.3. We will assume a weak perspective projection model for the camera.

A number of methods exist in the computer vision literature for estimating the basis shapes. In the factorization paper for structure from motion [104], the authors considered P points tracked across F frames in order to obtain two $F \times P$ matrices $\mathbf{U}$ and $\mathbf{V}$. Each row of $\mathbf{U}$ contains the x-displacements of all the P points for a specific time frame, and each row of $\mathbf{V}$ contains the corresponding y-displacements. It was shown in [104], that for 3D rigid motion and the orthographic camera model, the rank, r, of the concatenation of the rows of the two matrices $\begin{bmatrix}\mathbf{U}\\ \mathbf{V}\end{bmatrix}$ has an upper bound of 3. The rank constraint is derived from the fact that $\begin{bmatrix}\mathbf{U}\\ \mathbf{V}\end{bmatrix}$ can be factored into two matrices $\mathbf{M}_{2F\times r}$ and $\mathbf{S}_{r\times P}$, corresponding to the pose and 3D structure of the scene, respectively. In [107], it was shown that for non-rigid motion, the above method could be extended to obtain a similar rank constraint, but one

that is higher than the bound for the rigid case. We will adopt the method suggested in [107] for computing the basis shapes for each activity. We will outline the basic steps of their approach in order to clarify the notation for the remainder of the paper.

Given F frames of a video sequence with P moving points, we first obtain the trajectories of all these points over the entire video sequence. These P points can be represented in a measurement matrix as

$$\mathbf{W}_{2F\times P} = \begin{bmatrix} u_{1,1} & \cdots & u_{1,P} \\ v_{1,1} & \cdots & v_{1,P} \\ \vdots & \vdots & \vdots \\ u_{F,1} & \cdots & u_{F,P} \\ v_{F,1} & \cdots & v_{F,P} \end{bmatrix}, \tag{3.2}$$

where $u_{f,p}$ represents the x-position of the p^{th} point in the f^{th} frame and $v_{f,p}$ represents the y-position of the same point. Under weak perspective projection, the P points of a configuration in a frame f, are projected onto 2D image points $(u_{f,i}, v_{f,i})$ as

$$\begin{bmatrix} u_{f,1} & \cdots & u_{f,P} \\ v_{f,1} & \cdots & v_{f,P} \end{bmatrix} = \mathbf{R}_f \left(\sum_{i=1}^{K} l_{f,i} S_i \right) + \mathbf{T}_f, \tag{3.3}$$

where,

$$\mathbf{R}_f = \begin{bmatrix} r_{f1} & r_{f2} & r_{f3} \\ r_{f4} & r_{f5} & r_{f6} \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{R}_f^{(1)} \\ \mathbf{R}_f^{(2)} \end{bmatrix}. \tag{3.4}$$

$\mathbf{R}_f$ represents the first two rows of the full 3D camera rotation matrix and $\mathbf{T}_f$ is the camera translation. The translation component can be eliminated by subtracting out the mean of all the 2D points, as in [104]. We now form the measurement matrix $\mathbf{W}$, which is

represented in (3.2), with the means of each of the rows subtracted. The weak perspective scaling factor is implicitly coded in the configuration weights, $\{l_{f,i}\}$.

Using (3.2) and (3.3), it is easy to show that

$$\mathbf{W} = \begin{bmatrix} l_{1,1}\mathbf{R}_1 & \cdots & l_{1,K}\mathbf{R}_1 \\ l_{2,1}\mathbf{R}_2 & \cdots & l_{2,K}\mathbf{R}_2 \\ \vdots & \vdots & \vdots \\ l_{F,1}\mathbf{R}_F & \cdots & l_{F,K}\mathbf{R}_F \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_K \end{bmatrix} \tag{3.5}$$

$$= \mathbf{Q}_{2F\times 3K}.\mathbf{B}_{3K\times P},$$

which is of rank $3K$. The matrix $\mathbf{Q}$ contains the pose for each frame of the video sequence and the weights $l_1, ..., l_K$. The matrix $\mathbf{B}$ contains the basis shapes corresponding to each of the activities. In [107], it was shown that $\mathbf{Q}$ and $\mathbf{B}$ can be obtained using singular value decomposition (SVD), and retaining the top $3K$ singular values, as $\mathbf{W}_{2F\times P} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ and $\mathbf{Q} = \mathbf{U}\mathbf{D}^{\frac{1}{2}}$ and $\mathbf{B} = \mathbf{D}^{\frac{1}{2}}\mathbf{V}^T$. [1]

3.2.2 Special Case: Ground Plane Activities

A special case of activity modeling that often occurs is the case of ground plane activities, which are often encountered in applications such as visual surveillance. In these applications the objects are far away from the camera such that each object can be considered as a point moving on a common plane such as the ground plane of the scene

[1] The solution is unique up to an invertible transformation. Methods have been proposed for obtaining an invertible solution using the physical constraints of the problem. This has been dealt with in detail in previous papers [105, 129]. Although this is important for implementing the method, we will not dwell on it in detail in this dissertation and shall refer the reader to previous works.

under consideration. Because of the importance of such configurations, we study it in more detail and present an approach for using our shape based activity model to represent these ground plane activities. The 3D shapes in this case are reduced to 2D shapes due to the ground plane constraint. The main reason for using our 3D approach (as opposed to a 2D shape matching one) is the ability to match the trajectories across changes of viewpoint.

Our approach for this situation consists of two steps. The first step recovers the ground plane geometry and uses it to remove the projection effects between the trajectories that correspond to the same activity. The second step uses the deformable shape-based activity modeling technique to learn a nominal trajectory that represents all the ground plane trajectories generated by an activity. Since each activity can be represented by one nominal trajectory, we will not need multiple basis shapes for each activity.

3.2.2.1 First Step: Ground-Plane Calibration

Most of the outdoor surveillance systems monitor a ground plane of an area of interest. This area could be the floor of a parking lot, the ground plane of an airport, or any other monitored area. Most of the objects being tracked and monitored are moving on this dominant plane. We use this fact to remove the camera projection effect by recovering the ground plane and projecting all the motion trajectories back onto this ground plane. In other words, we map the motion trajectories measured at the image plane onto the ground plane coordinates to remove these projective effects. Many automatic or semi-automatic methods are available to perform this calibration [134, 109]. As the calibration process

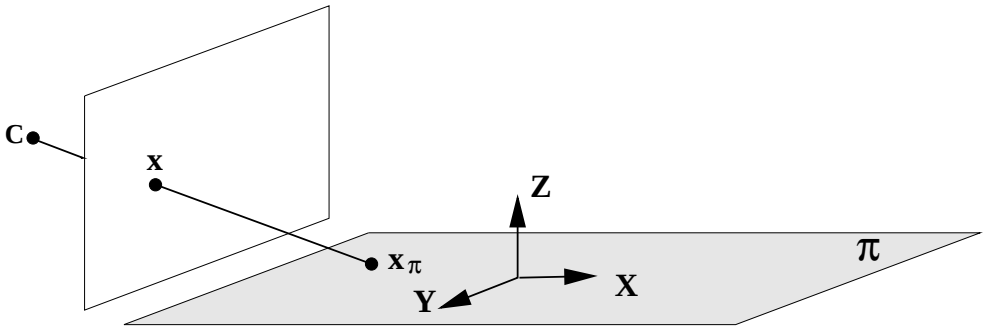


Figure 3.1: Perspective images of points in a plane[45]. The world coordinate system is moved in order to be aligned with the plane π.

needs to be performed only one time because the camera is fixed, we are using the semi-automatic method presented in [67], which is based on using some of the features which are often seen in man-made environments. We will give a brief summary of this method for completeness.

Consider the case of points lying on a world plane π, as shown in Figure 3.1. The mapping between points $\mathbf{X}_\pi = (X, Y, 1)^T$ on the world plane π and their image $\mathbf{x}$ is a general planar homography- a plane to plane projective transformation- of the form $\mathbf{x} = \mathrm{H}\mathbf{X}_\pi$, with H being a 3×3 matrix of rank 3. This projective transformation can be decomposed into a chain of more specialized transformations of the form

$$\mathrm{H} = \mathrm{H_S H_A H_P} \tag{3.6}$$

where $\mathrm{H_S}$, $\mathrm{H_A}$, and $\mathrm{H_P}$ represent similarity, affine and pure-projective transformations, respectively. The recovery of the ground plane up to a similarity is performed in two stages.

Stage 1: From projective to affine This is achieved by determining the pure projective transformation matrix $\mathrm{H_P}$. We note that the inverse of this projective transformation is

also a projective transformation $\hat{\mathrm{H}}_{\mathrm{P}}$, which can be written as

$$\hat{\mathrm{H}}_{\mathrm{P}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix} \tag{3.7}$$

where $\mathbf{l}_\infty = (l_1, l_2, l_3)^T$ is the vanishing line of the plane, defined as the line connecting all the vanishing points for lines lying on the plane.

From (3.7), it is evident that identifying the vanishing line is enough to remove the pure-projective part of the projection. In order to identify the vanishing line, two sets of parallel lines should be identified. Parallel lines are easy to find in man made environments (e.g. parking space markers, curbs, and road lanes).

Stage 2: From affine to metric The second stage of the rectification is the removal of the affine projection. As in the first stage, the inverse affine transformation matrix $\hat{\mathrm{H}}_{\mathrm{A}}$ can be written in the following form

$$\hat{\mathrm{H}}_{\mathrm{A}} = \begin{bmatrix} \frac{1}{\beta} & -\frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

Also, this matrix has two degree of freedoms represented by α and β. These two parameters have a geometric interpretation as representing the circular points, which are a pair of points-at-infinity that are invariant to Euclidean transformations. Once these points are identified, metric properties of the plane are available.

Identifying two affine invariant properties on the ground plane can be sufficient to obtain two constraints on the values of α and β. Each of these constraint is in the form of

a circle. These properties include known angle between lines, equality of two unknown angle, and a known length ratio.

3.2.2.2 Second Step: Learning Trajectories

After recovering the ground plane (i.e. finding the projective $\hat{H}_P$ and affine $\hat{H}_A$ inverse transformations) the motion trajectories of the objects are reprojected to their ground plane coordinates. Having m different trajectories of each activity, the goal is to obtain a nominal trajectory that represents all of these trajectories. We assume that all these trajectories have the same 2D shape up to a similarity transformation (translation, rotation, and scale). This transformation will compensate for the way the activity was performed in the scene. We use the factorization algorithm to obtain the shape of this nominal trajectory from all the motion trajectories.

For a certain activity that we wish to learn, let T_j be the j^{th} ground-plane trajectory of this activity. This trajectory was obtained by tracking an object performing the activity in the image plane over n frames, and projecting these points onto the ground plane as,

$$\begin{aligned} T_j &= \begin{bmatrix} x_{j1} \cdots x_{jn} \\ y_{j1} \cdots y_{jn} \\ 1 \cdots 1 \end{bmatrix} \\ &= \hat{H}_A \hat{H}_P \begin{bmatrix} u_{j1} \cdots u_{jn} \\ v_{j1} \cdots v_{jn} \\ 1 \cdots 1 \end{bmatrix} \end{aligned} \tag{3.9}$$

where u, v are the 2D image plane coordinates, x, y are the ground plane coordinates, and $\hat{H}_P$ and $\hat{H}_A$ are the pure-projective and affine transformations from image to ground

planes respectively.

Assume that, except for a noise term η_j, all the different trajectories correspond to the same 2D nominal trajectory S but have undergone 2D similarity transformations (scale, rotation and translation). Then

$$\begin{aligned} \mathrm{T}_j &= \mathrm{H}_{\mathrm{S}j}\mathrm{S} + \eta_j \\ &= \begin{bmatrix} s_j\cos\theta_j & -s_j\sin\theta_j & t_{xj} \\ s_j\sin\theta_j & s_j\cos\theta_j & t_{yj} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \cdots \tilde{x}_n \\ \tilde{y}_1 \cdots \tilde{y}_n \\ 1 \cdots 1 \end{bmatrix} + \eta_j \end{aligned} \tag{3.10}$$

where $\mathrm{H}_{\mathrm{S}j}$ is the similarity transformation between the j^{th} trajectory and S. This relation can be rewritten in inhomogeneous coordinates as

$$\begin{aligned} \hat{\mathrm{T}}_j &= \begin{bmatrix} s_j\cos\theta_j & -s_j\sin\theta_j \\ s_j\sin\theta_j & s_j\cos\theta_j \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \cdots \tilde{x}_n \\ \tilde{y}_1 \cdots \tilde{y}_n \end{bmatrix} + \begin{bmatrix} t_{xj} \\ t_{yj} \end{bmatrix} + \eta_j \\ &= s_j\mathrm{R}_j\mathrm{S} + \mathbf{t_j} + \eta_j \end{aligned} \tag{3.11}$$

where s_j, R_j and $\mathbf{t}_j$ represent the scale, rotation matrix and translation vector, respectively, between the j^{th} trajectory and the nominal trajectory S.

In order to explore the temporal behavior of the activity trajectories, we divide each trajectory into small segments at different time scales and explore these segments. By applying this time scaling technique, which will be addressed in detail in Section 3.4, we obtain m different trajectories, each with n points. Given these trajectories, we can

construct a measurement matrix of the form

$$\mathrm{W} = \begin{bmatrix} \hat{\mathrm{T}}_1 \\ \hat{\mathrm{T}}_2 \\ \vdots \\ \hat{\mathrm{T}}_m \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ y_{11} & \cdots & y_{1n} \\ \vdots & & \vdots \\ x_{m1} & \cdots & x_{mn} \\ y_{m1} & \cdots & y_{mn} \end{bmatrix}_{2m \times n} \tag{3.12}$$

As before, we subtract the mean of each row to remove the translation effect. Substituting from (3.11), the measurement matrix can be written as

$$\begin{aligned} \mathrm{W} &= \begin{bmatrix} s_1\mathrm{R}_1 \\ s_2\mathrm{R}_2 \\ \vdots \\ s_m\mathrm{R}_m \end{bmatrix} \mathrm{S} + \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_m \end{bmatrix} \\ &= \mathrm{P}_{2m \times 2}\mathrm{S}_{2 \times n} + \eta \end{aligned} \tag{3.13}$$

Thus in the noiseless case, the measurement matrix has a maximum rank of two. The matrix P contains the pose or orientation for each trajectory. The matrix S contains the shape of the nominal trajectory for this activity.

Using the rank theorem for noisy measurements, the measurement matrix can be factorized into two matrices $\tilde{\mathrm{P}}$ and $\tilde{\mathrm{S}}$ by using SVD and retaining the top two singular values, as shown before.

$$\mathrm{W} = \mathrm{UDV}^T \tag{3.14}$$

and taking $\tilde{\mathrm{P}} = \mathrm{U}'\mathrm{D}'^{\frac{1}{2}}$ and $\tilde{\mathrm{S}} = \mathrm{D}'^{\frac{1}{2}}\mathrm{V}'^T$, where $\mathrm{U}', \mathrm{D}', \mathrm{V}'$ are the truncated versions of $\mathrm{U}, \mathrm{D}, \mathrm{V}$ by retaining only the top two singular values. However, this factorization is not

unique, as for any non-singular 2×2 matrix Q

$$\mathrm{W} = \tilde{\mathrm{P}}\tilde{\mathrm{S}} = (\tilde{\mathrm{P}}\mathrm{Q})(\mathrm{Q}^{-1}\tilde{\mathrm{S}}) \tag{3.15}$$

So, we want to remove this ambiguity by finding the matrix Q that would transform $\tilde{\mathrm{P}}$ and $\tilde{\mathrm{S}}$ into the pose and shape matrices $\mathrm{P} = \tilde{\mathrm{P}}\mathrm{Q}$ and $\mathrm{S} = \mathrm{Q}^{-1}\tilde{\mathrm{S}}$ as in (3.13). To find Q we use the metric constraint on the rows of P, as suggested in [104].

By multiplying P by its transpose P^T, we get

$$\mathrm{P}\mathrm{P}^T = \begin{bmatrix} s_1\mathrm{R}_1 \\ \vdots \\ s_m\mathrm{R}_m \end{bmatrix} \begin{bmatrix} s_1\mathrm{R}_1 \cdots s_m\mathrm{R}_m \end{bmatrix} = \begin{bmatrix} s_1^2\mathbf{I}_2 & & \\ & \ddots & \\ & & s_m^2\mathbf{I}_2 \end{bmatrix} \tag{3.16}$$

where $\mathbf{I}_2$ is a 2×2 identity matrix. This follows from the orthonormality of the rotation matrices R_j. Substituting for $\mathrm{P} = \tilde{\mathrm{P}}\mathrm{Q}$, we get

$$\mathrm{P}\mathrm{P}^T = \tilde{\mathrm{P}}\mathrm{Q}\mathrm{Q}^T\tilde{\mathrm{P}}^T = \begin{bmatrix} \mathbf{a_1} \\ \mathbf{b_1} \\ \vdots \\ \mathbf{a_m} \\ \mathbf{b_m} \end{bmatrix} \mathrm{Q}\mathrm{Q}^T \begin{bmatrix} \mathbf{a_1}^T \, \mathbf{b_1}^T \cdots \mathbf{a_m}^T \, \mathbf{b_m}^T \end{bmatrix} \tag{3.17}$$

where $\mathbf{a}_i$ and $\mathbf{b}_i$, $i = 1 : m$, are the odd and even rows of $\tilde{\mathrm{P}}$, respectively. From (3.16) and (3.17), we obtain the following constraints on the matrix $\mathrm{Q}\mathrm{Q}^T$, $\forall i = 1, \ldots, m$, such that

$$\begin{aligned} \mathbf{a}_i\mathrm{Q}\mathrm{Q}^T\mathbf{a}_i^T &= \mathbf{b}_i\mathrm{Q}\mathrm{Q}^T\mathbf{b}_i^T = s_i^2 \\ \mathbf{a}_i\mathrm{Q}\mathrm{Q}^T\mathbf{b}_i^T &= 0 \end{aligned} \tag{3.18}$$

Using these $2m$ constraints on the elements of $\mathrm{Q}\mathrm{Q}^T$, we can find the solution for $\mathrm{Q}\mathrm{Q}^T$. Then Q can be estimated through SVD, and it is unique up to a 2×2 rotation matrix.

This ambiguity comes from the selection of the reference coordinate system and can be eliminated by selecting the first trajectory as a reference, i.e by selecting $\mathrm{R}_1 = \mathrm{I}_{2\times2}$.

3.2.2.3 Testing Trajectories

In order to test whether an observed trajectory T_x belongs to a certain learnt activity or not, two steps are needed:

1. Compute the optimal rotation and scaling matrix $s_x\mathrm{R}_x$ in the least square sense, such that,

$$\mathrm{T}_x \simeq s_x\mathrm{R}_x\mathrm{S} \tag{3.19}$$

$$\begin{bmatrix} x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \end{bmatrix} \simeq s_x\mathrm{R}_x \begin{bmatrix} \tilde{x}_1 & \cdots & \tilde{x}_n \\ \tilde{y}_1 & \cdots & \tilde{y}_n \end{bmatrix} \tag{3.20}$$

The matrix $s_x\mathrm{R}_x$ has only two degrees of freedom, which correspond to the scale s_x and rotation angle θ_x, we can write the matrix $s_x\mathrm{R}_x$ as

$$s_x\mathrm{R}_x = \begin{bmatrix} s_x\cos\theta_x & -s_x\sin\theta_x \\ s_x\sin\theta_x & s_x\cos\theta_x \end{bmatrix} \tag{3.21}$$

By rearranging (3.20), we get $2n$ equations in the two unknown elements of $s_x\mathrm{R}_x$ in the form

$$\begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_m \\ y_m \end{bmatrix} = \begin{bmatrix} \tilde{x}_1 & -\tilde{y}_1 \\ \tilde{y}_1 & \tilde{x}_1 \\ \vdots & \vdots \\ \tilde{x}_m & -\tilde{y}_m \\ \tilde{y}_m & \tilde{x}_m \end{bmatrix} \begin{bmatrix} s_x\cos\theta_x \\ s_x\sin\theta_x \end{bmatrix} \tag{3.22}$$

Again, this set of equations is solved in the least square sense to find the optimal $s_x\mathrm{R}_x$ parameters that minimize the mean square error between the tested trajectory and the rotated nominal shape for this activity.

2. After the optimal transformation matrix is calculated, the correlation between the trajectory and the transformed nominal shape is calculated and used for making a decision. The Frobenius norm of the error matrix is used as an indication of the level of correlation, which represents the mean square error (MSE) between the two matrices. The error matrix is calculated as the difference between the tested trajectory matrix T_x and the rotated activity shape, as follows

$$\Delta_x = \mathrm{T}_x - s_x\mathrm{R}_x\mathrm{S} \tag{3.23}$$

The Frobenius norm of a matrix **A** is defined as the square root of the sum of the absolute squares of its elements,

$$||A||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}^2|} \tag{3.24}$$

The value of the error is normalized with the signal energy to give the final normalized mean square error (NMSE) defined as

$$NMSE = \frac{||\Delta_x||_F}{||\mathrm{T}_x||_F + ||s_x\mathrm{R}_x\mathrm{S}||_F} \tag{3.25}$$

Comparing the value of this NMSE to NMSE values of learnt activities, a decision can be made as to whether the observed trajectory belongs to this activity or not.

3.3 Estimating the Deformability Index (DI)

In this section, we describe a method for estimating the amount of deformation in a deformable 3D shape model [91]. This method is based on applying subspace analysis on the trajectories of the object points tracked over a video sequence. The estimation of DI is essential for our activity modeling approach that has been explained above. From one point of view, DI represents the amount of deformation in the 3D shape representing the activity. In other words, it represents the number of basis shapes (k in (3.1)) needed to represent each activity. On the other hand, in the analysis of ground plane activities, the DI estimated can be used to estimate the number of activities in the scene (i.e., to find the number of nominal trajectories), as we assumed that each activity can be represented by a single trajectory on the ground plane.

We will use the word trajectory to refer to either the tracks of a certain point of the object across different frames, or to refer to the trajectories generated by different objects moving in the scene in the ground plane scenario.

Consider each trajectory obtained from a particular video sequence to be the realization of a random process. Represent the x and y coordinates of the sampled points on these trajectories for one such realization as a vector $\mathbf{y} = [u_1, ..., u_P, v_1, ..., v_P]^T$. Then from (3.5), it is easy to show that for a particular example with K distinct motion trajectories

(K is unknown)

$$\mathbf{y}^T = \left[l_1\mathbf{R}^{(1)}, ..., l_K\mathbf{R}^{(1)}, l_1\mathbf{R}^{(2)}, ..., l_K\mathbf{R}^{(2)}\right] * \begin{bmatrix} S_1 & & \\ \vdots & 0 & \\ S_k & & \\ & S_1 & \\ 0 & \vdots & \\ & S_k & \end{bmatrix} + \eta^T, \tag{3.26}$$

$$i.e., \quad \mathbf{y} = (\mathbf{q}_{1\times 6K}\mathbf{b}_{6K\times 2P})^T + \eta \quad = \quad \mathbf{b}^T\mathbf{q}^T + \eta, \tag{3.27}$$

where η is a zero-mean noise process. Let $\mathbf{R}_\mathbf{y} = E[\mathbf{y}\mathbf{y}^T]$ be the correlation matrix of $\mathbf{y}$ and $\mathbf{C}_\eta$ the covariance matrix of η. Hence

$$\mathbf{R}_\mathbf{y} = \mathbf{b}^T E[\mathbf{q}^T\mathbf{q}]\mathbf{b} + \mathbf{C}_\eta. \tag{3.28}$$

C_η represents the accuracy with which the feature points are tracked and can be estimated from the video sequence using the inverse of the Hessian matrix at each of the points. Since η need not be an IID noise process, C_η will not necessarily have a diagonal structure (but it is symmetric). However, consider the singular value decomposition of $\mathbf{C}_\eta = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T$, where $\mathbf{\Lambda} = \text{diag}[\mathbf{\Lambda}_s, 0]$ and $\mathbf{\Lambda}_s$ is an $L \times L$ matrix of non-zero singular values of $\mathbf{\Lambda}$. Let $\mathbf{P}_s$ denote the columns of $\mathbf{P}$ corresponding to the non-zero singular values. Therefore, $\mathbf{C}_\eta = \mathbf{P}_s\mathbf{\Lambda}_s\mathbf{P}_s^T$. Premultiplying (3.27) by $\mathbf{\Lambda}_s^{-\frac{1}{2}}\mathbf{P}_s^T$, we see that (3.27) becomes

$$\tilde{\mathbf{y}} = \tilde{\mathbf{b}}^T\mathbf{q}^T + \tilde{\eta}, \tag{3.29}$$

where $\tilde{\mathbf{y}} = \mathbf{\Lambda}_s^{-\frac{1}{2}}\mathbf{P}_s^T\mathbf{y}$ is a $L \times 1$ vector, $\tilde{\mathbf{b}} = \mathbf{\Lambda}_s^{-\frac{1}{2}}\mathbf{P}_s^T\mathbf{b}^T$ is a $L \times 6K$ matrix and $\tilde{\eta} = \mathbf{\Lambda}_s^{-\frac{1}{2}}\mathbf{P}_s^T\eta$. It can be easily verified that the covariance of $\tilde{\eta}$ is an identity matrix $\mathbf{I}_{L\times L}$. This is known

as the process of "whitening", whereby the noise process is transformed to be IID. Representing by $\mathbf{R}_{\tilde{\mathbf{y}}}$ the correlation matrix of $\tilde{\mathbf{y}}$, it is easy to see that

$$\mathbf{R}_{\tilde{\mathbf{y}}} = \tilde{\mathbf{b}}^T E[\mathbf{q}^T\mathbf{q}]\tilde{\mathbf{b}} + \mathbf{I} \;=\; \mathbf{\Delta} + \mathbf{I}. \tag{3.30}$$

Now, $\mathbf{\Delta}$ is of rank $6K$, where K is the number of activities. Representing by $\mu_i(\mathbf{A})$ the i^{th} eigenvalue of the matrix $\mathbf{A}$, we see that $\mu_i(\tilde{\mathbf{y}}) = \mu_i(\mathbf{\Delta}) + 1$ for $i = 1, ..., 6K$ and $\mu_i(\tilde{\mathbf{y}}) = \mathbf{1}$, for $i = 6K + 1, ..., L$. Hence, by comparing the eigenvalues of the observation and noise processes, it is possible to estimate the Deformability Index. This is done by counting the number of eigenvalues of $\mathbf{R}_{\tilde{\mathbf{y}}}$ that are greater than 1, and dividing that number by 6 to get the DI value. The number of basis shapes can then be obtained by rounding the DI to the nearest integer.

3.3.1 Properties of the Deformability Index

- For the case of a 3D rigid body, the DI is 1. In this case, the only variation in the values of the vector $\mathbf{y}$ from one image frame to the next is due to the global rigid translation and rotation of the object. The rank of the matrix $\mathbf{\Delta}$ will be 6 and the deformability index will be 1.

- Estimation of DI does not require explicit computation of the 3D structure and motion in (3.5), as we need only to compute the eigenvalues of the covariance matrix of the 2D feature positions. In fact, for estimating the shape and rotation matrices in (3.5) it is essential to know the value of K. Thus the method outlined in this Section should precede computation of the shape in Section 3.2. Using our method, it is possible to obtain an algorithm for deformable shape estimation

without having to guess the value of K.

- The computation of DI takes into account any rigid 3D translation and rotation of the object (as recoverable under a scaled orthographic camera projection model), even though it has the simplicity of working only with the covariance matrix of the 2D projections. Thus it is more general than a method that considers purely 2D image plane motion.

- The "whitening" procedure described above enables us to choose a *fixed* threshold of one for comparing the eigenvalues.

3.4 Experimental Results

We performed two sets of experiments to show the effectiveness of our approach for characterizing activities. In the first, we use 3D shape models to model and recognize the activities performed by an individual, e.g. walking, running, sitting, crawling, etc. We show the effect of using a 3D model in recognizing these activities from different viewing angles. In the second set of experiments, we provide results for the special case of ground plane surveillance trajectories resulting from a motion detection and tracking system [2]. We explore the effectiveness of our formulation in modeling nominal trajectories and detecting anomalies in the scene. In the first experiment, We assume a robust tracking of the feature points across the sequence. This enable us to focus on whether the 3D models can be used to disambiguate among different activities in various poses and the selection of the criterion to make this decision. However, as pointed out in the original factorization paper [104] and in its extensions to deformable shape model in [107], the rank constraint

algorithms can estimate the 3D structure even with noisy tracking results.

3.4.1 Application in Human Activity Recognition

We used our approach to classify the various activities performed by an individual. We used the motion-capture data [1] available from Credo Interactive Inc. and Carnegie Mellon University in the BioVision Hierarchy and Acclaim formats. The combined dataset included a number of subjects performing various activities, like walking, jogging, sitting, crawling, brooming, etc. For each of these activities, we had multiple video sequences consisting of 72 frames each, with different viewpoint sequences for many of the activities.

3.4.1.1 Computing the DI for Different Human Acivity

For the different activities in the database, we used an articulated 3D model for the body that contains 53 tracked feature points. We used the method described in Section 3.3 to the trajectories of these points to compute the DI for each of these sequences. These values are shown in Table 3.1 for various activities. Please note that DI is used to estimate the number of basis shapes needed for 3D deformable object modeling, not for activity recognition.

From Table 3.1, a number of interesting observations can be made. For the walk sequences, the DI is between 5 and 6. This matches the hypotheses in papers on gait recognition where it is mentioned that about five exemplars are necessary to represent a full cycle of gait [58]. The number of basis shapes increases for fast walk, as expected

Table 3.1: Deformability Index (DI) for Human Activities Using Motion Capture Data

	Activity	DI		Activity	DI
1	Male Walk (Seq. 1)	5.8	10	Broom (Seq. 2)	8.8
2	Male Walk (Seq. 2)	4.7	11	Jog	5.0
3	Fast Walk	8.0	12	Blind Walk	8.8
4	Walk throwing hands around	6.8	13	Crawl	8.0
5	Walk with drooping head	8.8	14	Jog while taking U-turn (Seq. 1)	4.8
6	Sit (Seq. 1)	8.0	15	Jog while taking U-turn (Seq. 2)	5.0
7	Sit (Seq. 2)	8.2	16	Broom in a circle	9.0
8	Sit (Seq. 3)	8.2	17	Female Walk	7.0
9	Broom (Seq. 1)	7.5	18	Slow Dance	8.0

from some of the results in [103]. When the stick figure person walks doing some other things (like moving head or hands or a blind person's walk), the number of basis shapes needed to represent it (i.e. the deformability index) increases from that of normal walk. The result that might seem surprising initially is the high DI for sitting sequences. On closer examination though, it was found that the stick figure, while sitting, was making all kinds of random gestures as if talking to someone else. That increased the DI for these sequences. Also, the DI is insensitive to changes in viewpoint (azimuth angle variation only), as can be seen by comparing the jog sequences (14 and 15 with 11) and broom sequences (16 with 9 and 10). This is not surprising since we do not expect the deformation of the human body to change due to rotation about the vertical axis. The DI, thus calculated, is used to estimate the 3D shape, some of which are shown in Figure 3.2 and are used in activity recognition experiments.

3.4.1.2 Activity Representation using 3D models

Using the video sequences and our knowledge of the DI for each activity, we applied the method outlined in Section 3.2 to compute the basis shapes and their combination coefficients (see (3.1)). The orthonormality constraints in [107] are used to get a unique solution for the basis shapes. We found that the first basis shape, S_1, contained most of the information. The estimated first basis shapes are shown in Figure 3.2 for six different activities. For this application, considering only the first basis shape was enough to distinguish between the different activities, i.e., the recognition results did not improve with adding more basis shapes, although the differences between the different models in-

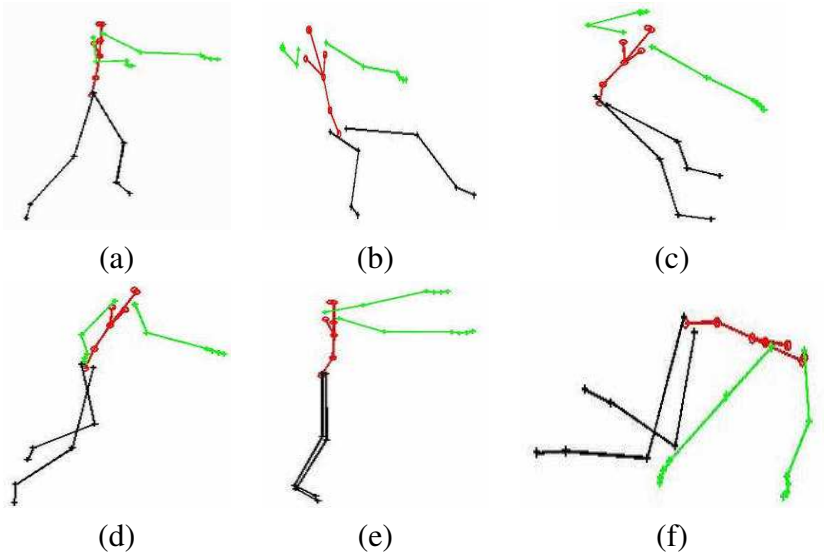
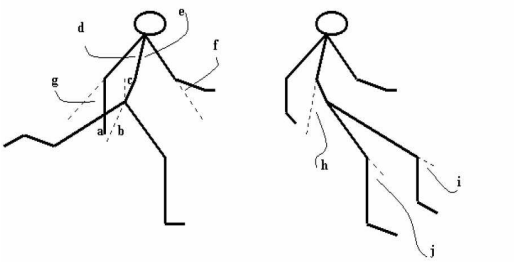


Figure 3.2: Plots of the first basis shape, S_1 for walk, sit and broom sequences, (a)-(c), and for jog, blind walk and crawl sequences, (d) - (f).

creased. This is a peculiarity of this dataset and will not be true in general. In order to compute the similarity measure, we considered the various joint angles between the different parts of the estimated 3D models. The angles considered are shown in Figure 3.3(a). The idea of considering joint angles for activity modeling has been suggested before in [20]. We considered the seven dimensional vector obtained from the angles as shown in Figure 3.3(a). The distance between the two angle vectors was used as a measure of similarity. Thus small differences indicated higher similarity.

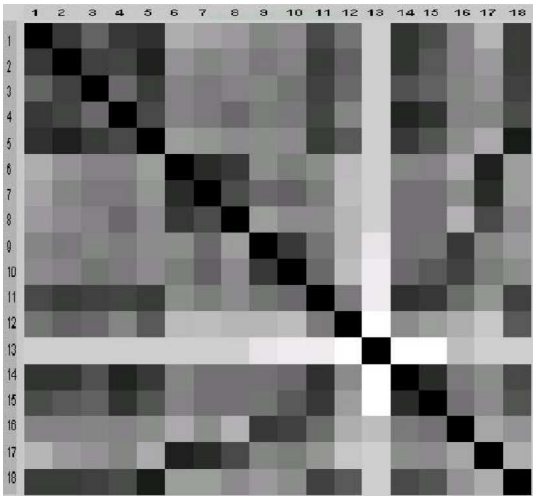
The similarity matrix is shown in Figure 3.3(b). The row and column numbers correspond to the numbers in Table 3.1 for 1-16, while 17 and 18 correspond to sitting and walking, where the training and test data are from two different viewing directions. For the moment, consider the upper 13 x 13 block of this matrix. We find that the different walk sequences are close to each other, this is also true for sitting and brooming sequences. The jog sequence, besides being closest to itself, is also close to the walk sequences. Blind



Angles we are using in our correlation criteria (ordered from highest weights to lowest)

1. c → angle btw. Hip-abdomen and vertical axis
2. h → angle btw. Hip-abdomen and chest
3. (a+b)/2 → average angle btw. Two legs and abdomen-hip axis
4. (b-a)/2 → the angle difference between two upper legs
5. (i+j)/2 → average angle btw. upper legs and lower legs
6. (d+e)/2 → average angle btw. Upper arms and abdomen-chest
7. (f+g)/2 → average angle btw. Upper arms and lower arms

(a)



(b)

Figure 3.3: (a): The various angles used for computing the similarity of two models is shown in the Figure. The text below describes the seven dimensional vector computed from each model and whose correlation determines the similarity scores. (b): The similarity matrix for the various activities, including ones with different viewing directions. The numbers correspond to the numbers in Table 3.1 for 1-16. 17 and 18 correspond to sitting and walking, where the training and test data are from two different viewing directions.

walk is close to jogging and walking. The crawl sequence does not match any of the rest and this is clear from Row 13 of the matrix. Thus, the results obtained using our method are reasonably close to what we would expect from a human observer, which support the use of this representation in activity recognition.

In order to further show the effectiveness of this approach, we used the obtained similarity matrix to analyze the recognition rates for different clusters of activities, We applied different thresholds on the matrix and calculated the recall and precision values for each cluster. The first cluster contains the walking sequences along with jogging and blind walk (activities 1-5,11, and 12 in Table 3.1). Figure 3.4(a) shows the recall vs. precision values for this activity cluster, we can see from the figure that we are able to

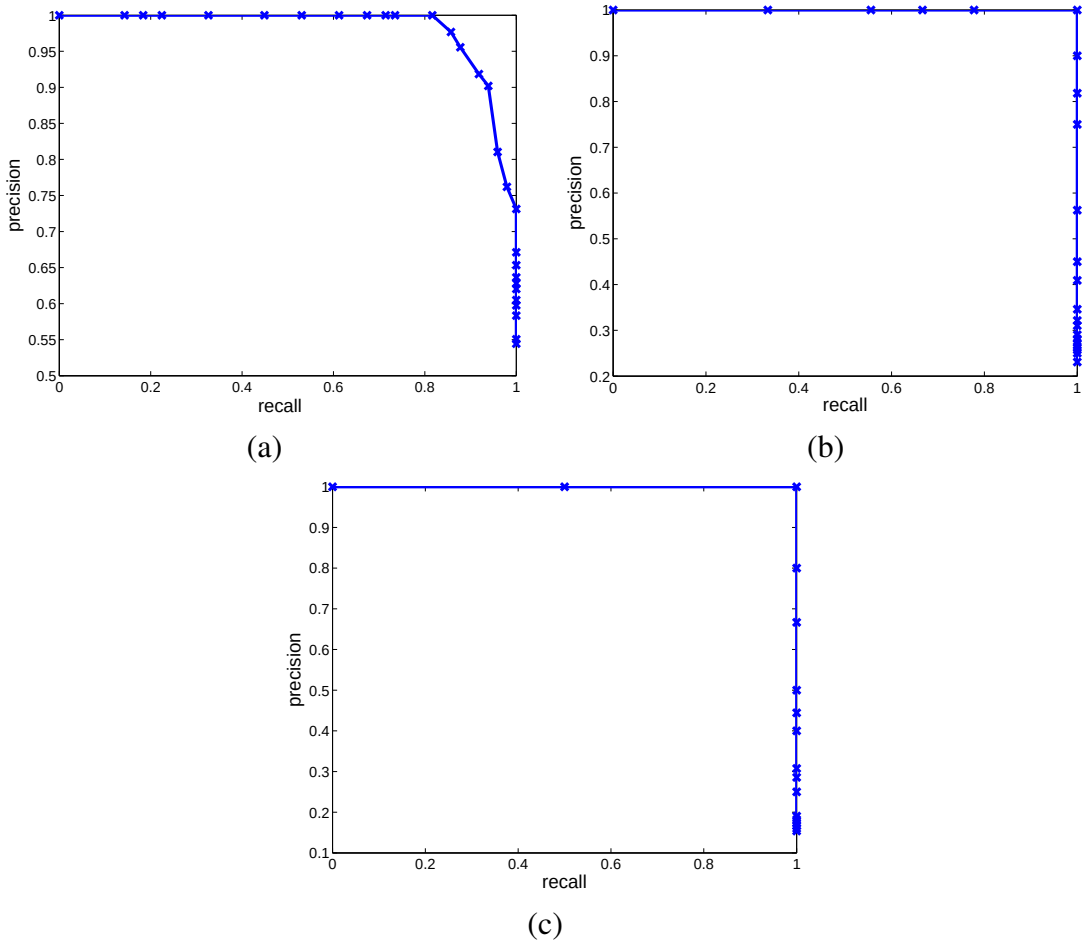
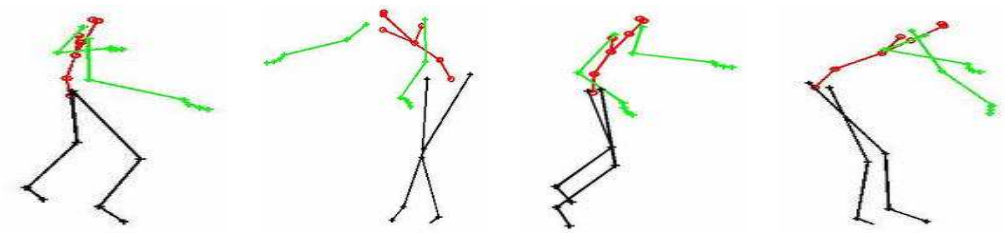


Figure 3.4: The recall vs. precision rates for the detection of three different clusters of activities. (a) Walking activities (activities 1-5,11, and 12 in Table 3.1) (b) Sitting activities (activities 6-9 in Table 3.1) (c) Brooming activities (activities 9 and 10 in Table 3.1)

identify 90% of these activities with a precision up to 90%. The second cluster consists of three sitting sequences (activities 6-8 in Table 3.1), and the third cluster consists of the brooming sequences (activities 9 and 10 in Table 3.1). For both of these clusters the similarity values were quite separated to the extent that we were able to fully separate the positive and negative examples. This resulted in the recall vs. precision curves as shown in Figure 3.4(b) and Figure 3.4(c).

3.4.1.3 View-Invariant Activity Recognition

In this part of the experiment, we consider the situation where we try to recognize activities when the training and testing video sequences are from different viewpoints. This is the most interesting part of the method, as it demonstrates the strength of using 3D models for activity recognition. In our dataset, we had three sequences where the motion is not parallel to the image plane, two for jogging in a circle and one for brooming in a circle. We considered a portion of these sequences where the stick figure is not parallel to the camera. From each such video sequence, we computed the basis shapes. Each basis shape is rotated, based on an estimate of its pose, and transformed to the canonical plane (i.e. parallel to the image plane). The basis shapes before and after rotation are shown in Figure 3.5. The rotated basis shape is used to compute the similarity of this sequence with others, exactly as described above. Rows 14-18 of the similarity matrix shows the recognition performance for this case. The jogging sequences are close to jogging in the canonical plane (Column 11), followed by walking along the canonical plane (Columns 1-6). For the broom sequence, it is closest to a brooming activity in the canonical plane



(a) (b) (c) (d)

Figure 3.5: (a) and (b) plot the basis shapes for jogging and brooming when the viewing direction is different from the canonical one. (c) and (d) plot the rotated basis shapes.

(Column 9 and 10). The sitting and walking sequences (columns 17 and 18) of the test data are close to the sitting and walking sequences in the training data, even though they were captured from different viewing directions.

3.4.2 Application in Characterization of Ground Plane Trajectories

Our second set of experiments was directed towards the special case of ground plane motion trajectories. The proposed algorithm was tested on a set of real trajectories, generated by applying a motion-detection and tracking system [2] on the force protection surveillance system (FPSS) dataset provided by U. S. Army Research Laboratory (ARL). These data sequences represent the monitoring of humans and vehicles moving around in a large parking lot. The normal activity in these sequences corresponds to a person moving into the parking lot and approaching his or her car, or stepping out of the car and moving out of the parking lot. We manually picked the trajectories corresponding to normal activities from the tracking results to assure stable tracking results in the training set.

In this experiment we deal with a single normal activity. However, for more complicated scenes, the algorithm can handle multiple activities by first estimating the number

of activities using the DI estimation procedure in Section 3.3, and then performing the following learning procedure for each activity.

3.4.2.1 Time Scaling

One of the major challenges in comparing activity is to remove the temporal variation in the way the activity is being executed. Several techniques were used to face this challenge as in [121], where the authors used Dynamic Time Warping (DTW) [83] to learn the nature of time warps between different instants of each activities. This technique could have been used in our problem as a pre-processing stage for the trajectories to compensate for these variations before computing the nominal shape of each activity. However, the nature of the ground plane activities in our experiment did not require such sophisticated techniques, so we used a much simpler approach to be able to compare trajectories of different lengths (different number of samples n), and to explore the temporal effect. We adopt the multi-resolution, time scaling approach described below:

- Each trajectory is divided into segments of a common length n. We pick $n = 50$ frames in our experiment.

- A multi-scale technique is used for testing different combinations of segments, ranging from the finest scale (the line segments) to the coarsest scale (the whole trajectory). This technique gives the ability to evaluate each section of the trajectory along with the overall trajectory. An example of the different training sequences that can be obtained from a $3n$ trajectory is given in Table 3.2, where $Downsample_m$ denotes the process of keeping every m^{th} sample and discarding the rest. We provide a

Table 3.2: The different trajectory sequences generated from a three-segments trajectory.

x_1	$\cdots$	x_n	x_{n+1}	$\cdots$	x_{2n}	x_{2n+1}	$\cdots$	x_{3n}
y_1	$\cdots$	y_n	y_{n+1}	$\cdots$	y_{2n}	y_{2n+1}	$\cdots$	y_{3n}

Scale	Segment Representation	Trajectory points	Processing type
1	(1,1)	$x_1 : x_n$ $y_1 : y_n$	*No Processing* *No Processing*
	(1,2)	$x_{n+1} : x_{2n}$ $y_{n+1} : y_{2n}$	*No Processing* *No Processing*
	(1,3)	$x_{2n+1} : x_{3n}$ $y_{2n+1} : y_{3n}$	*No Processing* *No Processing*
2	(2,1)	$x_1 : x_{2n}$ $y_1 : y_{2n}$	$Downsample_2$ $Downsample_2$
	(2,2)	$x_{n+1} : x_{3n}$ $y_{n+1} : y_{3n}$	$Downsample_2$ $Downsample_2$
3	(3,1)	$x_1 : x_{3n}$ $y_1 : y_{3n}$	$Downsample_3$ $Downsample_3$

representation of the segments in the form of an ordered pair (i, j) where i represent the scale of the segment and j represent the order of this segment within the scale i.

An important property of this time scaling approach is that it captures the change in motion pattern between segments because of grouping of all possible combinations of adjacent segments. This can be helpful as the abrupt change in human motion pattern, like sudden running, is a change that is worthy of being singled out in surveillance applications.

3.4.2.2 Ground Plane Recovery

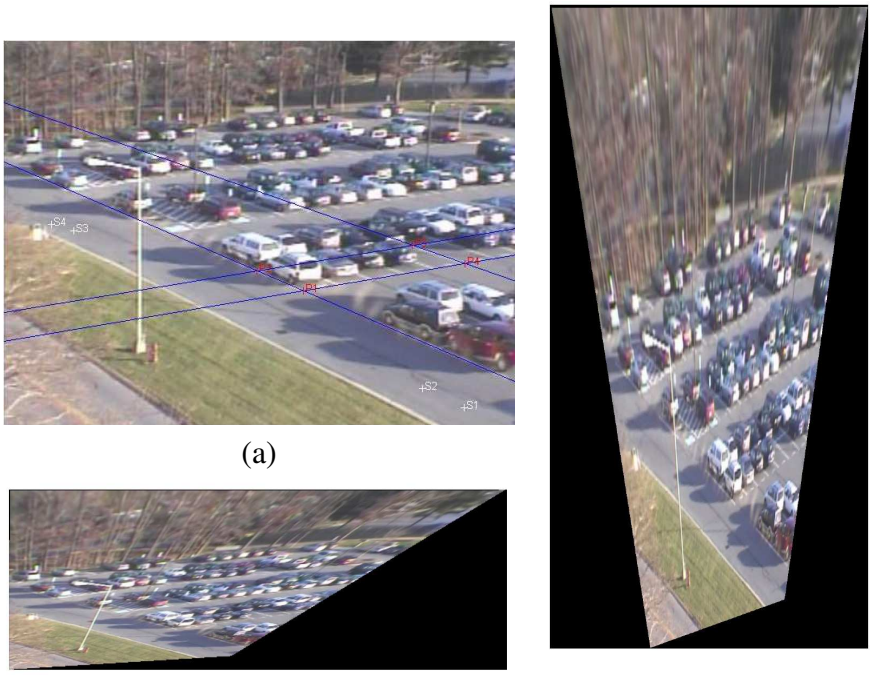
This is the first step in our method. This calibration process needs to be done once for each camera, and the transformation matrix can then be used for all the subsequent sequences because of the stationary setup. The advantage of this method is that it does not need any ground truth information and can be performed using some features that are common in man-made environments.

As described before, the first stage recovers the affine parameters by identifying the vanishing line of the ground plane. This is done using two parallel lines as shown in Figure 3.6(a), the parallel lines are picked as the horizontal and vertical borders of a parking spot. Identifying the vanishing line is sufficient to recover the ground plane up to an affine transformation as shown in Figure 3.6(b).

The second stage is to recover the ground plane up to a metric transformation, which is achieved using two affine invariant properties. The recovery result is shown in Figure 3.6(c). In our experiment we used the right angle between the vertical and horizontal borders of parking space, and the equal length of the tires span of a tracked truck across frames as shown by the white points (S_1, S_2) and (S_3, S_4) in Figure 3.6(a).

3.4.2.3 Learning the Trajectories

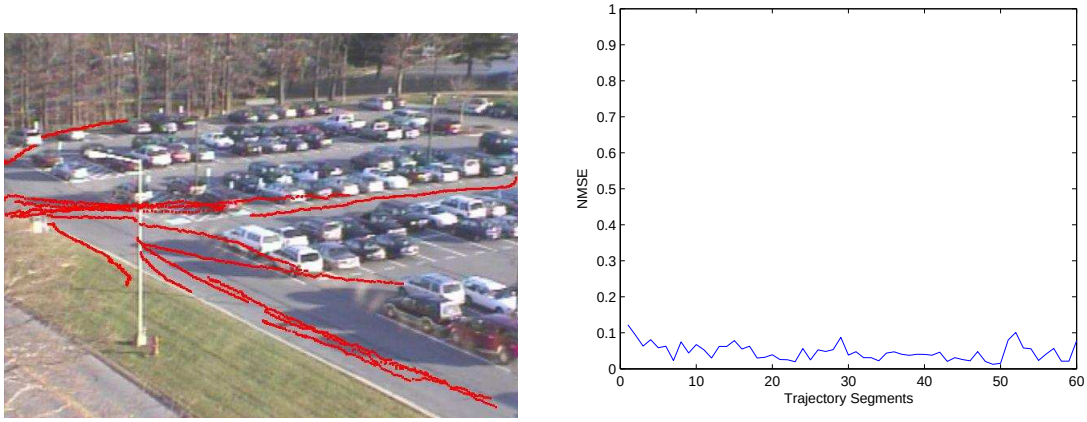
For learning the normal activity trajectory, we used a training data set containing the tracking results for 17 objects of different track length. The normal activity in this data corresponds to a person entering the parking lot and moving towards a car, or a person leaving the parking lot. The trajectories were first smoothed using a five-point moving



(a)

(b) (c)

Figure 3.6: The recovery of the ground plane. (a) The original image frame with the features used in the recovery process. (b) The affine rectified image.(c) The metric rectified image.



(*a*) (*b*)

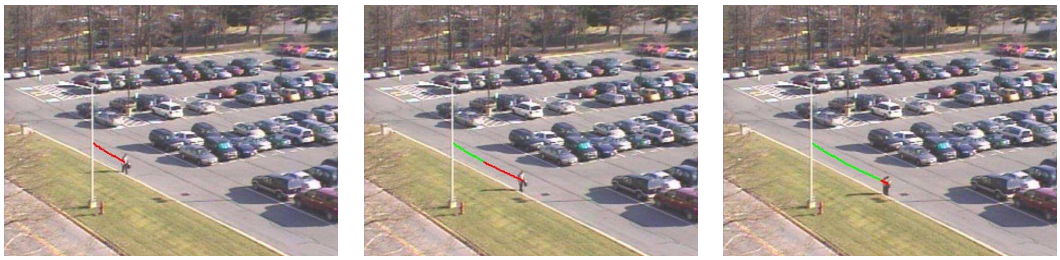
Figure 3.7: (a) The normal trajectories and (b) The associated normalized mean square error (NMSE) values.

averaging to remove tracking errors and then they were used to generate track segments of 50 points length as described earlier, resulting in 60 learning segments. Figure 3.7(a) shows the image plane trajectories used in the learning process, and each of the red points represents the center of the bounding box of an object in a certain frame.

This set of trajectories is used to determine the range of the NMSE in the case of a normal activity trajectory. Figure 3.7(b) shows the NMSE values for the segments of the training set sequence.

3.4.2.4 Testing Trajectories for Anomalies

First Abnormal Scenario This testing sequence represents a human moving in the parking lot and then stopping in the same location for some time. The first part of the trajectory, which lasts for 100 frames (two segments), is a normal activity trajectory, but the third segment represents an abnormal act. This could be a situation of interest in surveillance scenario. Figure 3.8 shows the different segments of the object trajectory, along



NMSE(1,1)=0.0225 NMSE(1,2)=0.0296 NMSE(2,1)=0.0146 NMSE(1,3)=0.3357 NMSE(2,2)=0.1809 NMSE(3,1)=0.1123

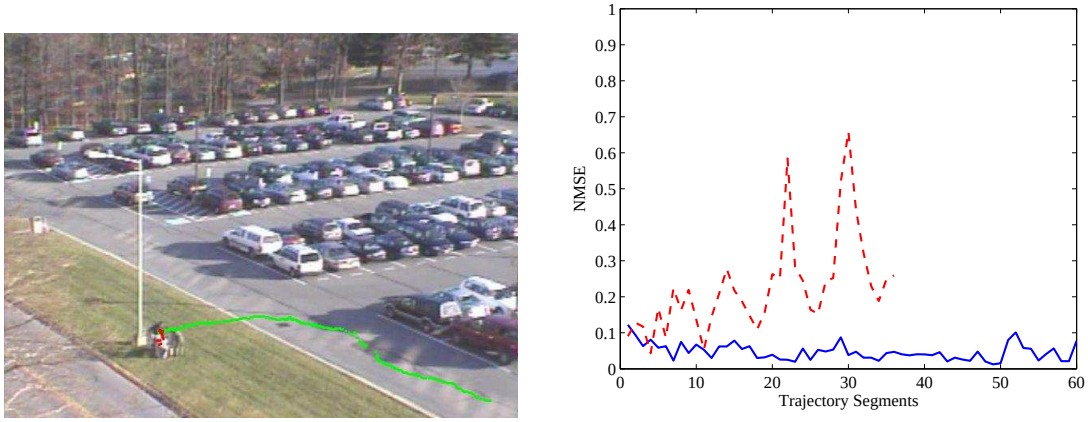
Figure 3.8: The first abnormal test scenario. A person stops moving at a point on his route. We see the increase in the normalized mean square error (NMSE) values when he/she stopped, resulting in a deviation from the normal trajectory

with the NMSE associated with each new segment. We see that as the object stops moving in the third segment, the NMSE values raise to indicate a possible drift of the object trajectory from the normal trajectory.

Second Abnormal Scenario In this abnormal scenario, a group of tracked humans drift from their path into the grass area surrounding the parking lot, stop there to lift a large box and then move the box. Figure 3.9(a) shows the object trajectory. Figure 3.9(b) shows a plot of the NMSE of all the segments, in red, with relative to the normal trajectory NMSE, in blue. It can be verified from the figure that the trajectory was changing from normal to abnormal one in the last three or four segments, which caused the NMSE of the global trajectory to raise.

3.5 Chapter Summary

In this chapter, we presented a framework for using 3D deformable shape models for activity modeling and representation. This has the potential to provide invariance



(*a*) (*b*)

Figure 3.9: The second testing scenario. (a) A group of people on a path towards a box. (b) The increase in the NMSE with time as the abnormal scenario is being performed

to viewpoint and more detailed modeling of illumination effects. The 3D shape is estimated from the motion trajectories of the points participating in the activity under a weak perspective camera projection model. Each activity is represented using a linear combination of a set of 3D basis shapes. We presented a theory for estimating the number of basis shapes, based on the DI of the 3D deformable shape. We also explored the special case of ground plane motion trajectories, which often occurs in surveillance applications, and provided a framework for using our proposed approach for detecting anomalies in this case. We presented results showing the effectiveness of our 3D model in representing human activity for recognition, and performing a ground plane activity modeling and anomaly detection. The main challenge in this framework will be in developing representations that are robust to errors in 3D model estimation. Also, machine learning approaches that take particular advantage of the availability of 3D models will be an interesting area of future research.

Chapter 4

Affine-Invariant Activity Recognition on the Grassmann Manifold of Deformation Subspaces

4.1 Introduction

In the previous chapter, we proposed an approach for activity representation and recognition based on recovering the 3D deformation shape basis generated by each activity. The results shown for this approach presented some promising discrimination properties for these 3D shapes, which were effective for performing activity recognition. However, several challenges can be identified with that approach.

One of these challenges is due to the issue of non-uniqueness of the solution to the factorization problem in the non-rigid case. This results in ambiguity in recovering the 3D basis shapes upto a metric transformation as shown by Xiao *et al.* [129], which can in turn results in large distances between the recovered basis for different realization of the same activity.

Another challenge with that approach lies in the way distances are computed between different activity basis shapes. As basis shapes are recovered up to a metric transformation, a similarity invariant features is used to compute the distance between two basis shapes. For example, we used the different angles between the limbs for performing the classification in the previous chapter. However, this distance model is hard to extend

to compute the distance between multiple basis shapes for each activity. Meanwhile, it does not allow for building statistical models for performing robust statistical classification when several training samples are available for each activity.

In this chapter, we propose to use statistical models on the Grassmann manifold for affine invariant action recognition. Instead of recovering the exact 3D basis shapes, we compute an affine invariant subspace representation of the action. This subspace representation captures the underlying deformation modes associated with the action under a linear basis deformation model similar to the model used in Non-rigid structure from motion literature (NRSFM). We design statistical classification rules on the Grassmann manifold and apply them to these subspaces for activity recognition.

Our approach has the advantage that it does not need a complete recovery of the actual 3D shape of the subject performing the action, which eliminates the need to solve for the metric upgrade in the NRSFM formulation. It also incorporates affine invariance into the learnt representation, which can cope with viewpoint and body type variations. An invariance to the temporal rate of execution of the action is inherent in our model as well, since the classification is performed on the view-invariant subspace representing the modes of deformation rather than the actual temporal trajectory within that subspace.

The idea of learning an invariant subspace representation for different human actions was previously proposed in [96]. In that work, the time was added as a forth dimension to the basis shapes to deal with execution rate invariance. A single subspace is learnt for each action using all the training samples, and the distance between the test sequence and that subspace determines the classification distance. In our approach, a deformation subspace is learnt for each realization and Grassmann manifold statistical tools allow us

to learn a mean subspace representation and different statistical modes of variations for each action.

4.1.1 Contributions

Our contribution in this chapter can be summarized as

1. Modeling the problem of affine-invariant activity recognition as classification problem on the space of deformation subspaces of landmark data.

2. Studying and proposing several statistical models on the Grassmann manifold to solve the activity classification problem.

3. Proposing a multi scale Bayesian formulation to deal with the variable degree of deformation for different actions.

4.1.2 Organization

The reminder of this chapter is organized as follows: In Section 4.2, we introduce the subspace representation for 3D deformation modes. Section 4.3 gives a brief overview of the Grassmann manifold and how to compute geodesics between points on this manifold. Section 4.4 describes the different Grassmann classification models proposed for activity recognition. Experimental results validating the proposed method for human action recognition are introduced in Section 4.5. The chapter is concluded with a summary in Section 4.6.

4.2 Activity Deformation Subspace Representation

We propose a framework for recognizing human actions by first extracting the trajectories of the various points on the body under arbitrary camera viewpoints, followed by a nonrigid 3D shape model fitted to the trajectories. The underlying intuition is that the learnt model completely characterizes the space of body poses associated with the action.

Following the same formulation of the previous chapter, we hypothesize that the deformable shape sequence corresponding to each action can be represented as a linear combination of 3D basis shapes. Mathematically, if we consider the trajectories of P points representing the shape (e.g. landmark points), then the overall configuration of the P points is represented as a linear combination of the basis shapes S_i as

$$S = \sum_{i=1}^{K} l_i S_i, \qquad S, S_i \in \Re^{3\times P}, l \in \Re. \tag{4.1}$$

where l_i represent the weight associated with the basis shape S_i.

Given F frames of a video sequence observing these P moving points from different camera view points, we first obtain the trajectories of all these points over the entire video sequence. These P points can be represented in a measurement matrix as

$$\mathbf{W}_{2F\times P} = \begin{bmatrix} u_{1,1} & \cdots & u_{1,P} \\ v_{1,1} & \cdots & v_{1,P} \\ \vdots & \vdots & \vdots \\ u_{F,1} & \cdots & u_{F,P} \\ v_{F,1} & \cdots & v_{F,P} \end{bmatrix}, \tag{4.2}$$

where $u_{f,p}$ represents the x-position of the p^{th} point in the f^{th} frame and $v_{f,p}$ rep-

resents the y-position of the same point.

Instead of the weak perspective projection assumed in the previous chapter, we assume a more general affine camera projection. The affine camera model covers the composite effect of an affine transformation of the 3D space, orthographic projection from 3D space to the image, and an affine transformation in the 2D image plane. Under this projection, the P points of a configuration in a frame f, are projected onto 2D image points $(u_{f,i}, v_{f,i})$ as

$$\begin{bmatrix} u_{f,1} & \cdots & u_{f,P} \\ v_{f,1} & \cdots & v_{f,P} \end{bmatrix} = \mathbf{A_f}\left(\sum_{i=1}^{K} l_{f,i} S_i\right) + \mathbf{T}_f, \tag{4.3}$$

where,

$$\mathbf{A}_f = \begin{bmatrix} a_{f1} & a_{f2} & a_{f3} \\ a_{f4} & a_{f5} & a_{f6} \end{bmatrix} \tag{4.4}$$

$\mathbf{A}_f$ represents the affine camera projection matrix and $\mathbf{T}_f$ is the camera translation.

The rank constraint on the measurement matrix can be derived by first eliminating the translation component by subtracting out the mean of all the 2D points, as in [104]. We now form the measurement matrix $\mathbf{W}$, which was represented in (4.2), with the means of each of the rows subtracted. Using (4.2) and (4.3), it is easy to show that

$$\begin{aligned} \mathbf{W} &= \begin{bmatrix} l_{1,1}\mathbf{A}_1 & \cdots & l_{1,K}\mathbf{A}_1 \\ l_{2,1}\mathbf{A}_2 & \cdots & l_{2,K}\mathbf{A}_2 \\ \vdots & \vdots & \vdots \\ l_{F,1}\mathbf{A}_F & \cdots & l_{F,K}\mathbf{A}_F \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_K \end{bmatrix} \\ &= \mathbf{M}_{2F\times 3K}.\mathbf{S}_{3K\times P}, \end{aligned} \tag{4.5}$$

Which is of rank $3K$. The matrix $\mathbf{M}$ contains the pose for each frame of the video sequence and the weights $l_1, ..., l_K$. The matrix $\mathbf{S}$ contains the basis shapes corresponding to each activity. Similar to SFM applications, if we assume an isotropic and Gaussian noise model [107], this factorization of the measurement matrix can be performed using SVD and retaining the top $3K$ singular values, as $\mathbf{W}_{2F\times P} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ and $\mathbf{M} = \mathbf{U}\mathbf{D}^{\frac{1}{2}}$ and $\mathbf{S} = \mathbf{D}^{\frac{1}{2}}\mathbf{V}^{\mathbf{T}}$.

Under the assumed linear basis shape model, the basis shapes in the matrix $\mathbf{S}$ can be used to reconstruct any of the 3D deformable shapes in the shape sequences. For a shape sequence that correspond to a certain activity, this enables the synthesis of different body poses that constitute the activity. However, the basis recovered using the SVD factorization are unique only up to an invertible transformation $\mathbf{G} \in \mathbb{GL}(3K, 3K)$, since

$$\mathbf{W} = \mathbf{M}\mathbf{S} = \mathbf{M}\mathbf{G}\mathbf{G}^{-\mathbf{1}}\mathbf{S} \tag{4.6}$$

This non-uniqueness implies the need to use statistical tools that are invariant to the linear transformation group $\mathbb{GL}(3K, 3K)$ while comparing basis shapes $\mathbf{S}$. In other words, two basis shape matrices $\mathbf{S_1}$ and $\mathbf{S_2}$ are considered equivalent if they are related by a linear invertible transformation in the form $\mathbf{S_1} = \mathbf{G}\mathbf{S_2}$. This equivalence class can be constructed by comparing the subspace spanned by the columns of $\mathbf{S_1}$ and $\mathbf{S_2}$ rather than comparing the matrices themselves. For each activity, this subspace represents the deformation space corresponding to the body poses of the activity, and we denote it by activity deformation subspace. This maps the problem of activity modeling and recognition to a problem of building statistical and classification tools on the space of $3k - dim$ subspaces in $\mathbb{R}^P$.

We note here that this situation arises from our activity recognition problem formulation. On the other hand, in NRSFM formulation, there is a need to recover exact 3D basis shapes that corresponds to the true camera motion projection matrices. This necessiates a search for a specific basis for the subspace spanned by the columns of $\mathbf{S}$. This is usually done by searching for the linear transformation $\mathbf{G}$, that enforce the orthonormality constrains on the rows of $\mathbf{M}$ [107] assuming a weak-perspective camera model. This is not an easy job due to the ambiguity of the problem as shown in [129], and further constrains on the nature of the deformation [106] is needed to reach for near optimal solution.

The space of $m - dimensional$ subspaces in $\mathbb{R}^n$ is called the Grassmann manifold and denoted by $\mathbb{G}(m, n)$. In order to build statistical and classification models for the deformation subspaces, we need to study the intrinsic geometry of this manifold and the ways to define distances and statistics on this manifold. In the next section, We will review the geometry of the Grassmannian $\mathbb{G}(m, n)$ as a quotient space over the orthogonal group $SO(n)$.

Affine Invariance of the deformation subspaces

View-invariant activity recognition refers to the ability of the model to capture the activity under arbitrary camera viewpoints or geometric transformation of the objects. One of the interesting geometric transformation group is the affine group. Assuming affine invariance, two sets of activity 3D shapes are considered equivalent if they are related by a set of 3D affine transformations. For example, a sequence of 3D landmark shapes $(S^1, S^2, \cdots, S^F)$ is equivalent to $(\mathbf{A}^1 S^1, \mathbf{A}^2 S^2, \cdots, \mathbf{A}^F S^F$ for arbitrary set of affine

transformations. These transformations can be different for every frame as in the case of different camera viewpoints, or the same affine transformation can be shared among all the frames, for example accounting for anthropometric variation between different people performing the activity.

The deformation subspace representation proposed in this chapter achieve this affine invariance. As it assumes an affine camera projection model. Hence, all the affine variation are captured in the camera motion matrix $\mathbf{M}$ while computing the deformation subspace.

4.3 Geometry of Grassmannian

The Grassmann manifold $\mathbb{G}(m,n)$ is defined as the space of $m-dimensional$ subspaces in $\mathbb{R}^n$. Several textbooks [47, 63] describe the structure of $\mathbb{G}(m,n)$ with a focus on its geometric and calculus. Edelman *et. al.* [30] use the differential geometry of Grassmann and other orthogonally constrained manifolds in order to provide gradient solutions to optimization problems. Srivastava et al. [99] derived the geodesics and analyzed the associated structure via Lie group theory. Statistical methods on Stiefel and Grassmann manifolds were proposed in [112] for several computer vision applications.

Any element in the manifold $\mathbb{G}(m,n)$, i.e. any m-dimensional subspace of $\mathbb{R}^n$ can either be represented by its projection operator (uniquely) or by an orthonormal basis (non-uniquely). In the former representation, a convenient approach is to view $\mathbb{G}(m,n)$ as a quotient space $SO(n)/(SO(m)\times SO(n-m))$ where $SO(n)$ is the Lie group of $n\times n$ real-valued rotation matrices. A lie group is a differentiable manifold with a group structure.

$SO(n)$ forms a group with matrix multiplication as the group operation. The advantage of this approach is to utilize well-known results form Lie group theory in deriving algorithms on $\mathbb{G}(m,n)$.

It is well known that geodesic paths on $SO(n)$ are given by a one-parameter exponential flow, i.e. $t \mapsto exp(tB)$, where $B \in \mathbb{R}^{n\times n}$ is a skew-symmetric matrix. Since $\mathbb{G}(m,n)$ is a quotient space of $SO(n)$, geodesics in $SO(n)$ are also geodesics in $\mathbb{G}(m,n)$ as long as they are perpendicular to the orbits generated by the subgroup $SO(m) \times SO(m-n)$. This implies that geodesics in $\mathbb{G}(m,n)$ are given by a one-parameter exponential flow $t \mapsto exp(tB)$, where skew-symmetric B is further restricted to be on the form

$$B = \begin{pmatrix} 0 & A^T \\ -A & 0 \end{pmatrix}, A \in \mathbb{R}^{(n-m)\times m} \tag{4.7}$$

The superscript T denotes the matrix transpose. The sub-matrix A specifies the tangent vector representing the direction and the speeds of the geodesic flow. Please refer to [99] for details.

The canonical metric on the Grassmann manifold is the restriction of the orthogonal group metric to the horizontal space. Let B_1 and B_2 be two tangent vectors on the form described in (4.7). Then

$$g(B_1, B_2) = trace({A_1}^T A_2) \tag{4.8}$$

which is equivalent to the standard inner product in $n(n-m)$-dimensional Euclidean space.

Given two points on the Grassmann manifold specified by the projection matrices P_1 and P_2, we can define the motion parameters such as displacement and the velocity for going from P_1 to P_2 in unit time. This is done by calculating the matrix X which defines

a geodesic α from P_1 to P_2, where X is in the form defined in (4.7). Thus, the geodesic distance between P_1 and P_2 can be calculated using (4.8) in the form

$$d_g(P_1, P_2) = g(X, X) = trace(X^T X) \tag{4.9}$$

4.3.1 Computation of Grassmann intrinsic Mean

Computation of the mean of several points on the Grassmann manifold can be the first step in computing the conditional statistics of a certain action. The intrinsic mean or Karcher mean μ is defined as a local minimizer in $\mathbb{G}(p, 3k)$ of the sum-of-squared Riemannian distances to each point. Thus, this intrinsic mean is given by

$$\mu = \underset{x \in \mathbb{G}(p,3k)}{\operatorname{argmin}} \sum_{i=1}^{N} d_g(x, x_i)^2 \tag{4.10}$$

where $d_g(.,.)$ is the Riemannian geodesic distance defined in (4.9). A gradient descent approach was proposed in [80, 66] to solve this minimization problem for Riemannian manifolds.

4.3.1.1 Principal Geodesic Analysis

Analogous to the principal components of a vector space, there exist a 1-parameter subgroup called the principal geodesic curves which explain the variability of the data points lying on a manifold around their intrinsic mean.

Under the assumption that the data points lie in a small neighborhood about the mean point μ, it was shown in [34] that solving for the principal geodesic components

boils down to performing PCA for the tangent vectors $log_\mu(x_i) \in T_\mu\mathcal{M}$. This method is called principal geodesic analysis (PGA).

4.4 Activity Classification on Grassmannian

Given several realization for each activity, in the form of the tracks of p feature points, we learn a set of deformation subspaces $\{\mathbf{S^1}, \mathbf{S^2}, \cdots, \mathbf{S^k}\}$ of different deformation order K for each realization as discussed in Section 4.2. For a given deformation order K, the problem of action recognition can be modeled as a multi-category classification problem on the Grassmann manifold $\mathbb{G}(p, 3k)$. Traditional vector space classifiers can not be applied directly due to the curved nature of this manifold, several modification to these classification approaches is needed in order to obey the manifold geometry.

In this section, we present several approaches to solve this classification problem. In the first approach, we use the nearest-neighbor rule for classification. This method has the advantage that it is easily adapted to Riemannian manifolds and does not need to compute any statistical properties of the data for each class. Then we present several approaches for building statistical models on the manifold in order to use Bayesian classification. Finally, We discuss a Bayesian formulation to marginalize the learnt distributions over the variable deformation order K.

4.4.1 Nearest-Neighbor Grassmann Classification

Nearest-Neighbor (NN) rule seems a simple and natural candidate for manifold classification. As it relies only on the choice of a metric function between the different

training points and the test point. Hence, the rule can be simply applied to the Grassmann manifold by choosing the geodesic distance in (4.9) as a distance metric.

Let $D^n = S_1, S_2, \cdots, S_n$ denotes a set of n labeled training deformation subspaces for different actions. For a test sequence with a deformation subspace S, the NN classification rule would assign S to the same action as its nearest training point in D^n in terms of the geodesic distance

$$label(S) = label(\underset{S_i \in D^n}{\operatorname{argmin}}\ d_g(S, S_i)) \tag{4.11}$$

where $label(S)$ refers to the action label corresponding to the training deformation subspace S, and $d_g(.,.)$ is the geodesic distance on Grassmann manifold defined in (4.9).

The NN rule is a suboptimal procedure. In Euclidean vector spaces, it results in classification error that is greater than the minimum Bayes error rate. However, it was proved that with an unlimited number of samples the error rate is never worse than twice the Bayes error rate [23]. This bound is not guaranteed for data points on Riemannian manifolds. However, the nearest-neighbor rule still seems appealing due to its computational simplicity, as we only need to compute the geodesic distance between the deformation subspace of the test sample and each of the training samples.

4.4.2 Bayesian Grassmann Classification on Tangent Plane

The Bayesian classification rule represents the optimal classifier in terms of minimizing the classification error. However this optimality is under the assumption of known prior probabilities of the classes, and known class conditional densities. This assumption seems reasonable in Euclidean vector spaces where we can estimate these densities ef-

ficiently using maximum-likelihood or Bayesian parameter estimation methods. On the other hand, applying these rules into Riemannian manifolds require efficient methods to learn statistical models and class conditional distributions on these spaces. This is not an easy task due to the nonlinearity and high dimensionality of these manifolds. We now present several approaches for learning these conditional densities on the Grassmann manifold. Given these density estimates, a simple likelihood test is used for classification of test sequences.

Building high order statistical models directly on Riemannian manifolds is rather difficult to perform. This is mainly due to the non-linearity of these manifolds. So a common approach is to build the statistical model on the tangent plane of the manifold at some reference point. The tangent plane is a vector space and hence more conventional statistics can be applied. Usually, the reference point is chosen to be the mean point of the set of samples used to learn the distribution.

Given n realizations of an activity represented by n points on the Grassmann manifold $S_1, S_2, \cdots, S_n$, we first compute the Karcher mean of these points μ using (4.10). Then we compute the tangent vectors $A_1, A_2, \cdots, A_n$ representing the directions and the speeds of the geodesic flow from the mean μ to each of the sample points as outlined in (4.7).

As shown in section 4.2, the tangent plane for the deformation subspace Grassmann manifold $\mathbb{G}(p, 3k)$ is a $p(p-3k)$-dim vector space, equipped with the metric $g(.,.)$ presented in (4.8), which is equivalent to the standard inner product in $p(p-3k)$-dimensional Euclidean space. Hence, the problem of estimating the conditional density function is reduced to estimating a density function for the tangent vectors $A_1, A_2, \cdots, A_n$ in $\mathbb{R}^{p(p-3k)}$.

Due to the high dimensionality of the tangent plane, specially for a large number of landmarks p, a large number of samples is needed to estimate the density functions. A common approach is to assume that the variation of the velocity vectors $\{A_i, i = 1 : n\}$ are mostly restricted to m-dimensional subspace of $\mathbb{R}^{p(p-3k)}$, called the principal subspace, where $m << p(p-3k)$. Principal component analysis (PCA) is used to learn the orthogonal basis for that subspaces in the tangent plane. As noted in section 4.2, this is equivalent to calculating the principal geodesic curves of the data points on the Grassmann manifold. We denote the linear projection of A_i to the principal subspace by $\tilde{A}_i$. We still have to decide on the form the probability distribution takes. Following are two different alternatives.

4.4.2.1 Gaussian model for principal coefficients

A common approach is to assume a multivariate normal model on the tangent plane vectors projected into the principal subspace, i.e. we model $\tilde{A}_i \sim \mathcal{N}(O, \Sigma)$, for a covariance matrix $\Sigma \in \mathbb{R}^{m \times m}$.

Estimation of the model parameters μ and K from the observed deformation subspaces is straightforward. We first compute the Karcher mean μ of the samples on the Grassmann manifold as shown in (4.10). Using μ and an observed subspace S_i. find the tangent vector $A_i \in \mathbb{R}^{(p-3k)\times p}$ that specify the direction and speed of the geodesic flow from μ to S_i in unit time. From the observed values of A_i, we estimate the principal subspace and covariance matrix. Extracting the dominant eigenvectors of the covariance matrix, one can capture the dominant m modes of variations. A projection of the tangent

vectors into the principal subspace is denoted by $\tilde{A}$. The density function associated with a samples S is given by:

$$h(S\,;\mu,\Sigma) \propto exp(-\tilde{A}^T\Sigma^{-1}\tilde{A}/2)/(det(\Sigma)) \tag{4.12}$$

Assuming uniform prior for all action classes. The Bayesian classification assigns a test deformation subspace S to the class with the highest likelihood $h(S\,;\mu,\Sigma)$.

An special case of interest occurs when $m = (p - 3k)$ and $\Sigma = I_m$, then the density function would be on the form:

$$h(S\,;\mu,\Sigma) \propto exp(-\parallel A \parallel^2/2) \tag{4.13}$$

In this case, the likelihood-based decision rule is reduced to minimizing the geodesic distance between the test sample and the Grassmann mean corresponding to each action. The maximum likelihood test assigns the test sample to the action with the nearest mean in terms of the geodesic distance.

4.4.2.2 Non-parametric model for principal coefficients

Instead of assuming a parametric model for $\tilde{A}$, we can estimate the probability density function using kernel density estimation. We use a Gaussian kernel, and assume independence among the different dimensions of the principal subspace. This approach can be very efficient in cases where the Gaussian assumption of principal coefficients is not justified. However, as most non-parametric approaches it requires sufficiently large number of training samples for each class.

4.4.3 Deformation Order and Computing the Marginal Likelihood

The deformation order K represents the amount of deformability in the shape sequence for the activity, as it quantizes the number of 3D basis shapes needed to fully reconstruct the deformable shapes corresponding to the different body poses constituting the activity. It also determines the actual Grassmann manifold where the deformation subspaces lie. In all of the above classification approaches, we assume a known and fixed deformation order K for all of the action classes.

In section 3.3, we presented an approach to estimate that factor using the singular values of the data covariance matrix, after performing a whitening process on the noisy observation. This approach was proves successful is estimating the deformation factor, termed as Deformability Index (DI) for different activities. However, it assumes that the noise covariance matrix is either known *apriori* or can be estimated from the data.

We use an alternative procedure in this section instead of estimating a single value for the deformation order. We first compute the conditional distribution of the test sequence $h(S/K = k)$ for different deformation orders K using any of the approaches proposed earlier. We then compute the marginal distribution $h(S)$ using these conditional densities and the prior probabilities $P(K = k)$.

$$h(S) = \sum_{k=1}^{k_{max}} h(S/K = k)P(K = k) \tag{4.14}$$

The values of the prior likelihood $P(k)$ is computed for each test sequence as a function of the ratio of the singular values of the original observation matrix used to compute the deformation subspace.

4.5 Experimental Results

We carried out a set of experiments to evaluate and verify the effectiveness of our proposed approach for activity modeling and classification. We use the motion capture dataset [1] available from Carnegie Mellon University in all of the experiments. The combined dataset includes a number of subjects performing eight different activities such as walking, running, jumping, dribbling, kicking, boxing, and salsa dancing. These activities were picked form the datasets because they have enough number of realizations under different conditions and performers. The total number of sequences used in our experiment was 136 sequences.

The dataset provides 3D landmark tracks for 41 marked landmark points on the human body while performing the action. We used these 3D location to generate 2D projected tracks using arbitrary affine camera projection models. The 2D tracks for each realization were used to generate the deformation subspace corresponding to each activity for different values of the deformation order K.

Figure 4.1 shows the pairwise similarity matrix between the deformation subspaces of all the dataset realizations at different values of the deformation order, with the Grassmann geodesic distance between the subspaces is used as a measure of dissimilarity. The overall block structure of the similarity matrix for various deformation order suggests the effectiveness of using the distances between deformation subspaces for separating different activities.

We also note from Figure 4.1 how the effect of the deformation order varies for different actions. For example, the walking and running sequences are hard to separate at

	K=1	K=2	K=3	K=4	K=5	K=6
Overall	96.32	97.06	95.56	95.56	94.12	89.7
Walking Sequences	97.5	100	100	100	90	95
Running Sequences	100	100	100	100	95.24	85.71
Jumping Sequences	100	100	100	96.15	100	88.46
Dribbling Sequences	83.33	91.67	91.67	83.33	91.67	75
Kicking Sequences	88.89	100	88.89	100	88.89	77.78
Boxing Sequences	100	71.43	100	85.71	85.71	85.71
Dancing Male Sequences	85.71	100	42.86	85.71	100	100
Dancing Female Sequences	100	92.86	100	92.86	100	100

Table 4.1: A summary of the percentage recognition rate results using the Nearest-neighbor classification for different deformation orders K. The first row shows the overall recognition results, and the subsequent rows show the classification results for each action.

$K = 1$, while they are better separable for larger values of K such as $K = 2$ or $K = 3$. This seems intuitive since the two action share a very similar rigid body pose represented by the first order subspace, while higher degrees of deformation will help distinguish between the two activities. On the other hand, as the deformation order K increases over the value needed to capture the actual shape deformation, the effect of noise increases. Specially for actions with low degree of deformability like walking.

4.5.1 Classification Results for Nearest Neighbor Classification

Nearest-Neighbor Classification is performed on the data sequences using the pairwise geodesic distances shown in Figure 4.1 as a metric. The overall success classification rate results for different deformation order K are shown in Table 4.1. The table also shows the breakdown of the classification rates for different actions at different value of the deformation orders K.

The results show good recognition capabilities based on the NN classification rule,

with a maximum correct recognition rate of 97% at $K = 2$. The number of sequences per action plays a crucial rule in the success of NN classification. For that reason, actions with limited number of samples like boxing and dribbling can have fluctuating recognition results as compared to walking and running. An interesting observation is that at low order of deformation, many misclassification occur between the dancing male and dancing female sequences which share a very close set of body pose deformations. However, we note that this misclassification is resolved as we move to higher deformation orders $K = 5$ and $K = 6$.

4.5.2 Classification Results for Bayesian Classification

Although the performance results achieved using the nearest-neighbor rule was promising, it can vary significantly with increasing the number of classes. NN rule also suffers from sensitivity to outliers, and does not provide a theoretical framework to combine results from different Grassmann manifolds corresponding to different deformation orders. For these reason, we performed another set of experiments using the Bayesian framework formulation presented in section 4.4.2.

Due to the limited number of samples for each class in our dataset, we conducted the experiments in two different methodology. In the first set of experiments, we compute baseline performance measure using all the sequences in the dataset for training. Although this results in an overlap between the training and testing data samples, it provides a baseline performance measure to identify any possible problems. In the second set of experiments we use a leave-one-out cross evaluation method, where we test each

sequence using a model learnt using all the other sequences.

In all of these experiments, we compute the Grassmann mean of the different training deformation subspaces of each action and the tangent vectors corresponding to the geodesic flow from this mean to each training sample. We fit the different distribution models presented in Section 4.4.2 to these tangent vectors and estimate the parameters for the likelihood function. A likelihood test is then performed to assign test samples to the maximum likelihood action model.

4.5.2.1 Gaussian model for tangent vectors

In this experiment we model the tangent vectors with a multivariate density distribution with identity covariance matrix in the $p(p-3k)$-dim tangent plane. In this special case, the density function takes the form shown in (4.13), and the maximum likelihood test is reduced to simply computing the geodesic distance between the test sample and each of the learnt action mean subspaces. The only data statistics that are needed for that model are the mean shape of the training subspaces.

Figure 4.2 shows the log-likelihood value for every sequences-model pair using the leave one out cross validation method. The recognition rates using these likelihood values are shown in Table 4.2. We can note that the recognition results varies significantly with the deformation order for different actions, with maximum recognition rate of 89.71%. The different distribution models are combined as shown in Section 4.4.3 to compute the marginal distribution over K. The log-likelihood values and the action confusion matrix using this marginal distribution are shown in Figure 4.3. The figure clearly shows

Validation Method	K=1	K=2	K=3	K=4	K=5	K=6	Marginal
All Sequence	83.82	95.59	95.59	95.59	83.09	93.38	97.79
Leave One Out	83.82	89.71	89.71	87.5	73.53	82.35	97.05

Table 4.2: A summary of the percentage recognition rate results using the Gaussian model for tangent vector for Bayesian classification for different deformation orders K and for the model learnt by marginalizing over K

that combining the different deformation order distributions results in more separable likelihood for different actions. This is reflected in an increase in the recognition rate to 97.05%.

4.5.2.2 Multivariate Gaussian model for principal coefficients

In this experiment, we wanted to test whether estimating more statistical variation of the signal can result in a better classification rates. In order to do this, we first reduce the dimensionality of the tangent plane by computing the PCA subspace of the training tangent vectors, and project the tangent vectors into that subspace. We fit a multivariate Gaussian model to these PCA coefficients. Tables 4.3 and 4.4 show the recognition rates using a Gaussian model and a mixture of two Gaussian respectively. The action confusion matrix for the two models are shown in Figure 4.4 for the marginal over deformation order K models.

One of the problems with this parametric model is the need for a sufficient number of samples to obtain a reliable estimate of the parameters and the principal subspace. For our dataset, some actions like boxing had only five training samples, which results in a very unreliable estimates of the model parameters. In spite of this, we were able to obtain reasonable recognition rates of 86.76% and 88.97%. We would assume that using more

data sequences can improve these models significantly.

Validation Method	K=1	K=2	K=3	K=4	K=5	K=6	Marginal
All Sequences	94.12	88.24	77.21	72.79	38.24	41.18	94.85
Leave One Out	86.76	72.79	68.38	61.76	34.56	41.91	86.76

Table 4.3: A summary of the percentage recognition rate results of Bayesian classification using a multivariate Gaussian model for the principal coefficients of tangent vector at different deformation orders K, and for the marginal distribution over K.

Validation Method	K=1	K=2	K=3	K=4	K=5	K=6	Marginal
All Sequences	96.32	94.12	91.18	86.03	47.06	49.26	98.53
Leave One Out	88.24	73.53	71.32	60.29	31.62	41.18	88.97

Table 4.4: A summary of the percentage recognition rate results of Bayesian classification using a mixture of Gaussian model with two components for the principal coefficients of tangent vector at different deformation orders K, and for the marginal distribution over K.

4.5.2.3 Non-parametric model for principal coefficients

In this last set of experiments, instead of assuming a specific parametric form, we use a non-parametric kernel model with Gaussian kernel to learn the distribution of the principal coefficients. The recognition results for this model is shown in Table 4.5, while the likelihood and confusion matrix for the marginal distribution are shown in Figure 4.5.

Validation Method	K=1	K=2	K=3	K=4	K=5	K=6	Marginal
All Sequences	94.12	93.38	86.03	85.3	45.59	50.73	96.32
Leave One Out	87.5	75.73	66.17	63.24	33.09	38.97	86.76

Table 4.5: A summary of the percentage recognition rate results of Bayesian classification using a non-parametric model for the principal coefficients of tangent vector at different deformation orders K, and for the combined k.

4.6 Chapter Summary

In this chapter, we presented an approach for activity recognition and classification via modeling activity deformation subspaces on Grassmann manifold. The proposed approach captures the underlying deformation of the body corresponding to the different body poses that constitute the activity. Meanwhile it maintains affine view invariance representation. The deformation subspaces for each activity is estimated from the motion trajectories of the points participating in the activity under an affine camera projection model. We formulated the problem of modeling and classifying the deformation subspaces as a classification problem on the Grassmann manifold. We proposed several classification models that captures the variability of the activity samples while following the manifold intrinsic geometry. We presented results showing the effectiveness of our 3D model in representing human activity for recognition and classification.

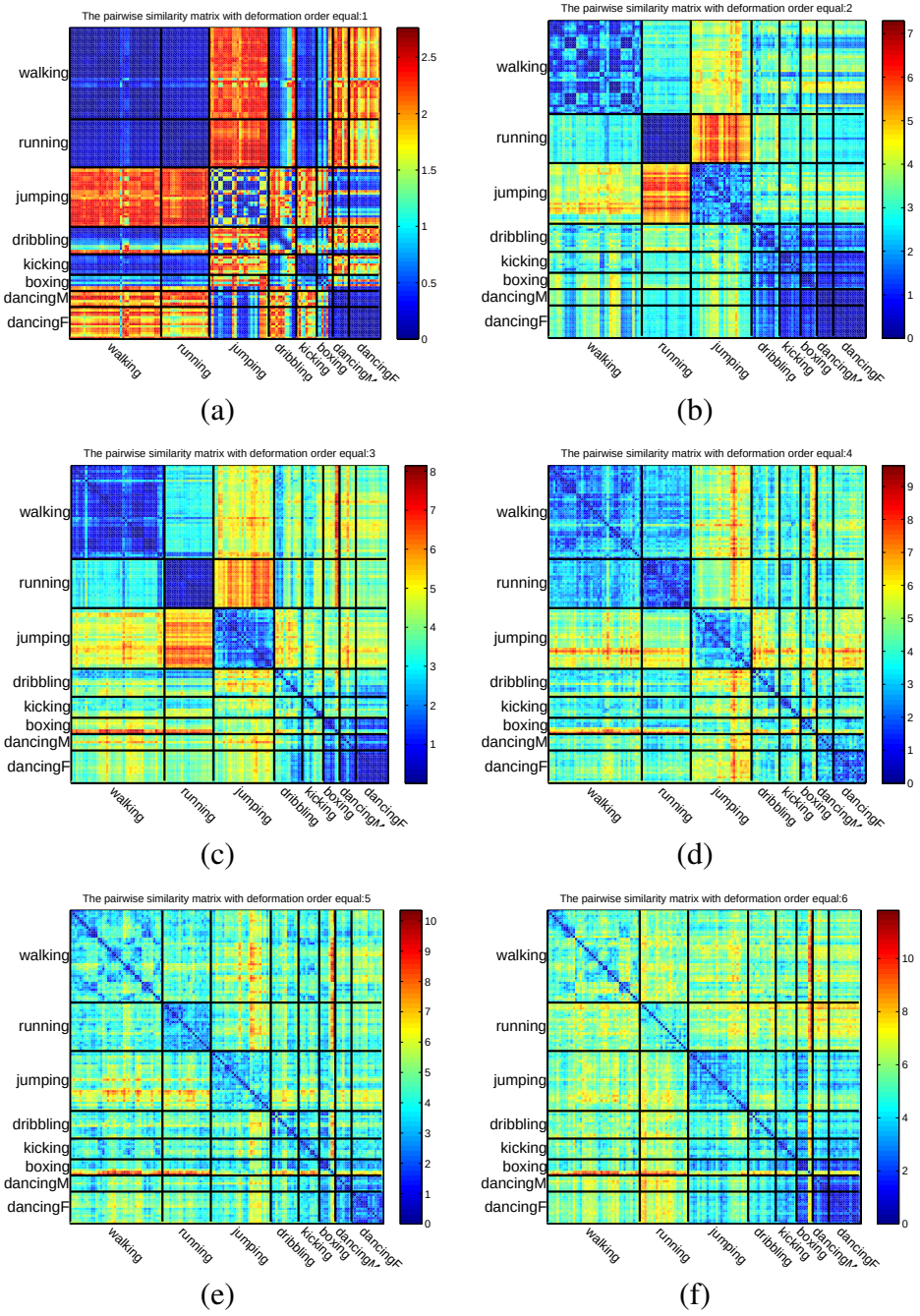


Figure 4.1: The 136 X 136 Grassmann pairwise similarity matrix between all of the different sequences used in the experiments for orders of deformation 1 to 6 shown in a-f respectively. Warmer color represent larger geodesic distances. Black separation lines are added between sequences corresponding to different action for easier analysis. (This figure is best viewed in color)

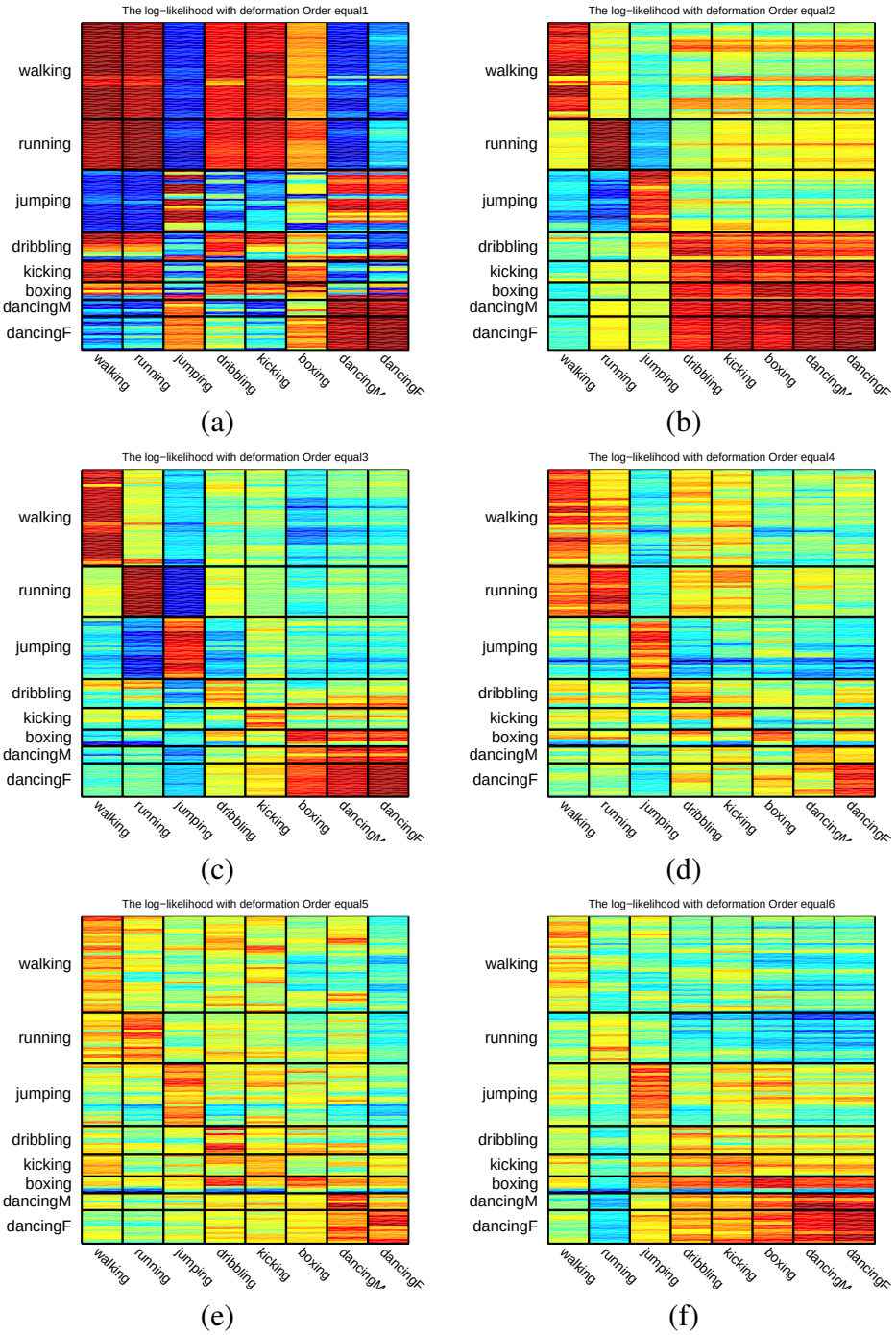
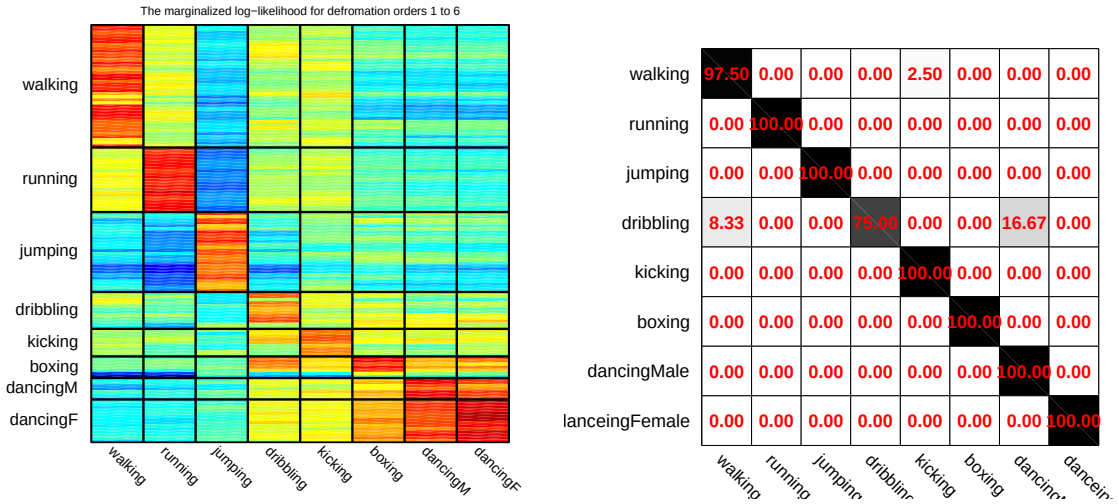
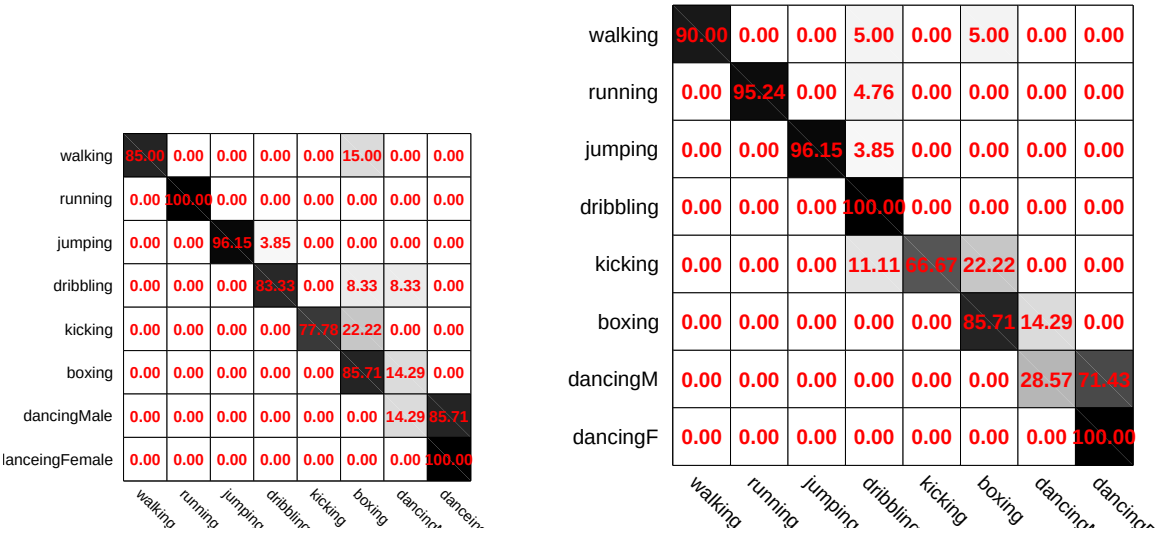


Figure 4.2: The log-likelihood values at different deformation orders using the Gaussian model for tangent vectors and for leave-one-out cross validation. (This figure is best viewed in color)



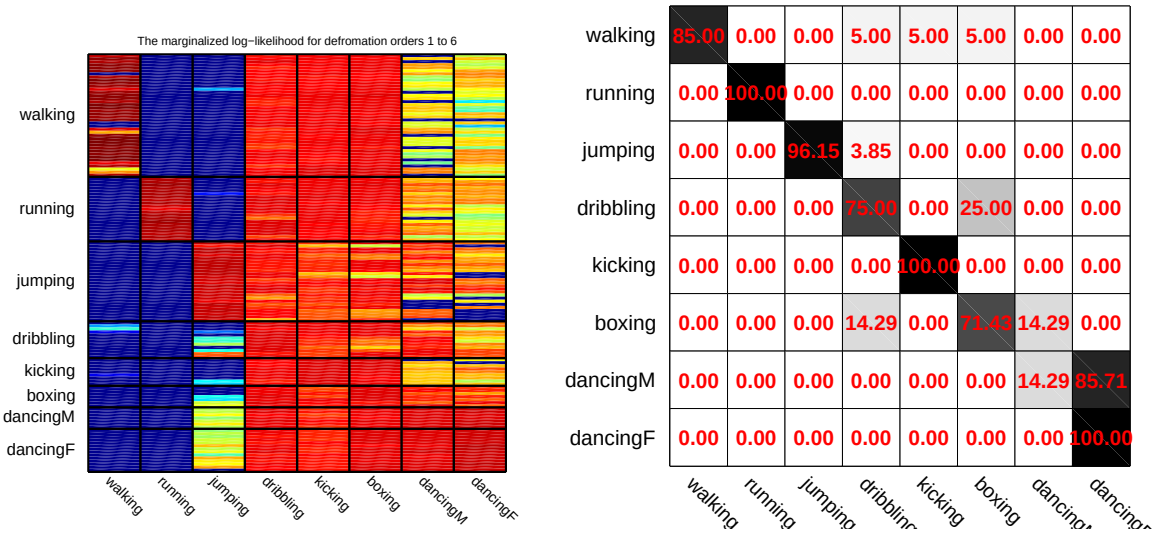
(a) (b)

Figure 4.3: The performance measures for using the Gaussian model for tangent vectors after computing the marginal distribution over K. Using leave-one-out cross validation. a) The log-likelihood values. (b)The action confusion matrix.



(a) (b)

Figure 4.4: The action confusion matrix for using the Gaussian model for the principal coefficients after computing the marginal distribution over K. Using leave-one-out cross validation. (a) Using a single Gaussian Model. (b)Using a Mixture of two Gaussians.



(a) (b)

Figure 4.5: The confusion matrix for action classification using the non-parametric model for the principal coefficients of the tangent vectors. (a) Using all the sequences for training. (b) Using Leave-one-out cross validation.

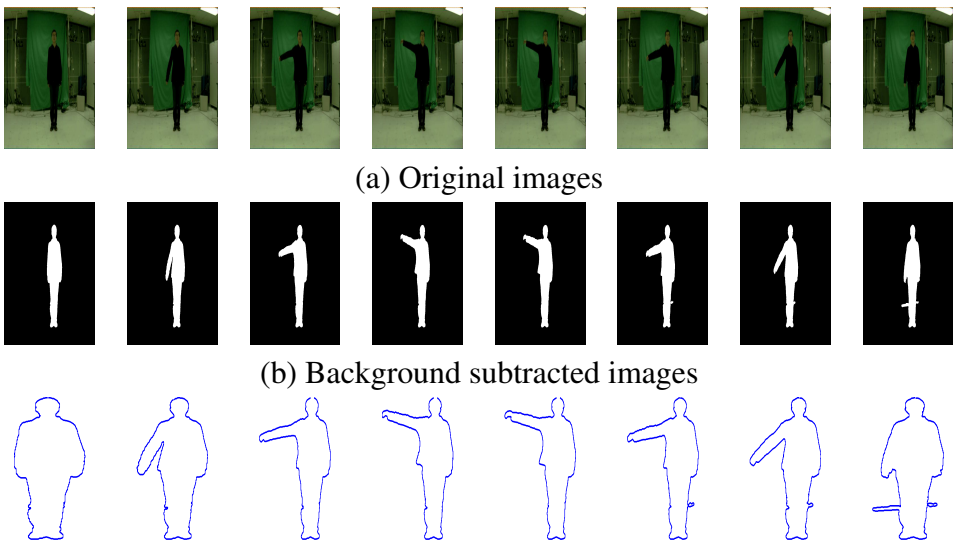
Chapter 5

Silhouette-based Gesture and Action Recognition via Modeling Trajectories on Riemannian Shape Manifolds

5.1 Introduction

The problems of modeling and recognition of human gestures and actions from a sequence of images have received considerable interest in the computer vision community, as one of the many problems that aim at achieving a high level automated understanding of video data. This interest is motivated by many applications in different areas in human-computer interaction [46], robotics [11], security, and multimedia analysis.

Many of the existing approaches for gesture recognition model gestures as a temporal sequence of feature points representing the human pose at each time instant. The choice of these features usually depends on the application domain, image quality or resolution, and computational constraints. Features such as exemplar key frames [126], optical flow [31] and feature points trajectories [96, 17] have been frequently used to represent the raw, high-dimensional video data in several approaches. The major challenge of most of these features is that they require highly accurate low-level processing tasks such as tracking of interest points. This accuracy turns out to be very hard to achieve in gesture recognition scenarios because of fast articulation, self occlusion, and different resolution levels that are encountered in different applications.



(a) Original images

(b) Background subtracted images

(c) Contour curve representation

Figure 5.1: An example of different frames of a humans performing a ”Turn Right” control gesture. (a) The original video frames (b) The background subtracted images and (c) The sequence of contours. Most of the information about the gesture can be modeled using only the contour curves

In order to overcome this limitation, silhouette-based approaches have been receiving increasing attention recently [13, 119, 123]. These approaches focus on the use of the shape of the binary silhouette of the human body as a feature for gesture recognition. They rely on the observation that most human gestures can be recognized using only the shape of the outer contour of the body as shown in Figure 5.1 for a ”Turn Right” control gesture. The most important advantage of these features is being easy to extract from the raw video frames using object localization and background subtraction algorithms, which are low-level processing tasks and relatively higher accuracy can be achieved in these tasks under different conditions.

An important question in silhouette-based approaches is: how can we represent the shape of these silhouettes in an efficient and robust way? Several shape representation features have been used in the literature for this purpose, including chain codes [36],

Fourier descriptors [133], shape moments [50], and shape context [10]. For most of these features, the feature vector is treated as a vector in a Euclidean space in order to use standard vector space methods for modeling and recognition. This assumption is not usually valid as these feature lie in a low-dimensional, non-Euclidean space. Working directly on these spaces can provide models and discriminative measures that may result in an improved performance. One way to explore this lower-dimensional space, is to try to learn its structure from the training data using dimensionality reduction techniques combined with a suitable notion of local discriminative measure between visual data features. This technique was recently used [33, 12, 123] for human action recognition and pose recovery. The problems with this technique come from the limitations of data driven manifolds such as a lack of robust statistical models, and the difficulty in extrapolation and matching of new data.

The limitations of data driven manifolds methods noted above have shifted the attention of many computer vision researchers towards the use of analytic manifold theory. This shift was also supported by the fact that many features in computer vision lie on curved space because of the geometric nature of the problems. Several of these manifolds were used in problems like object detection and tracking [114], affine invariant shape clustering [9], and activity modeling [112]. The use of such manifolds offers a wide variety of statistical and modeling tools that arise from the field of differentiable geometry. These tools have found applications in problems such as target recognition [43], parameter estimation [98], clustering and dimensionality reduction [40], classification [115], and statistical analysis [35, 112].

The choice of the right feature and space to model the shape of the silhouettes is

not the only issue in silhouette-based methods. An equally important problem is the efficient modeling of the dynamics of temporal variations of these feature as the gesture progresses. The importance of both shape and dynamic cues for modeling human movement was noted and demonstrated experimentally in [118]. Various models were used for modeling these dynamics, ranging between statistical generative models [84, 130, 16, 49], and the more recent discriminative models [125, 124]. The invariance to temporal rate of execution of action in such models is crucial for achieving accurate recognition [120].

5.1.1 Motivation and Overview of Approach

In this chapter, we explore the use of shape analysis on manifolds for human actions and gesture recognition. Our approach falls into the category of the silhouette-based approaches described earlier. Each silhouette is represented by a planar closed curve corresponding to the contour of this silhouette, and we are interested in evolving shapes of these curves during actions and gestures. We will use a recent approach for shape analysis [56, 57, 101], that uses differential geometric tools on the shape spaces of closed curves. Similar ideas have also been presented in [62, 131, 132]. While there are several ways to analyze shapes of closed curves, an elastic analysis of the parameterized curves is particularly appropriate in this application. This is because: (1) the elastic matching of curves allows nonlinear registration and improved matching of features (e.g. body parts) across silhouettes, (2) this method uses a square-root representation under which the elastic metric reduces to the standard $\mathbb{L}^2$ metric and thus simplifies the analysis, and (3) under this metric the re-parameterizations of curves do not change the Riemannian

distances between them and thus help remove the parametrization variability from the analysis. Furthermore, such geometric approaches are useful because they allow us to perform intrinsic statistical analysis tasks, such as shape modeling and clustering, on such Riemannian spaces [100].

Using a square-root representation of contours, each human gesture is transferred into a sequence of points on the shape space of closed curves. Thus, the problem of action recognition becomes a problem of modeling and comparing dynamical trajectories on the shape space. We propose two different approaches to model these trajectories.

In the first approach, we propose a template-based approach to learn a unique template trajectory representing each gesture. One of the main challenges in template-based method is to account for variation in temporal execution rate. To deal with this problem, we use a modified version of the DTW algorithm to learn the warping functions between the different realizations of each gesture. We use the geodesic distances on the shape space to match different points on the trajectories in order to learn the warping functions. An iterative approach is then used to learn a mean trajectory on the shape space and to compute the temporal warping functions.

In the second approach, we utilize the geometry of the shape space more efficiently in order to cope with the different variations within each gesture caused by changes in execution style, body shape and noise. Each gesture is modeled as a Markov model to represent the transition among different clusters on the shape space of closed curve. We learn these models by decoupling the problem into two stages. In the first stage, we cluster the individual silhouette shapes using the Affinity Propagation (AP) clustering technique [38], and build statistical model of variation within each cluster. In the second stage, a

hidden Markov Model (HMM) is used to learn the transition between different clusters for each gesture.

Extensive experiments were conducted to test the performance of our algorithms. We used two different data sets of video sequences representing different control gestures and regular actions, with a total of 226 video sequences. The data sets contained many variations in terms of the number of subjects, execution styles, and temporal execution rates.

5.1.2 Contributions

Our contribution in this work can be summarized as

1. Posing the problem of gesture and action recognition as one of classifying the trajectories on a Riemannian shape space of closed curves.

2. Proposing a template-based model and a Markovian graphical model for modeling the time series data of points on the shape manifold. These models were designed to fully adhere to the geometry of the manifold and to model the statistical variations of the data on this manifold.

3. Provide a comprehensive set of experimental analysis of the proposed models on two different datasets for gesture and action recognition.

5.1.3 Organization

The reminder of this chapter is organized as follows: in Section 5.2, we describe the square-root representation of closed curves and the resulting shape space. We also give a

brief overview of the computation of distances and statistics on this manifold. Section 5.3 describes the two dynamical model approaches used for gesture modeling. Experimental results validating the proposed method for human gesture recognition are introduced in Section 5.4. The chapter is concluded with a summary in Section 5.5.

5.2 Manifold Representation of Silhouettes

As mentioned earlier, we will use the square-root elastic representation [56, 57] to construct a shape space of closed curves in $\mathbb{R}^2$. Under this framework, the contour of the human silhouette in each frame represents a point in the shape space and each gesture represents a temporal trajectory on that space. We then use principles from Riemannian geometry combined with the structure of the shape space to build statistical models for these trajectories for representation and recognition.

We represent the shape of a contour curve β, parameterized arbitrary, by its square-root velocity function: $q : \mathbb{S}^1 \rightarrow \mathbb{R}^2$, where

$$q(t) = \frac{\dot{\beta}(t)}{\sqrt{\|\dot{\beta}(t)\|}}. \tag{5.1}$$

where $\|.\|$ is the Euclidean norm in $\mathbb{R}^2$, and t is an arbitrary coordinate on $\mathbb{S}^1$. The quantity $\|q(t)\|$ is the square-root of the instantaneous speed, and the ratio $q(t)/\|q(t)\|$ represents the instantaneous direction along the curve. The curve β can be recovered within a translation, using $\beta(t) = \int_0^t \|q(s)\| \, ds$. Let C be the set of all unit-length, closed planar curves that are represented by their square-root velocity functions. If we do not impose the closure condition then this set is a Hilbert sphere; with the closure constraint C is a submanifold of the unit sphere. C is called a pre-shape space.

The geodesics between points in $\mathcal{C}$ are computed using a path-straightening approach. In this approach, first introduced in [61], the geodesic path between the two points is first initialized with an arbitrary path. Then, this path is iteratively straightened using a gradient approach and the limit point of this algorithm is a geodesic path. To be effective for shape analysis, the representation and the geodesics between the points must be invariant to shape-preserving transformations. These transformations include pose (rotation, scale, and location) and the parameterizations of the curves. Note that the square-root velocity representation is already invariant to scale and translation. The remaining two are studied as groups acting on $\mathcal{C}$: rotation by the action of $SO(2)$ and re-parametrization by the action of $\mathcal{D}$, where $\mathcal{D} = \{\gamma : \mathbb{S}^1 \to \mathbb{S}^1\}$ is the space of all orientation-preserving diffeomorphisms. The resulting (transformation invariant) shape space $\mathcal{S}$ is defined as a quotient space as follows

$$\mathcal{S} = \mathcal{C}/(SO(2) \times \mathcal{D}). \tag{5.2}$$

Elements of this quotient space are equivalence classes of the type:

$$[q] = \{O(q \circ \gamma)\sqrt{\dot{\gamma}} | O \in SO(2),\ \gamma \in \mathcal{D}\}\ .$$

These are the orbits of $SO(2) \times \mathcal{D}$ in $\mathcal{C}$ and each such orbit represents a shape uniquely. The problem of finding the geodesics between two shapes, or two orbits, relies on first solving the optimization problem:

$$d([q_0], [q_1]) = \min_{O \in SO(2), \gamma \in \mathcal{D}} d(q_0, O(q \circ \gamma)\sqrt{\dot{\gamma}}). \tag{5.3}$$

This is iteratively solved using a gradient descent approach that finds these optimal transformation parameters. Once the optimal rotation and re-parameterization are obtained,

one can draw a geodesic path between q_0 and $O^*(q \circ \gamma^*)\sqrt{\dot{\gamma}^*}$ in C using path straightening. This results in a geodesic path between the orbits $[q_0]$ and $[q_1]$ in the shape space $\mathcal{S}$. The reader is referred to [57, 62] for more details on algorithms for finding shape geodesics. In the remainder of the paper, we will use q to denote the equivalence class $[q]$ to reduce notation.

In order to build models for gesture and action recognition on the described shape space, one needs to build statistical tools to characterize the different distribution of the data samples and to cope with random noise and errors in low-level processing. The simplest statistical measure, represented by the mean of a group of points in $\mathcal{S}$ was presented in [62]. It uses the Karcher mean formulation as presented in (2.2), to compute the intrinsic mean of a set of points in $\mathcal{S}$.

Higher order statistical models for shapes in $\mathcal{S}$ were presented in [100]. In that approach, all the sample points are projected to the tangent plane $T_\mu(\mathcal{S}) \in \mathbb{L}^2$, where μ is the mean shape of these sample points. In order to overcome the infinite-dimensionality of $\mathcal{S}$, they propose to approximate the tangent function $g \in T_\mu(\mathcal{S})$ by a finite basis; in this case this finite basis comes from Fourier analysis. They further reduce the dimensionality on this vector space by assuming that the data lies on a lower dimensional subspace, and use PCA on the tangent plane to learn this subspace and project all the points into it. The projected points on the low-dimensional subspace can then be modeled either in a parametric form as a multi-variate normal distribution, or in a non-parametric form using kernel density function. We refer the reader to [100] for more details and experimental analysis.

5.3 Modeling Gesture Dynamics

Using the described shape model for closed curves, the dynamic sequence of shapes corresponding to a particular gesture or action will correspond to a sequence of m points on the form

$$Q = q_1, q_2, \cdots, q_m, \tag{5.4}$$

where $q_i \in \mathcal{S}$ for $i = 1, 2, \ldots, m$, and $\mathcal{S}$ is the quotient shape space described in section 5.2. We propose to use the trajectories on the shape space corresponding to these sequence as a feature for modeling and recognition of different gestures. Because of the special nature of the curved shape space, vector space methods can not be directly applied. Hence, a modified versions of these methods that obey the space geometry is proposed.

In this section, we present two approaches for modeling and recognition of these trajectories. The first is a template based non-parametric approach, where a template model is learned from different instances of the same gesture. In order to account for variations in the execution rates within different realizations, we apply an iterative approach utilizing the DTW approach [83] to align the different trajectories temporarily and then learn an average template representation on the manifold for recognition. In the second approach, we build a parametric model by clustering the different points on the manifold into several clusters using the Affinity Propagation (AP) clustering technique [38]. Each gesture trajectory is modeled as a sequence of Markovian jumps between some of these clusters. The transition probabilities of the Markov model are learnt using the standard forward-backward approach [84], while non-parametric statistical observation models are directly built on the Riemannian manifold.

In each of these approaches we address two different stages of the problem:

1. The learning problem: Given N labeled realization $Q_1, Q_2, \cdots Q_N$ of a gesture, where each realization is a time series of shape points in the form shown in (5.4), we would like to learn a nominal (or average) trajectory model of this gesture.

2. The classification problem: Given a test sequence Q, and M different gesture template models. How can we classify the test sequence into one of these models?

5.3.1 Template based Model

In this model, we learn a template trajectory Q^* for each gesture given the training realizations of this gesture. A usual choice is to compute Q^* as the average of all the training trajectories at each time instants. This choice is optimal in terms of minimizing the sum of distances between the template and all of the training examples at each time instant.

This approach has two limitations. The first is the fact that trajectories are not of the same length due to variations in execution rates of the same gesture. This also results in a lack of temporal synchronization between different segments of these trajectories. Therefore, a temporal alignment of these trajectories is crucial before computing the template. The second issue is that all of the points on the different learning trajectories reside on the shape space $\mathcal{S}$. Thus we need to make sure that both the temporal matching of shape points and computation of the template Q^* follow the underlying geometry of that manifold.

In order to solve the temporal variation problem, we propose using the DTW algo-

rithm [83], which is used to find the optimal non-linear warping function to match a given time-series to a template while adhering to certain restrictions such as the monotonicity of the mapping in the time domain. This approach is very popular in speech recognition and has been used in several vision applications such as shape averaging [69], and rate-invariant activity recognition [120]. The optimization process is usually performed using dynamic programming approaches given a measure of similarity between the features of the two sequences at different time instants.

Adapting the DTW algorithm to features that reside on Riemannian manifolds is a straightforward task, since DTW can operate with any measure of similarity between the different temporal features. Hence, we use the geodesic distance between the different shape points $d(q_i, q_j)$ given in (5.3) as a distance function between the shape features at different time instants.

5.3.1.1 Iteratively learning the template trajectory on the manifold

Given the N labeled realization $Q_1, Q_2, \cdots Q_N$ of a certain gesture, we solve for the nominal shape sequence $Q^* \in \mathcal{S}$ that has the minimum distance to all the N realizations, while accounting for the different non-linear temporal warpings between these realization and Q^*. This can be formulated as computing the average trajectory on the shape space $\mathcal{S}$ of all the warped versions of the N realizations as described next.

$$Q^* = AVG_{\mathcal{S}}(f_1(Q_1), f_2(Q_2), \cdots f_N(Q_N)). \tag{5.5}$$

where, $AVG_{\mathcal{S}}$ stands for the average of the realization computed with respect to the shape manifold $\mathcal{S}$, and $f_i(Q_i)$ represent the non-linear warped function that was calculated using DTW to match Q_i to the template Q^*.

According to (5.5), given a set of warping functions $f_i, i = 1 : N$, we can solve for Q^*. This is achieved by computing the intrinsic mean on $\mathcal{S}$ of all the warped sequences at each time instants. We use the Karcher mean on the manifold as described in Section 2.2.1 for this computation. However, a template Q^* is needed in order to compute these warping functions.

We solve this dual problem in an iterative manner as shown in Algorithm 1 by iterating between computing the non-linear warping functions to a given template and updating the template as the average of the warped realizations.

The initialization of this algorithm is performed by choosing one of the N labeled realizations as the initial Q^*. This choice can be done randomly. However, we found that better performance can be achieved if we choose the sequence that best represents all of the N labeled realization. This is the sequence that achieves the minimum average distance with all the labeled realizations when picked as a template and all the sequences are warped to it.

5.3.1.2 Classification of a test sequence

In the testing stage, we are given test shape sequence Q_t and M different gesture templates. We want to classify Q_t as one of these gestures. This classification problem is solved using the nearest neighbor rule, by warping all the models to Q_t using the geodesic

Algorithm 1 Learning gesture template Q^* from N labeled realization $Q_1, Q_2, \cdots Q_N$

1: Initialize Q^* to one of the N realizations

2: **repeat**

3: **for** n=1 to N **do**

4: Find DTW to warp Q_n to Q^*

5: **end for**

6: Update Q^* as the Karcher mean of all the N warped realizations

7: **until** Convergence or t times

distance in $\mathcal{S}$ as a distance measure and computing the total warping distance to each of these models. Q_t is then assigned to the model with the smallest warping distance.

5.3.2 Graphical-based Statistical Model

The simplicity of template-based approaches encouraged researchers to use it in many practical applications, where they have shown great success under conditions of small variations and low noise within the training data. Under these conditions, the mean can be sufficient to characterize the time-series. Unfortunately, in many other scenarios, these conditions do not hold. Hence, we need higher order statistical models to capture the variation within the time-series data.

Graphical models such as the Hidden Markov Models (HMM) have proven more effective than template-based methods in modeling time-series data, specially in areas such as speech processing [84]. This is so because these models can learn more information about the special statistical variation of both the observation data and the dynamical

transition between different states.

A standard HMM with continuous observation is shown in Figure 5.2. This is a statistical generative model that represent the time series of observations in the form of a set of transitions between several abstract hidden states, where the state at time t is denoted by S_t. The transition between these states is governed by a Markovian model parameterized by the state transition probability $P(S_t|S_{t-1})$. The observation at each time instant q_t is statistically dependent on the state S_t according to an observation probability density function $f(q_t|S_t)$. A common choice for this density function is a mixture of Gaussians. The model parameters are learnt using the standard Baum-Welch algorithm [8]. This ensures that the state transition and observation density functions are optimized in the maximum likelihood sense.

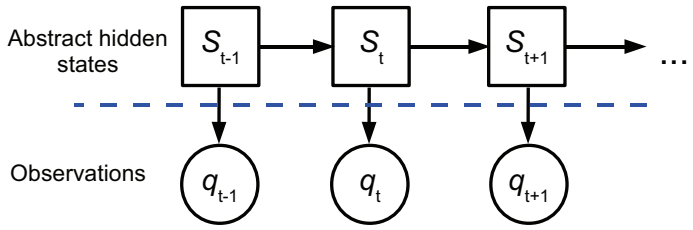


Figure 5.2: A graphical model of unfolding a standard continuous state HMM

To the best of our knowledge, most of the published work on graphical models such as HMM studies modeling time-series data on Euclidean spaces. However, several challenges appear if we want to generalize these methods to time-series data on special manifolds. One of the challenges is that in order to solve for the optimal parameters in the maximum likelihood sense, we need to provide an analytical form for the observation density functions and compute the gradient of the likelihood of a sequence in terms of the

parameters of these functions.

In order to avoid the mathematical difficulty of directly learning a graphical model on the Riemannian manifold, we propose to decouple the two problems of learning the abstract state transitions and learning the observation distributions. This is accomplished by introducing a new layer of intermediate observations X_t in the form illustrated in Figure 5.3. The resulting model in this case is similar to the decoupled exemplar-based HMM presented in [32]. The difference is that we will learn these exemplars to represent different clusters of data on the shape manifold. We will also learn the observation density functions as a non-parametric density function for each cluster on that manifold.

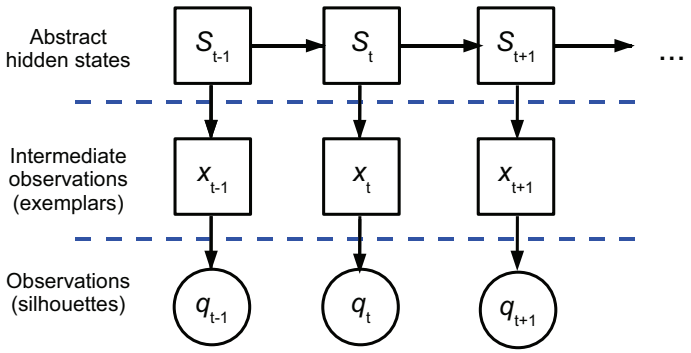


Figure 5.3: A graphical model of unfolding the exemplar-based HMM used in modeling the gesture dynamics

In our approach we model the dynamics of each gesture as a Markov model of L hidden abstract states $H^{(1)}, H^{(2)}, \cdots, H^{(L)}$. Let S_t denotes the state of the system at time t. The state of the system is emitting an intermediate discrete observation representing the cluster (exemplar) of the system and denoted by x_t. This observation can take any value in the set $\mathcal{X} = \{x^k | k = 1, \cdots K\}$, where K is the number of the learnt clusters on the shape manifold. Hence, the system dynamics is determined by the state transition $P(s_t|s_{t-1})$, and

the exemplar observation probabilities $P(x_t|s_t)$.

The final observation q_t corresponds to the contour curve velocity function for a certain frame, and represents a point on the shape space $\mathcal{S}$. These observations are modeled using a mixture distribution such that the observation probability for an observation q_t at time t given the state of the system s_t can ba calculated as

$$f(q_t|s_t) = \sum_{k=1}^{K} f(q_t|x_t = x^k)P(x_t = x^k|s_t) \tag{5.6}$$

where, $f(q_t|x_t = x^k)$ represents the non-parametric probability density function for cluster k on the shape space $\mathcal{S}$.

Learning the model for each gesture includes learning the state transition probabilities $P(S_t|S_{t-1})$, the exemplar clusters observation probabilities $P(x_t|S_t)$, and the shape observation density functions given each cluster $f(q_t|x_t = x^k)$. We follow a decoupling approach for learning these parameters. We first cluster the gesture shape points into several clusters and build a non-parametric statistical model for the data points in each cluster to represent the observation density function. The state transition and cluster observation probabilities are then learnt from the training samples using the standard forward-backward technique commonly used in applications that use discrete observation HMMs.

5.3.2.1 Clustering of gesture shapes

The clustering of the shape points to learn the exemplar clusters can be done directly on the quotient shape space $\mathcal{S}$ by using a stochastic simulated annealing approach to search for the optimal cluster assignment in the configuration space [100]. However,

this method requires a high computation burden and is not guaranteed to converge to a globally optimal solution. We use a simpler and computationally efficient AP clustering approach on the matrix of the pairwise geodesic distances to find the cluster assignment.

Affinity Propagation is an unsupervised clustering technique presented by Frey and Dueck in [38]. It finds a set or exemplar points to represent the data and assigns every data point to one of these exemplars. The search for these exemplars is performed in an iterative manner using a message passing algorithm on a graph, where each data point is represented by a node on this graph. This method was proven superior to other clustering approaches in many applications such as clustering images of faces, detection of genes in micro-array data, and identification of cities that are efficiently accessed by airline travel. It was also shown to be more computationally efficient in terms of processing time and handling sparse data.

There are two main advantages of using AP for our problem. The first is that this method only requires a notion of similarity between every pair of data points and does not assume a Euclidean space. Therefore, we can use the negative of the geodesic distance between the shape points as a geometrically accurate similarity measure, resulting in accurate clustering on the shape manifold. The second advantage is that this method does not require any prior knowledge of the number of clusters and the final clustering is not dependent on the initialization as in K-means clustering.

Given all of the training data shape points, we run the AP algorithm on the matrix of the pairwise similarity between shapes, calculated as the negative of the geodesic distance between theses shapes on the shape space $\mathcal{S}$. The output of the algorithm is a set of K exemplar shapes $q^1, q^2, \cdots, q^K$, and an assignment of each data point to one of

these exemplar. An example of these exemplar shapes for different clusters of the control gesture data set is shown in Figure 5.4. We observe that some similar shapes may be split between two close clusters. This may be a problem with clustering. However, our HMM dynamic model is robust to these errors as each state can be associated with many exemplar states.

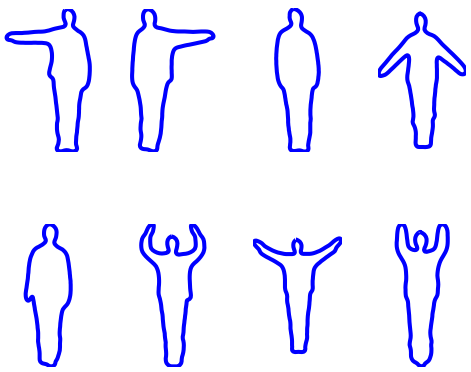


Figure 5.4: Some exemplar shapes representing clusters computed on the shape manifold using AP.

An important question in this step is the amount of data to use for learning these exemplar clusters. We have two choices. The first is to make these clusters gesture dependent by using the shape data corresponding to each gesture to learn a different set of clusters. The second choice is to perform the clustering on all the training data for all the gestures which result in a shared set of exemplars for all the different HMM's. Although the first choice may reduce the computational complexity as we need to calculate significantly fewer geodesic distances, we adopt the second approach for the following reasons. First, the physical nature of the problem suggests that usually many gestures share a set of poses which are similar or very close in shape. The second reason is that global clustering

results in more data points per cluster which helps in providing a better estimate for the non-parametric observation density functions. Finally, the global clustering approach has also the intuitive idea of representing the whole manifold as a mixture of distributions representing our observation model. Our experimental analysis also proves the superiority of global clustering approach as will be shown in the result section.

5.3.2.2 Building the observation model

Given the clustering result of the AP algorithm, we learn a non-parametric density function $f(q_t|x_t = x^k)$ for each cluster x_k. This density function captures the variability between the different samples of the cluster and determines the main mode of variation within each cluster. However, the learning of these density functions on the shape space is difficult, mainly because of the two problems of nonlinearity and infinite dimensionality of the shape space $\mathcal{S}$.

We deal with these challenges using the method described in [100] and summarized in Algorithm 2. The problem of nonlinearity is avoided by building these distributions on the tangent space $T_{\mu^k}\mathcal{S}$ to $\mathcal{S}$ at the mean shape of the cluster μ^k rather than on $\mathcal{S}$ itself. This approximation is valid because the data within the cluster are very close to each other and will be in a small neighborhood around μ^k. Therefore, for every shape $q \in \mathcal{S}$ in the k^{th} cluster we compute a tangent vector $g \in T_{\mu^k}\mathcal{S}$, where g represents the initial velocity of the geodesic path between the mean shape μ_k and the shape q.

The problem of dimensionality is solved by assuming that the variations in the tangent vectors g are mostly restricted to an m dimensional subspace. We use PCA to learn

these low-dimensional subspaces for each cluster. Each of the m coefficients representing the low-dimensional projections $\tilde{\mathbf{g}}$ is then modeled with a non-parametric density function using kernel density estimation (with a Gaussian kernel) technique.

Algorithm 2 Learning the observation non-parametric density function for the k-th cluster $f(q_t|x_t = x^k)$ from the clustered shape points $q_1, q_2, \cdots q_z$

1: Compute the intrinsic mean μ^k for the z data points using (2.2)
2: **for** i=1 to z **do**
3: Compute the tangent element $g_i \in T^k_\mu(\mathcal{S})$, such that geodesics from μ^k reaches q_i in unit time
4: **end for**
5: Perform a local PCA on the data points $\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_z$ to learn an m-dimensional subspace
6: **for** i=1 to z **do**
7: Project the data point $\mathbf{g}_i$ into $\tilde{\mathbf{g}}_i$ on the m-dimensional subspace learnt
8: **end for**
9: Learn a non-parametric density function for each of the m coefficients of $\tilde{\mathbf{g}}_i, i = 1 : z$

5.3.2.3 Learning the dynamical model

For each gesture, we want to train a HMM that captures the underlying dynamics of transition between the states (state transition probabilities $P(S_t|S_{t-1})$), and the intermediate (cluster) discrete observation probability $P(x_t|S_t)$. This is a standard problem in HMM and we solve it this using the Baum-Welch method for discrete output HMM [84].

Given N realization of each gesture, where each m-point realization is in the form $Q = q_1, q_2, \cdots, q_m, \quad q_i \in \mathcal{S}$, we assign each of these points to one of the learnt clusters to obtain a sequence of cluster assignments $\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_m, \quad \hat{x} \in \mathcal{X}$. These sequence of clusters are used as discrete observation vectors in the Baum-Welch algorithm to learn the model for each gesture.

The way we assign each shape observation to a cluster can largely affect the performance and accuracy of the model. A simple choice for this assignment is to use a nearest-neighbor rule where the shape point is assigned to its nearest cluster, which is calculated by computing the geodesic distance between this point and all the cluster means, and picking the cluster with the minimum geodesic distance. This can be formulated in the following way

$$\hat{x}_i = \min_{j=1}^{K} d(q_i, \mu^j), \quad for \quad i = 1 : m \tag{5.7}$$

where μ^j is the intrinsic mean for the jth cluster computed on the shape manifold, and $d(.,.)$ is the geodesic distance on $\mathcal{S}$.

Although nearest neighbor assignment performs relatively well, it sometimes produces wrong assignments as it only uses the mean point as a representative of each cluster and ignores the statistical variations within the cluster. These wrong assignments result in inaccurate HMM model parameters.

Therefore, in order to fully utilize the information we learnt about the statistical properties of the data within each cluster we use a ML assignment. In this method we use the cluster observation models learnt in the previous steps $f(q_t|x_t = x^k)$ and assign q_t to

the most likely cluster as follows

$$\hat{x}_i = \max_{j=1}^{K} f(q_i|x_t = x^j), \quad for \quad i = 1 : m \tag{5.8}$$

5.3.2.4 Classification of a test sequence

Given a sequence of observation shapes $Q = q_1, q_2, \cdots, q_m, \quad q_i \in \mathcal{S}$ and a set of M HMM models corresponding to M different gestures, we first compute the conditional observation likelihood $f(q_t|s_t)$ using the learnt cluster density functions $f(q_t|x_t = x^k)$ and the exemplar clusters observation probabilities $p(x_t = x^k|s_t)$. This is calculated by summing over all possible clusters as shown in (5.6).

Given these observations likelihood and the state transition probabilities $p(S_t|S_{t-i})$, computing the likelihood of the test sequence is a simple HMM evaluation problem. We solve it using the standard forward technique and dynamic programming [84]. The test sequence is then classified using the maximum likelihood rule.

5.4 Experimental Results

We carried out an extensive set of experiments to evaluate and verify the effectiveness of using the contour curve shape manifold and the two proposed methods in modeling and recognition of human actions and gestures. The experiments also investigate the effect on performance with changing some of the system choices like the cluster assignment method and whether the clusters are shared among different gesture models or not. These experiments were performed using two different datasets representing the human action and control gesture scenarios respectively. For both datasets, background subtracted im-

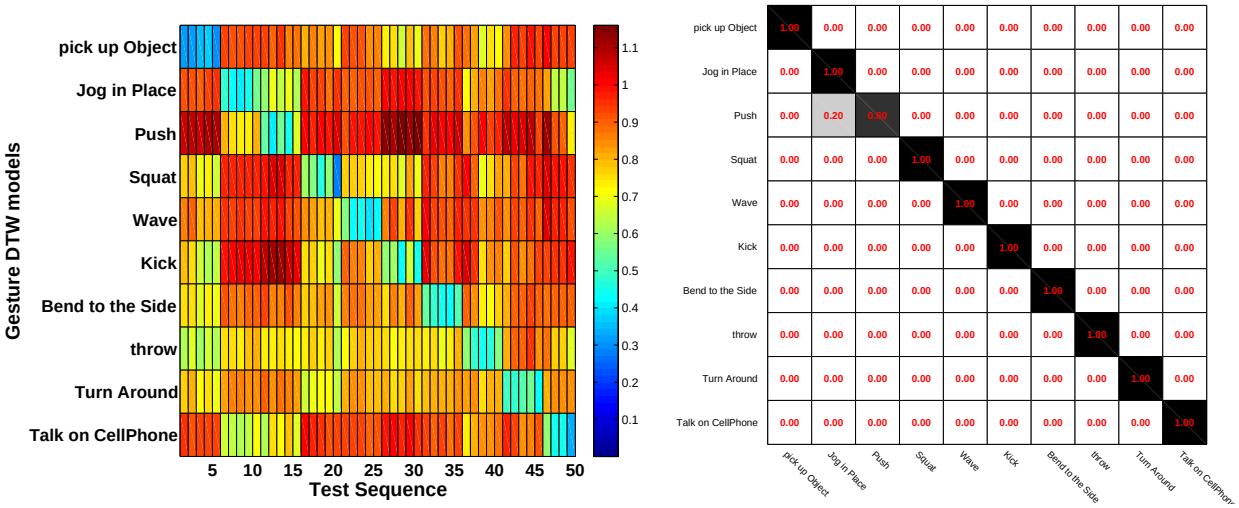
ages with relatively good resolution and quality were available. We used these images to extract the contour of the human as the boundary of the main binary silhouette in the images.

5.4.1 UMD Common Actions Dataset

In the first set of experiments we use the UMD common activity dataset [121] to perform action modeling and recognition. This dataset consists of 10 common activities with 10 different instants of each activity performed by the same actor, which brings the total number of sequence to 100 sequences. Each of these sequences is 80 frames long, which means a total of 8000 frames are in the whole dataset.

5.4.1.1 Template-Based approach

We first use the dataset to evaluate the template based approach presented in Section 5.3.1. We split the dataset into two disjoint sets, of five sequence each. We used the even sequence for learning the template trajectory on the shape manifold while we tested using the remaining five odd sequences per gesture. Figure 5.5-a shows the 10 x 50 similarity matrix between each of the test sequences and each of the learnt action models. Each of these distances corresponds to the warping distance computed on the shape manifold between the test sequence and the model templates. Each row corresponds to a single action model, while each column corresponds to a different test sequence. The strong block diagonal nature of the similarity matrix indicates that the modeling is accurate and is effective for recognition.



(a) (b)

Figure 5.5: Results of using the template based method for classification on the UMD action dataset (a) The 10 x 50 similarity matrix between the 50 test sequences and the 10 action models learnt (better viewed in color) (b) The confusion matrix for action classification

We use this similarity matrix to perform an action recognition experiment, where we assign each test sequence to the model with the smallest warping distance. The resulting confusion matrix for this experiment is shown in Figure 5.5-b, showing an correct recognition rate of 98%.

5.4.1.2 Graphical-Model Based Approach

We used the same dataset to test the graphical model approach presented in Section 5.3.2. In the first stage we use AP to cluster the data points on the manifold using the pairwise geodesic distances between every pair of shapes. We use the preference parameter value, as mentioned in [38], to change the number of clusters to check the effect of this change on our performance. As in the template-based method experiment, we use the data corresponding to the odd sequences for learning the clusters and their statistical

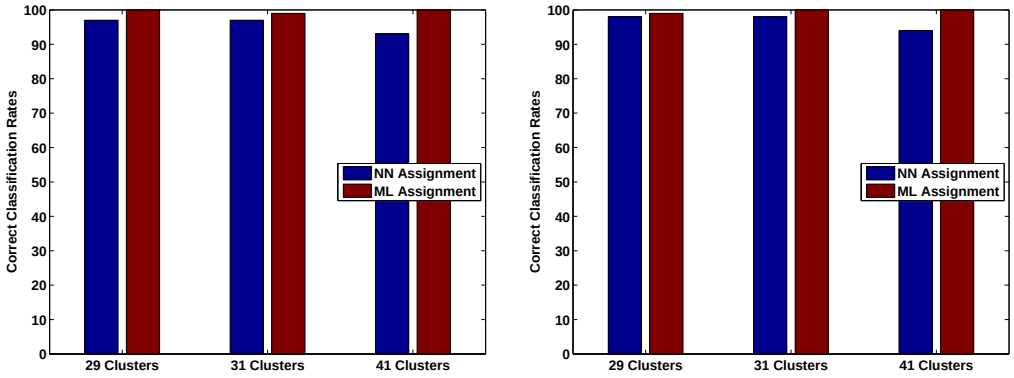
observation density functions.

The clustering results along with statistical models of each cluster are used to learn the dynamics of each action as presented in Section 5.3.2. We first assign the gesture data from the training set into different clusters and use the cluster assignment streams to learn the HMM dynamics of the action. We used two different assignment approaches as described in Section 5.3.2. In the NN assignment method we assign the shape points into the nearest cluster in terms of the geodesic distances with the mean of the cluster, while in the ML assignment we use the learnt statistical model to assign the shape observation into the most likely cluster in terms of the likelihood function.

Figure 5.6 shows the correct classification rates for two validation experiments under different number of clusters and different assignment methods. The first experiment, shown in figure 5.6-a, is similar to the template-based experiment, where we split the data sequences into two disjoint set and use the first set for training and use the second for testing. The second experiment, shown in figure 5.6-b), is using the leave-one-out cross validation method, where we test each sequence using a model learnt using all the other sequences.

The performance results are very close, however we can note a couple of observations:

1. The NN assignment method performs slightly poorly with increasing number of clusters. On the other hand, the ML performance slightly increases with the same number of clusters change. This can be due to the fact that with more clusters the NN assignment can easily assign similar shapes to different clusters, while the ML



(a) (b)

Figure 5.6: UMD common activity dataset action classification results. (a) Classification rates for splitting the data into training and testing sets. (b) Classification rates for leaving-one-out experiment.

method does not make the same mistake.

2. Although both assignment methods are performing well, the ML method is performing slightly better with average recognition rate of 99.67% in both experiments, while NN assignment methods achieved an average recognition rate 96.67% and 95.67% in the first and second experiments respectively. This is expected because the statistical model allows for learning the modes of variation within each cluster, which result in a more accurate modeling of the within cluster variation.

We grouped the recognition rates for all of our experiments in Table 5.1. We also include the results reported in [120] on the same data set. We can note that our HMM approach has a comparable result to the perfect recognition rate reported in [120].

5.4.2 Control Gesture Dataset

The UMD action dataset has been used in several publication on action modeling and recognition. However, the level of variability within the dataset is very limited by

approach	evaluation method	recog. rate %
Template-based approach	split	98
Graphical-based approach with NN assignment	split	96.67
Graphical-based approach with NN assignment	leave one out	95.67
Graphical-based approach with ML assignment	split	99.67
Graphical-based approach with ML assignment	leave one out	99.67
Veeraraghavan et al [120]	leave one out	100

Table 5.1: A summary of the recognition results of our various approaches and evaluation methods on the UMD common actions dataset as compared to the results reported in [120] on the same dataset.

the single actor performing the actions. For this reason, we carried out another set of experiments using a different dataset of human control gestures recently introduced in [68]. The dataset consists of 14 arm control gestures performed by three different actors with different clothes and body frames. Each person repeats a gesture for 3-5 cycles. The length of each cycle varies a lot between 80 and 250 frames for the shortest and longest sequences respectively. Figure 5.7 shows the different gestures in the dataset performed by the first person with an overlay of the extracted contour curves captured every 10 frames.

For this dataset, the authors provided us with a set of background subtracted frames with good resolution. However, the quality of background subtraction was not consistent within all the frames, with the result of added background pixels and poor contour extraction as shown in Figure 5.8. Our method, as in most silhouette-based methods, assumes a relatively clean background subtraction. However, we did not manually fix these errors and use it to test the robustness of our approaches. Another issue was that the different cycles for each person performing a gesture is provided in terms of a long continuous sequence of frames. We had to manually perform a temporal segmentation of the data in

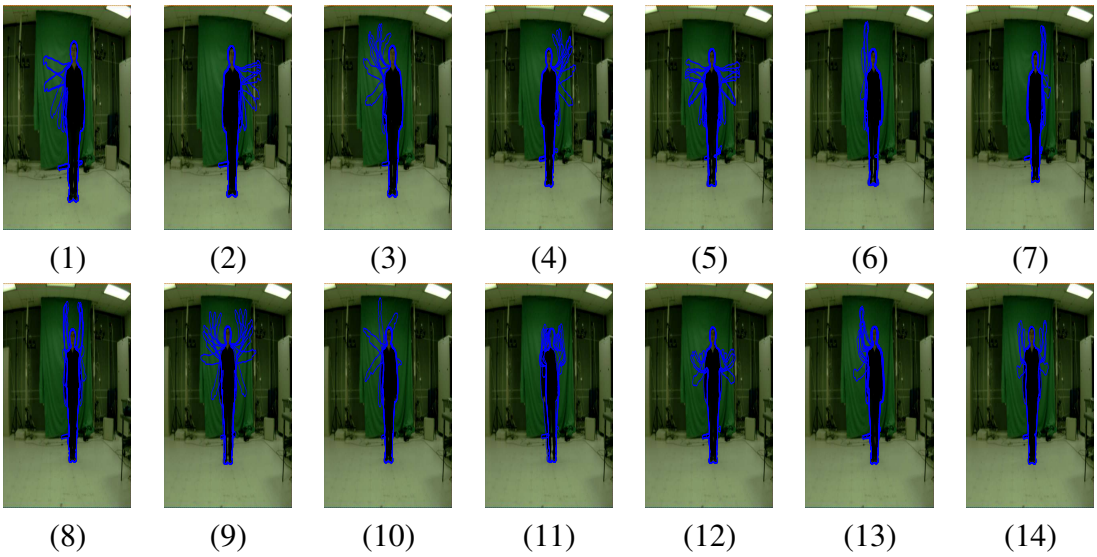


Figure 5.7: The different control gestures in the data set used in our gesture recognition experiment [68]. (1) Turn Right, (2) Turn Left, (3) Attention Right, (4) Attention Left, (5) Flap, (6) Stop Right, (7) Stop Left, (8) Stop Both, (9) Attention Both, (10) Start Engine, (11) Go back, (12) Close Distance, (13) Speed up, and (14) Come Near.

order to obtain a separate sequences for each cycle. The quality of such segmentation was crucial for the DTW template-based approach, where we will show the results before and after fixing the segmentation errors in the next section.

Figure 5.8: Some of the errors in the background subtraction errors in the control gesture dataset resulting in inaccurate silhouette extraction.

5.4.2.1 Template-Based approach

We ran our template-based approach for learning a template for each gesture and used that template for recognition. Figure 5.9 shows the correct classification rates for the different experiments we performed. In the first experiment, we achieved a very poor

recognition rate of 42%, although we have been using all the sequences for training. After carefully inspecting the resulting templates, we found out that poor segmentation of the sequences results in poor performance. This shows the importance of using cleanly segmented sequences for learning templates when the DTW method is used. In experiment 2, we fixed the segmentation issue, which improved the classification rate to around 60%.

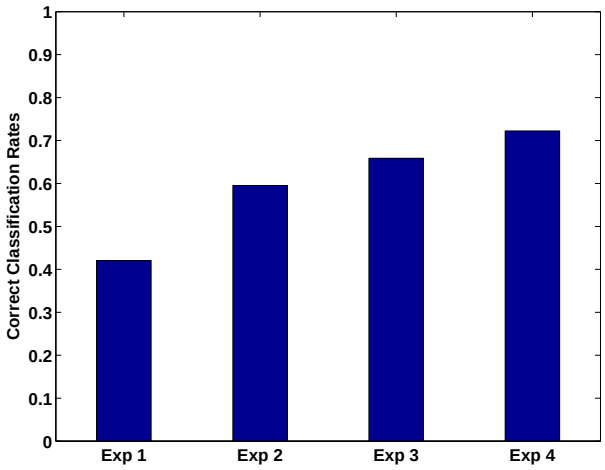


Figure 5.9: The correct classification results for using the template-based approach on the gesture dataset. Experiment 1: Bad segmentation of sequences and random initialization. Experiment 2: Correct segmentation of sequences and random initialization. Experiment 3: Initialization using the best sequence, all sequence used for training. Experiment 4: Initialization using the best Sequence, leave-one-out cross validation

In the last two experiments, we changed the way we initialize the computation of the mean trajectory. Instead of initializing using a random trajectory of the training sequences, we compute the DTW between each training sequence as a model and the rest of the sequences. We initialize the mean trajectory computation using the sequence with the smallest average DTW error to all the other sequences. This change resulted

in increasing the correct classification rate to around 69% while using all the sequences for training and testing as a baseline experiment. In the last Experiment (experiment 4), we perform a leave-one sequence-out cross validation, where we build a model for each test sequence with a training set that exclude that sequence and use all the remaining sequences. Surprisingly, this performed better than the baseline results of using all the data for training, with a correct classification rate of 72%. The similarity matrix and confusion matrix corresponding to the different experiments are shown in figures 5.10 and 5.11 respectively.

We further analyzed the classification results by examining the correct recognition rate for each gesture and each person individually, these results are shown in Figure 5.12-a and 5.12-b respectively. From these figures, we can note the recognition rate varies dramatically across different gestures. For example the Go Back gesture was recognized correctly in only 10% of the cases, while the Flap, Stop Left, and Stop Right gestures achieved perfect recognition rates in most of the experiments. This is due to the fact that some gestures are more distinguishable using only the silhouette information than other gestures. On the other hand, the recognition rates were very close for different persons performing the gestures.

5.4.2.2 Graphical-Model Based Approach

We repeated the same experiments performed on the UMD action dataset for the control gestures dataset but with a slight variation in the setup. Experiment 1 represents a baseline experiment where we use all of the training sequence for learning the model.

This should perform really well and it can point out to any problems within the modeling itself regardless of the generalization of the model to unseen sequences. Experiment 2 represents the same leave-one-sequence-out experiment performed previously for the UMD action dataset. Due to the fact that we have three different persons performing each gesture, we performed a third experiment representing a leave-one-person-out cross validation, where we exclude all the sequences corresponding to one person from training the model used for the test sequences corresponding to that person. This will test the generalization capabilities of our model for new body frames that have not been used in the training phase. This can be a very challenging problem because of the limited number of training sequences remaining after excluding all the sequences for one person.

Figures 5.13 and 5.14 shows the correct classification rates using HMM with five and seven states respectively. The figures shows the performance variations while using different number of clusters and switching between ML and NN cluster assignment methods. Figure 5.15 shows the overall confusion matrix for these experiments combining the results for each experiment and assignment method under different number of clusters and HMM states.

Form these graphs we notice the following:

1. The ML assignment method is performing slightly better than the NN rule with maximum recognition rates of 100%, 94%, and 82% for experiments 1,2, and 3 respectively. compared to maximum correct recognition rates of 99%, 94%, and 76% for the NN assignment methods. Also, the average correct classification rate of ML assignment was 97%, 90%, and 73% for the experiments 1,2, and 3 respectively,

compared to an average rate of 91%, 85%,and 70% for the NN assignment method.

2. A careful study of the confusion matrices in Figure 5.15 shows that certain gestures are very confusing even in the baseline experiment. which lower the average classification rate. For instance, the two gestures of Go back and come Near (gestures 11 and 14 in the tables) are usually confused alot. These two gestures are similar except for the orientation of the palm of the hand and the direction of the motion. This makes them highly indistinguishable using only the silhouette images, even for an human observer.

3. The recognition rates are very similar when we change the number of states, which suggests that the performance does not vary much with the HMM design parameters.

4. The performance is relatively bad in the leave one person out experiment. We believe that this may be mainly due to the low number of training sequences in this case.

Due to the recent introduction of the dataset, we could only compare our recognition rates to the results reported in [68]. Table 5.2 group our results for different approaches and evaluation methods as compared to [68]. We can note that our method managed to achieve comparable performance with 94% maximum recognition rate as compared 95.24% recognition rate reported in [68]. However, we were able to achieve this results using only the silhouette of the background subtracted sequences, while [68] was combining both the shape and motion information obtained using the color images and optical

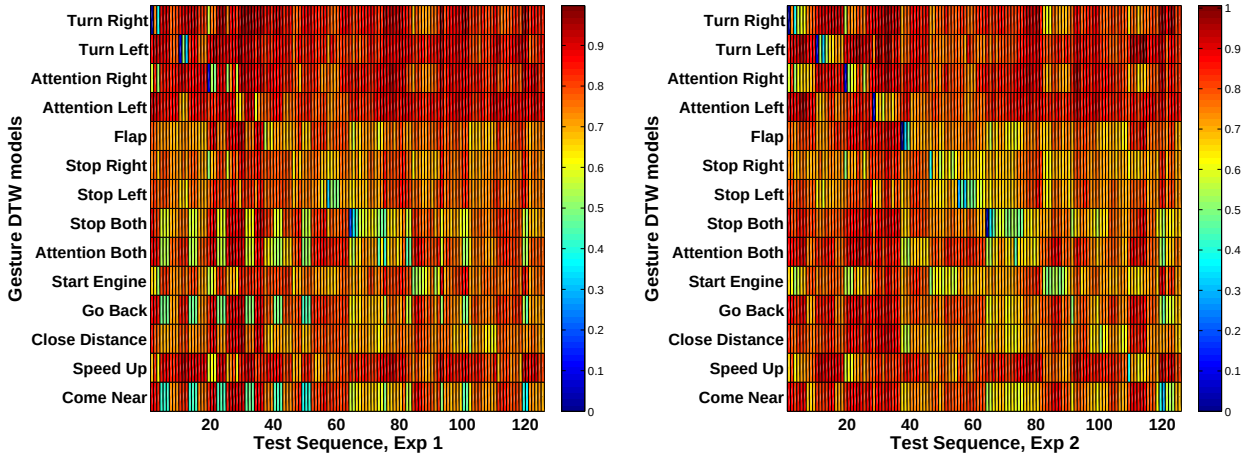
approach	evaluation method	max recog. rate %
Template-based approach	leave one out	72
Graphical-based approach with NN assignment	leave one sequence out	94
Graphical-based approach with NN assignment	leave one person out	76
Graphical-based approach with ML assignment	leave one sequence out	94
Graphical-based approach with ML assignment	leave one person out	82
Lin et al [68], shape only	leave one out	92.86
Lin et al [68], shape and motion	leave one out	95.24

Table 5.2: A summary of the recognition results of our various approaches and evaluation methods on the control gesture dataset as compared to the results reported in [68] on the same dataset.

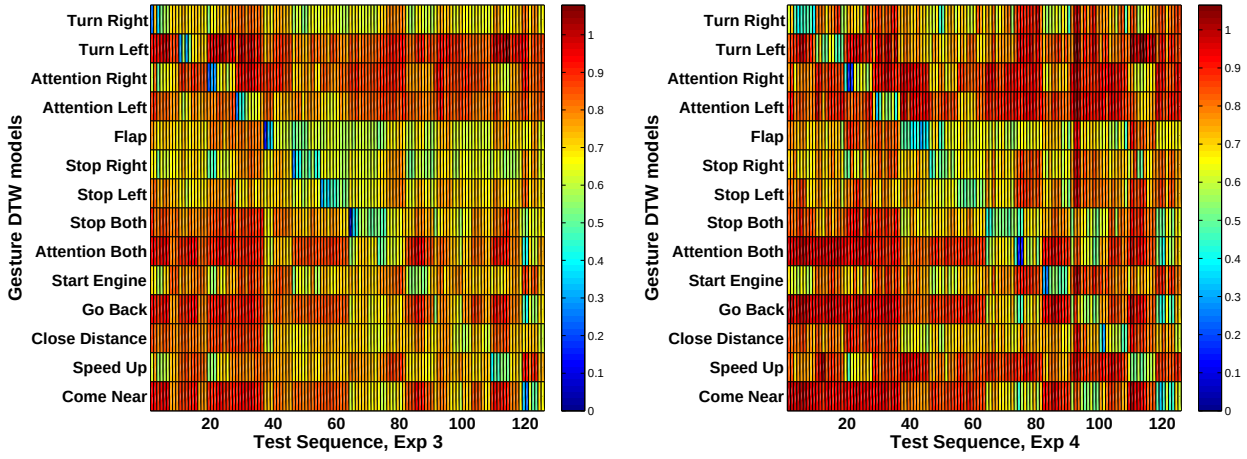
flow computations. In fact, our results exceed the performance reported in the same paper using only the shape cues for recognition.

5.5 Chapter Summary

In this paper, we presented a novel gesture recognition technique using shape manifolds. Contours of the silhouette are extracted and represented as 2D closed elastic curves parameterized using the square-root parametrization. This representation is intrinsically invariant to both translation,scale, and re-parametrization of the curve. Each gesture is modeled as a temporal trajectory on the resulting Riemannian manifold of 2D elastic curves. We proposed template and graphical-based HMM approaches for modeling these trajectories. The two approaches capture the statistical variability of the data, while adhering to the underlying geometry of the manifold. The two approaches were proven successful experimentally using two data sets of human control gesture and actions. In future, we plan to extend the proposed approach to other vision problems that can be modeled using dynamical trajectory on differentiable manifolds.



(a) (b)



(c) (d)

Figure 5.10: The 126 X 14 similarity matrix between all of the test sequences and the 14 gesture models using the template-based approach for experiments 1-4 are shown in a-d respectively (better viewed in color)

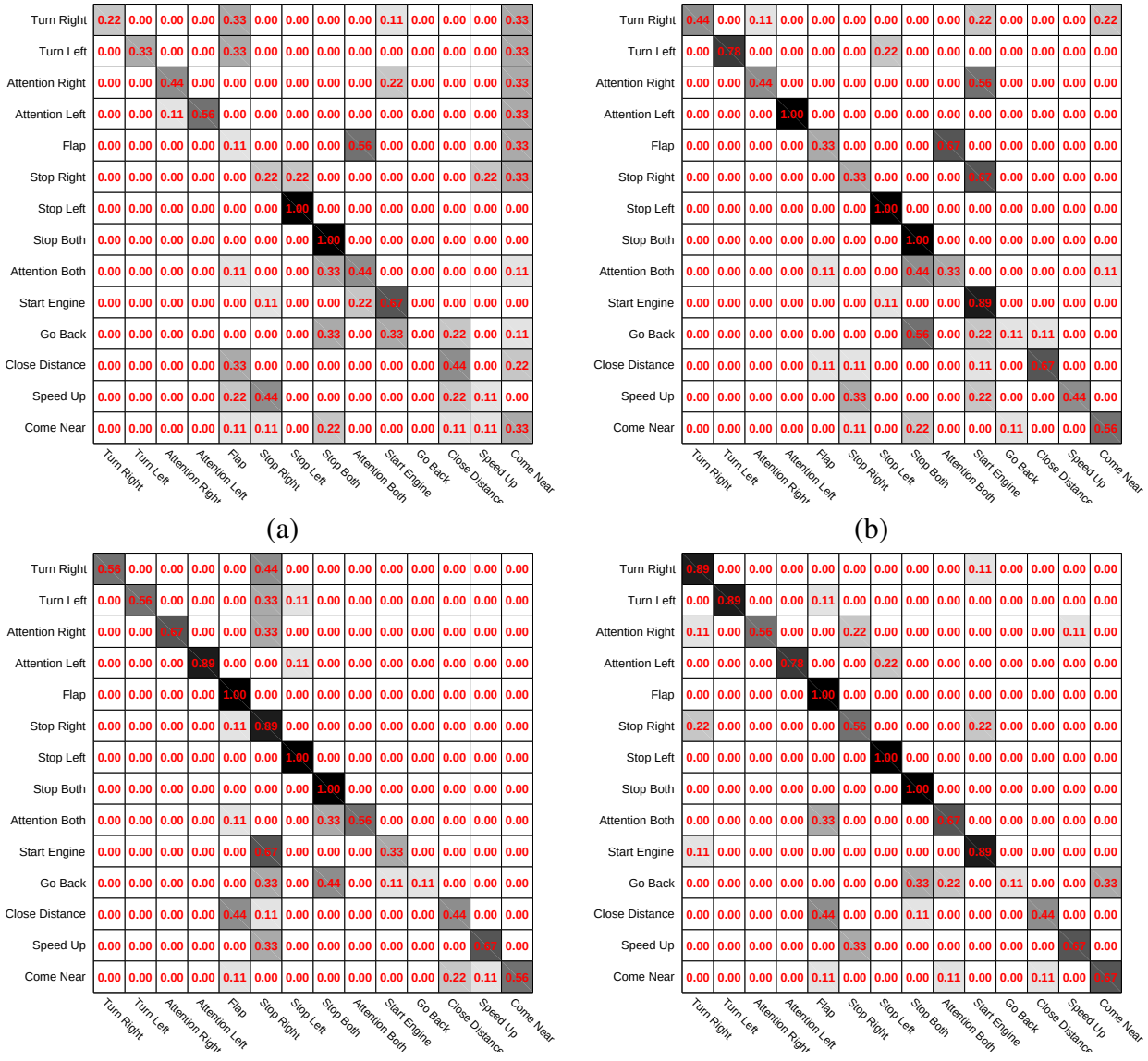
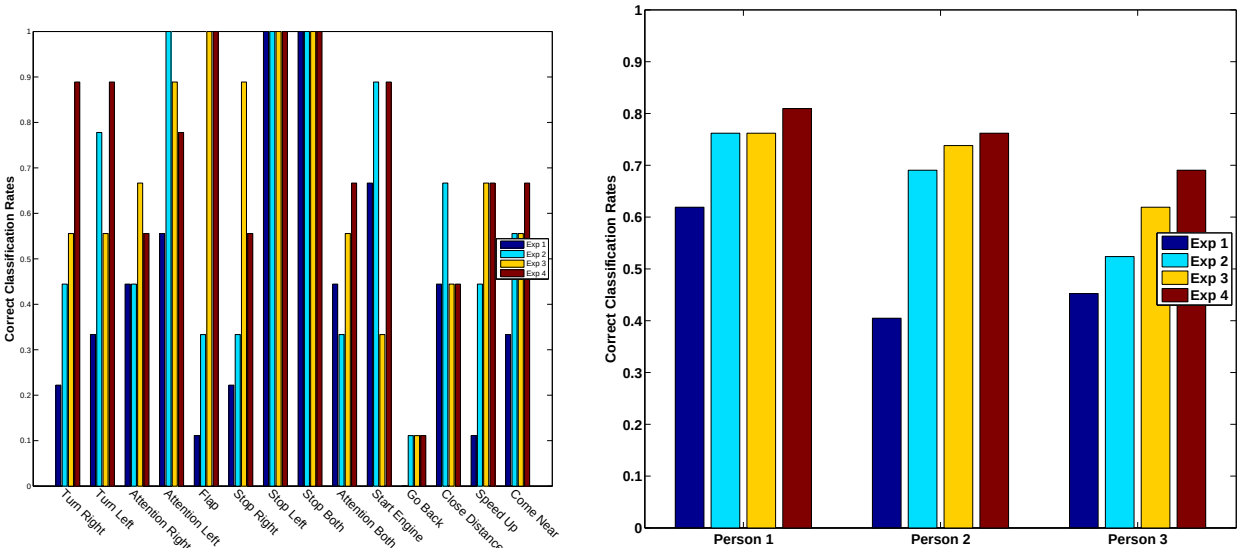
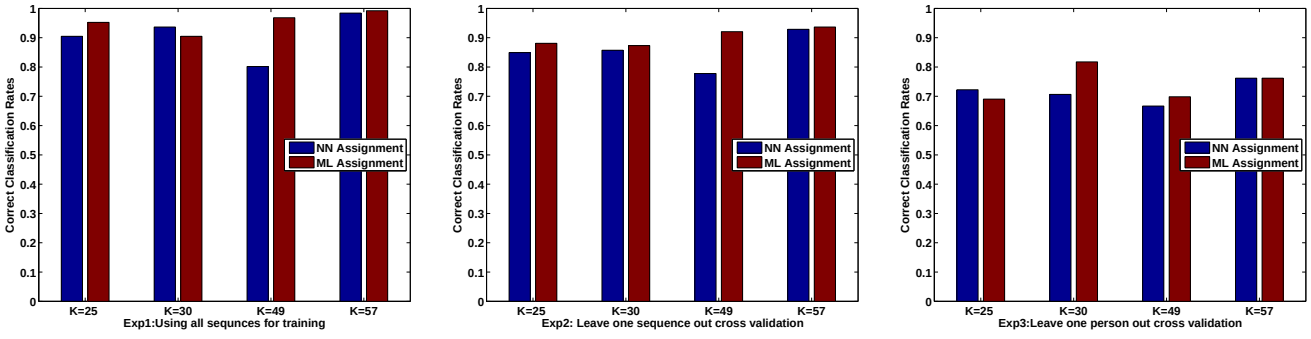


Figure 5.11: The confusion matrix for gesture classification using the template-based method for experiments 1-4 are shown in a-d respectively



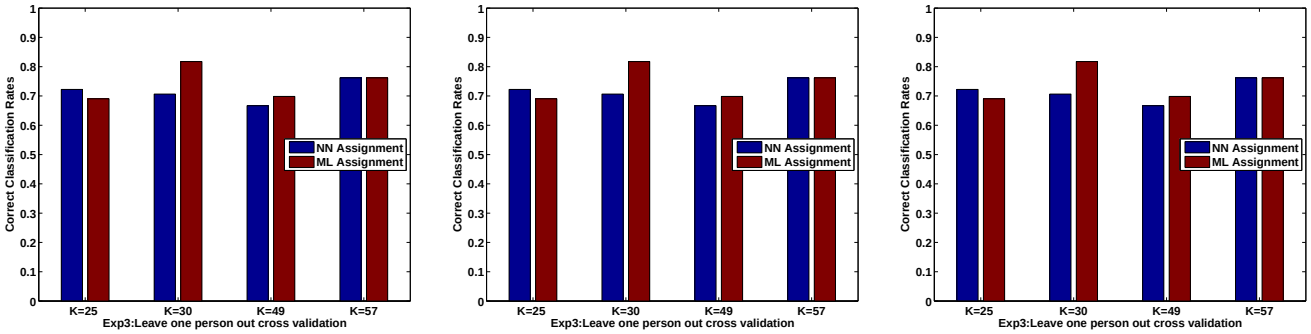
(a) (b)

Figure 5.12: Gesture dataset classification results using the template-based approach (a) The correct classification rates for each gestures in different experiments (b) The correct classification rates for each person in different experiments



(a) (b) (c)

Figure 5.13: The gesture classification results using the graphical model approach, with 5 states HMM. It show the correct classification rates using each of ML and NN assignment and with varying number of clusters (a) Using all the sequences for training (b) Leaving one sequence out cross validation and (c) Leaving one person out cross validation



(a) (b) (c)

Figure 5.14: The gesture classification results using the graphical model approach, with 7 states HMM. It show the correct classification rates using each of ML and NN assignment and with varying number of clusters (a) Using all the sequences for training (b) Leaving one sequence out cross validation and (c) Leaving one person out cross validation

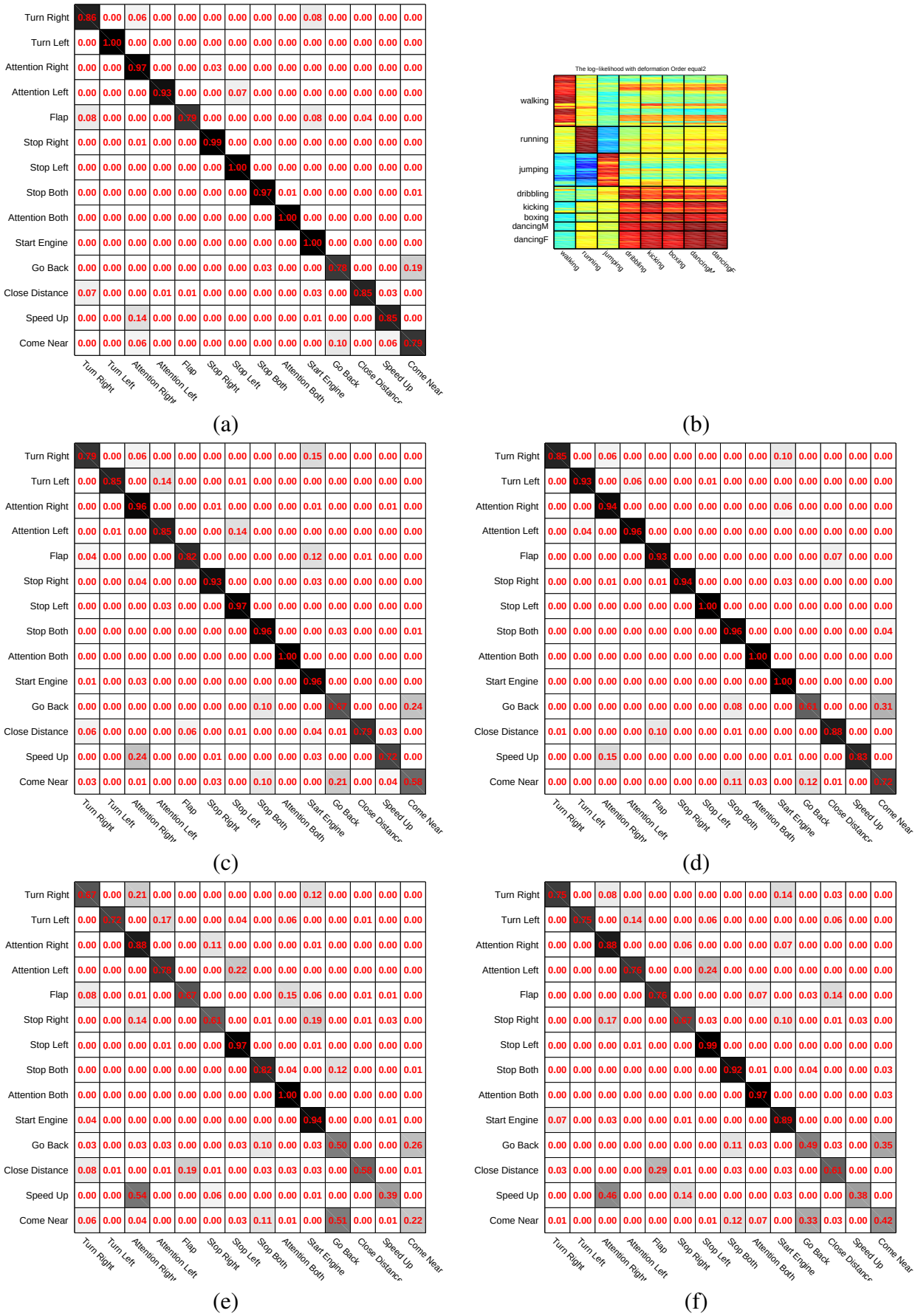


Figure 5.15: The combined confusion matrix for the classification experiment using the graphical based approach with NN assignment (left column) and ML assignment (right column). (a)Experiment 1, NN assignment, (b)Experiment 1, ML assignment, (c)Experiment 2, NN assignment, (d)Experiment 2, ML assignment, (e)Experiment 3, NN assignment, and (f)Experiment 3, ML assignment

Chapter 6

Future Work

There are multiple avenues in theory as well as in application domains for extending the findings of this dissertation. We discuss a few below.

6.1 Learning Temporal Pattern Templates On Riemannian Manifolds

As shown in this dissertation, many computer vision problems can be modeled as a pattern recognition problem of a temporal sequence of points on a Riemannian manifold. Several models have been proposed for solving the analogous problem for data lying on Euclidean spaces. However, very limited attention have been given to this problem using concepts from differential geometry.

One of the approaches we plan to investigate is to learn a temporal template of the training sequences on the Riemannian manifolds. In chapter 5, we proposed an approach to learn temporal templates using DTW and Karcher mean computations. However, this approach does not enforce any regulatory or smoothing constraints on the resulting template. We are interested in developing a more theoretical analysis of this problem using tools from differential geometry. We will use an analysis similar to the work represented in [92] for fitting curves to a finite set of points on a manifold. The problem is formulated as follows.

Given M realization of a temporal pattern, each realization consists of a sequence

of N points $p_{0j}, p_{1j} \cdots, p_{Nj}$ that lie on a Riemannian manifold $\mathcal{M}$. We would like to learn a template sequence γ in $\mathcal{M}$ that best represent this pattern. This template trajectory is fitted to the training data by taking two goals of conflicting nature into consideration:

1. The curve should fit all of the realizations as well as possible, this can be formulated with an energy function

$$E_d(\gamma) = \sum_{i=0}^{N} \sum_{j=1}^{M} d^2(\gamma(t_i), p_{ij}) \tag{6.1}$$

   Where d denotes the geodesic distance function on the Riemannian manifold $\mathcal{M}$

2. The second goal is to have the resulting curve to be as regular as possible, this can be formulated similar to [92] by either minimizing the length of the curve or minimizing the change in velocity along the curve. This goal is measured by another regularity function $E_{s,i}(\gamma), i = 1 \; or \; 2$ defined in the following way

$$E_{s,1}(\gamma) = \frac{1}{2} \int_0^1 \|\dot{\gamma}(t)\|^2 \, dt \tag{6.2}$$

$$E_{s,2}(\gamma) = \frac{1}{2} \int_0^1 \|\ddot{\gamma}(t)\|^2 \, dt \tag{6.3}$$

   The problem is then formulated as minimizing a function of the form

$$\begin{aligned} E: \quad \Gamma \quad &\to \mathbb{R} \\ \gamma \quad &\mapsto E(\gamma) := E_d(\gamma) + \lambda E_s(\gamma) \end{aligned}$$

where Γ is the set $H^1([0,1], \mathcal{M})$ of all continuous paths $\gamma : [0,1] \to \mathcal{M}$ whose weak first derivative is locally square integrable in every chart of $\mathcal{M}$, and λ is a positive, real regularization parameter.

The parameter λ is used to mitigate between the two goals mentioned above. when λ is large, the regularity condition is more emphasized on the expense of the fitting condition. While small λ gives more importance to the fitting constraint.

This minimization is performed using the steepest-descent iteration in the search space Γ. The chosen descent direction is steepest in terms of the Palais metric [65].

$$\langle\langle \upsilon, \omega \rangle\rangle = \langle \upsilon(0), \omega(0) \rangle_{\gamma(0)} + \int_0^1 \left\langle \frac{D\upsilon(t)}{dt}, \frac{D\omega(t)}{dt} \right\rangle_{\gamma(t)} dt \tag{6.4}$$

6.2 Shape-Constrained Non-Rigid Structure form Motion

The factorization algorithm [104] has proven effective in solving the SFM problem for rigid objects. However, extending this algorithm to the non-rigid objects has turned out to be quite tricky. It was shown in [18] that the factorization algorithm can be extended to the non-rigid case if the underlying deformation is modeled as a linear combination of basis shapes. The solution is based on exploiting the orthonormality constraint on the camera rotation matrices. However, in most cases the recovered shape and motion are inaccurate. This inaccuracy is a result of the degeneracy of the problem when solely using the orthonormality constraints as shown by Xiao *et. al.* [129].

In order to remove this ambiguity several approaches have been proposed. In [129], a closed form solution was reached by enforcing a set of heuristic constraints on the basis shapes. Torresani et al. [106] proposed a Gaussian prior for the shape coefficients. Del Bue et al. [26] assumed that the nonrigid shape contains a significant number of points which behave in a rigid fashion. Recently, Bartoli *et al*, [7] proposed that we can itera-

tively find the shape basis based on the assumption that each basis shape should decrease the reprojection error left due to the earlier basis. A novel approach was presented in [82] by replacing the linear subspace approach with a data-driven manifold learnt through a manifold learning approach.

We propose to study the problem of nonrigid structure from motion in a novel way by imposing constraints of the nature of the deformation of the shape. In our formulation, the 3D landmark configuration, corresponding to the shape of the deformable object at each time instant, can be modeled as a point on a Riemannian shape space. We plan to investigate how to improve the accuracy of the NRSFM results by imposing priors on the object shape deformation in this shape space, and imposing regulation constraints on the deformation of the shape between successive frames.

Another interesting direction of research would be to combine the above constraints from the shape spaces with the constraints on the camera rotation matrices under orthographic projection. As these matrices can be modeled as points on the special Euclidean group $SE(3)$ as shown in [95].

6.3 Activity Recognition via Modeling Feature Point Trajectories on the Riemannian manifold of parameterized curves

In this dissertation, we presented several approaches that aim at modeling human actions based on the structure of the shape represented by landmark points. These approaches worked similarly to the NRSFM in what is known as the shape space, where the configuration of the landmarks at each time instant is considered as a shape point. A

dual approach at the problem is to consider the same approach from the trajectory space. Analyzing the different trajectories generated by the different body parts during the span of the action. Recently, the trajectory space was used in solving the NRSFM, where each trajectory is modeled as a linear combination of a set of generic bases [5]. Analysis of the trajectories of hands has also been attempted in [85] for action recognition.

3D trajectories corresponding to different body parts can be represented using the square-root elastic representation of curves as shown in chapter 5. Under this representation, these open curves will form a Riemannian manifold. This will allow us to compute geodesic distances between the different trajectories and build view and re-parametrization invariance into these distance measures.

Bibliography

[1] Cornegie mellon university graphics lab motion capture database. website: *mocap.cs.cmu.edu*.

[2] M. F. Abdelkader, R. Chellappa, Q. Zheng, and A. Chan. Integrated Motion Detection and Tracking for Visual Surveillance. In *Proc. of IEEE International Conference on Computer Vision Systems*, page 28, New York, NY, Jan 2006.

[3] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73:428–440, 1999.

[4] J. K. Aggarwal and Sangho Park. Human motion: Modeling and recognition of actions and interactions. In *Proceedings of the Second Intl. Symp. on 3D Data Processing, Visualization, and Transmission, 3DPVT 2004*, pages 640–647, Washington, DC, USA, Sept. 2004.

[5] Ijaz Akhter, Yaser Ajmal Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. In *Neural Information Processing Systems*, Vancouver, B.C., Canada, December 2008.

[6] D. Ayers and R. Chellappa. Scenario recognition from video using a hierarchy of dynamic belief networks. In *Proc. of Intl. Conf. on Pattern Recognition*, pages 835–838, Washington, DC, USA, September 2000.

[7] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, P. Sayd, and C.F. LASMEA. Coarse-to-fine low-rank structure-from-motion. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, Alaska,, June 2008.

[8] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[9] E. Begelfor and M. Werman. Affine invariance revisited. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages II: 2087–2094, Miami, FL, June 2006.

[10] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 24:509–522, April 2002.

[11] M.A.-A. Bhuiyan, M.E. Islam, N. Begum, M. Hasanuzzaman, Chang Hong Liu, and H. Ueno. Vision based gesture recognition for human-robot symbiosis. *Tenth Intl. Conf. on Computer and information technology, ICCIT 2007*, pages 1–6, Dec. 2007.

[12] M.J. Black and Y. Yacoob. Parameterized modeling and recognition of activities. In *Proc. of IEEE International Conf. on Computer Vision*, pages 120–127, Bombay, Jan 1998.

[13] AF Bobick and JW Davis. The recognition of human movement using temporal templates. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(3):257–267, March 2001.

[14] M. Brand. Morphable 3d models from video. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume 2, page 456, Kauai, Hawaii, December 2001.

[15] M. Brand. A direct method for 3d factorization of nonrigid motion observed in 2d. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 122–128, San Diego, California, June 2005.

[16] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 994 –999, San Juan, Puerto Rico, June 1997.

[17] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 568–574, San Juan, Puerto Rico, June 1997.

[18] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 690–696, Hilton Head, SC, June 2000.

[19] H. Buxton and S. Gong. Visual surveillance in a dynamic and uncertain world. *Artificial Intelligence*, 78(1-2):431–459, 1995.

[20] L. Campbell and A. Bobick. Recognition of Human Body Motion Using Phase Space Constraints. In *Proc. of IEEE International Conf. on Computer Vision*, pages 624–630, Cambridge, Massachusetts, June 1995.

[21] W. Chen and S. F. Chang. Motion trajectory matching of video objects. In *Proceedings of SPIE, Storage and Retrieval for Media Databases*, volume 3972, pages 544–553, 2000.

[22] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active Shape Models: Their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

[23] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[24] Samarjit Das and Namrata Vaswani. Nonstationary shape activities: Dynamic models for landmark shape change and applications. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 32:579 – 592, April 2009.

[25] J. Davis and A. Bobick. The representation and recognition of action using temporal templates. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 928–934, San Juan, Puerto Rico, June 1997.

[26] Alessio Del Bue, Xavier Llad, and Lourdes Agapito. Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *CVPR*, pages 1191–1198, Washington, DC, USA, June 2006. IEEE Computer Society.

[27] M.P. do Carmo. *Riemannian Geometry*. Birkhuser, Englewoods Cliffs, 1992.

[28] C. Dousson, P. Gabarit, and M. Ghallab. Situation recognition: Representation and algorithms. In *Proc. Intl. Joint Conf. on AI*, pages 166–172, Chambery, France, 1993.

[29] I. Dryden and K. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, 1998.

[30] A. Edelman, T.A. Arias, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1999.

[31] AA Efros, AC Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. of IEEE International Conf. on Computer Vision*, pages 726–733, Nice, France, 2003.

[32] A. Elgammal, V. Shet, Y. Yacoob, and LS Davis. Learning dynamics for exemplar-based gesture recognition. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 571 – 578, Madison, WI, June 2003.

[33] A.M. Elgammal and C.S. Lee. Inferring 3D body pose from silhouettes using activity manifold learning. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages II: 681–688, Washington D.C., June 2004.

[34] P.T. Fletcher, Conglin Lu, S.M. Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23(8):995–1005, Aug. 2004.

[35] P.T. Fletcher, S. Venkatasubramanian, and S. Joshi. Robust statistics on riemannian manifolds via the geometric median. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, Alaska, June 2008.

[36] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Trans. Electron. Comput.*, 10:260–268, June 1961.

[37] B.J. Frey and N. Jojic. Flexible models: A powerful alternative to exemplars and explicit models. In *IEEE Computer Society Workshop on Models vs. Exemplars in Computer Vision*, pages 34–41, Honolulu, Hawaii,, Dec. 2001.

[38] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[39] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73:82–98, January 1999.

[40] A. Goh and R. Vidal. Clustering and dimensionality reduction on riemannian manifolds. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 1–7, Anchorage, Alaska,, June 2008.

[41] C.R. Goodall and K.V. Mardia. Projective shape analysis. *Journal of Computational and Graphical Statistics*, pages 143–168, 1999.

[42] U. Grenander. *General Pattern Theory*. Oxford university Press, 1993.

[43] U. Grenander, M. I. Miller, and A. Srivastava. Hilbert-Schmidt lower bounds for estimators on matrix Lie groups for ATR. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 20:790–802, Aug. 1998.

[44] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 22–31, Santa Barbara, CA,, 1998.

[45] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[46] Md. Hasanuzzaman, T. Zhang, V. Ampornaramveth, H. Gotoda, Y. Shirai, and H. Ueno. Adaptive visual gesture recognition for human-robot interaction using a knowledge-based software platform. *Robot. Auton. Syst.*, 55(8):643–657, 2007.

[47] S. Helgason. *Differential geometry, Lie groups, and symmetric spaces*. Academic press, 1978.

[48] S. Hongeng and R. Nevatia. Multi-Agent Event Recognition. In *Proc. of IEEE International Conf. on Computer Vision*, pages II: 84–91, Honolulu, Hawaii,, Dec. 2001.

[49] S. Hongeng and R. Nevatia. Large-scale event detection using semi-hidden Markov models. In *Proc. of IEEE International Conf. on Computer Vision*, pages 1455–1462, Nice, France., Oct. 2003.

[50] M.K. Hu. Visual pattern recognition by moment invariants. *IEEE Trans. on Information Theory*, 8:179–187, February 1962.

[51] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. on Systems, Man and Cybernetics - Part C: Applications and Reviews*, 34(3):334–352, August 2004.

[52] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J. Weber. Automatic symbolic traffic scene analysis using belief networks. In *Proc. AAAI*, pages 966–972, Seattle, WA, July 1994.

[53] S. Intille and A. Bobick. A framework for recognizing multi-agent action from visual evidence. In *Proc. AAAI*, pages 518–525, Orlando, FL, July 1999.

[54] G. Johansson. Visual Perception of Biological Motion and a Model for Its Analysis. *Perception and Psychophysics*, 14(2):201–211, 1973.

[55] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8):609–615, August 1996.

[56] S.H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn. A novel representation for riemannian analysis of elastic curves in Rn. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 1–7, Minneapolis, MN, June 2007.

[57] S.H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn. Removing shape-preserving transformations in square-root elastic (SRE) framework for shape analysis of curves. In *Proc. of intl. conf. on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, pages 387–398, Ezhou, China, August 2007.

[58] A. Kale, A. Sundaresan, A. Rajagopalan, N. Cuntoor, A. RoyChowdhury, V. Kruger, and R. Chellappa. Identification of humans using gait. *IEEE Trans. on Image Processing*, 13(9):1163–1173, July 2004.

[59] D.G. Kendall, D. Barden, T.K. Carne, and H. Le. *Sape and Shape Theory*. John Wiley and Sons, 1999.

[60] D.G. Kendall, D. Barden, T.K. Carne, and H. Le. *Shape and Shape Theory*. John Wiley and Sons, 1999.

[61] E. Klassen and A. Srivastava. Geodesics between 3d closed curves using path-straightening. In *Proc. of European Conference on Computer Vision*, pages I: 95–106, Graz, Austria, May 2006.

[62] Eric Klassen, Anuj Srivastava, and Washington Mio. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 26:372–383, March 2004.

[63] S. Kobayashi and K. Nomizu. *Foundations of differential geometry Volume I*. Wiley-Interscience, 1996.

[64] Y. Kuniyoshi and H. Inoue. Qualitative recognition of ongoing human action sequences. In *Proc. Intl. Joint Conf. on AI*, pages 1600–1609, Chambery, France, 1993.

[65] J. Langer and D.A. Singer. Curve-straightening in Riemannian manifolds. *Annals of Global Analysis and Geometry*, 5(2):133–150, 1987.

[66] H. Le. Locating frechet means with application to shape spaces. *Advances in Applied Probability*, 33(2):324–338, 2001.

[67] D. Liebowitz and A. Zisserman. Metric Rectification for Perspective Images of Planes. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 482–488, Santa Barbara, CA, June 1998.

[68] Z. Lin, Z. Jiang, and L.S. Davis. Recognizing Actions by Shape-Motion Prototype Trees. In *Proc. of IEEE International Conf. on Computer Vision*, pages 444–451, Kyoto, Japan, Oct. 2009.

[69] P. Maurel and G. Sapiro. Dynamic shapes average. In *Proc. Second IEEE Workshop Variational, Geometric and Level Set Methods in Computer Vision*, Nice, France, August 2003.

[70] W. Mio and A. Srivastava. Elastic-String Models for Representation and Analysis of Planar Shapes. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 10–15, Washington, DC, June 2004.

[71] Washington Mio, Anuj Srivastava, and Shantanu Joshi. On shape of plane elastic curves. *International Journal of Computer Vision*, 73:307–324, 2007.

[72] Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81:231–268, 2001.

[73] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104(2-3):90–126, Nov-Dec 2006.

[74] H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Computing*, 6:59–74, May 1988.

[75] B. Neumann and H.J. Novak. Event models for recognition and natural language descriptions of events in real-world image sequences. In *Proc. Intl. Joint Conf. on AI*, pages 724–726, Karlsruhe, Germany, 1983.

[76] J. Owens and A. Hunter. Application of the self-organizing map to trajectory classification. In *Proc. IEEE Intl. Workshop on Visual Surveillance*, pages 77–83, Dublin , Ireland, July 2000.

[77] V. Parmeswaran and R. Chellappa. View invariants for human action recognition. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 613–619, Madison, WI, June 2003.

[78] V. Patrangenaru and K. Mardia. Affine shape analysis and image analysis. *Stochastic Geometry, Biological Structure and Images, Dept. of Statistics, University of Leeds*, pages 56–62, 2003.

[79] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[80] Xavier Pennec and Inria Projet Epidaure. Probabilities and statistics on riemannian manifolds: Basic tools for geometric measurements. In *IEEE Workshop on Nonlinear Signal and Image Processing*, pages 194–198, Antalya, Turkey, June 1999.

[81] P. J. Phillips, S. Sarkar, I. Robledo, P. Grother, and P. J. Bowyer. The gait identification challenge problem: Data sets and baseline algorithm. In *Proc. of Intl. Conf. on Pattern Recognition*, volume 1, page 10385, Quebec, Canada, Aug. 2002.

[82] V. Rabaud and S. Belongie. Re-thinking non-rigid structure from motion. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK, June 2008.

[83] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[84] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of The IEEE*, 77(2):257–286, February 1989.

[85] C. Rao, A. Yilmaz, and M. Shah. View-Invariant Representation and Recognition of Actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[86] P. Remagnini, T. Tan, and K. Baker. Agent-oriented annotation in model based visual surveillance. In *Proc. of IEEE International Conf. on Computer Vision*, pages 857–862, Santa Barbara, CA, June 1998.

[87] Sami Romdhani, Volker Blanz, and Thomas Vetter. Face identification by fitting a 3d morphable model using linear shape and texture error functions. In *Proc. of European Conference on Computer Vision*, pages 3–19, London, UK, 2002. Springer-Verlag.

[88] N. Rota and M. Thonnat. Activity Recognition from video sequence using declarative models. In *Proc. European Conf. on A.I.*, pages 673–680, Berlin, Germany, Aug. 2000.

[89] A. Roy-Chowdhury. A measure of deformability of shapes, with applications in human motion analysis. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 398–404, San Diego, California, June 2005.

[90] A. Roy-Chowdhury and R. Chellappa. Factorization approach for event recognition. In *CVPR Event Mining Workshop*, 2003.

[91] Amit K. Roy-Chowdhury. Towards a measure of deformability of shape sequences. *Pattern Recognition Letters*, 28(15):2164–2172, 2007.

[92] C. Samir, PA Absil, A. Srivastava, and E. Klassen. Fitting curves on Riemannian manifolds using energy minimization. In *Proceedings of the IAPR Conference on Machine Vision Applications*, pages 422–425, Yokohama, Japan, May 2009.

[93] T.B. Sebastian, P.N. Klein, and B.B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 26(5):550–571, May 2004.

[94] G. Shaffer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[95] A. Shaji, S. Chandran, and D. Suter. Manifold optimisation for motion factorisation. In *Proc. of Intl. Conf. on Pattern Recognition*, pages 1–4, Tampa, FL, Aug. 2008.

[96] Yaser Sheikh, Mumtaz Sheikh, and Mubarak Shah. Exploring the space of a human action. In *Proc. of IEEE International Conf. on Computer Vision*, volume 1, pages 144–149, Beijing, China, Oct. 2005.

[97] A. Srivastava, S.H. Joshi, W. Mio, and X. Liu. Statistical Shape Analysis: Clustering, Learning and Testing. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 27(4):590–602, 2005.

[98] A. Srivastava and E. Klassen. Monte Carlo extrinsic estimators for manifold-valued parameters. *IEEE Trans. on Signal Processing*, 50:299–308, February 2001.

[99] A. Srivastava and E. Klassen. Bayesian and Geometric Subspace Tracking. *Advances in Applied Probability*, 36:43–56, March 2004.

[100] Anuj Srivastava, Shantanu H. Joshi, Washington Mio, and Xiuwen Liu. Statistical shape analysis: Clustering, learning, and testing. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 27:590–602, Apr. 2005.

[101] Anuj Srivastava, Eric Klassen, Shantanu H. Joshi, and Ian H. Jermyn. Shape analysis of elastic curves in euclidean spaces. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, in review, 2010.

[102] T. Starner and A. Pentland. Visual recognition of american sign language using hidden Markov models. In *Proc. Intl. Workshop on Face and Gesture Recognition*, pages 189–194, Killington, VT, 1995.

[103] Rawesak Tanawongsuwan and Aaron F. Bobick. Modelling the effects of walking speed on appearance-based gait recognition. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 783–790, Washington, DC., June 2004.

[104] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9:173–154, 1992.

[105] L. Torresani and C. Bregler. Space-Time Tracking. In *Proc. of European Conference on Computer Vision*, pages 801 – 812, Copenhagen, Denmark, May 2002.

[106] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 30(5):878–892, 2008.

[107] L. Torresani, D.B. Yang, E.J. Alexander, and C. Bregler. Tracking and Modeling Non-Rigid Objects with Rank Constraints. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages I:493–500, Honolulu, Hawaii, Dec. 2001.

[108] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *International Journal of Computer Vision*, 48(1):9–19, 2002.

[109] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, Aug. 1987.

[110] S. Tsuji, A. Morizono, and S. Kuroda. Understanding a simple cartoon film by a computer vision system. In *Proc. Intl. Joint Conf. on AI*, pages 609–610, Cambridge, MA, 1977.

[111] P. Turaga and R. Chellappa. Locally time-invariant models of human activities using trajectories on the grassmannian. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 2435–2441, Miami, Florida, USA, June 2009.

[112] P.K. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, Alaska, June 2008.

[113] O. Tuzel, F.M. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Proc. of European Conference on Computer Vision*, pages II: 589–600, Graz, Austria, May 2006.

[114] O. Tuzel, F.M. Porikli, and P. Meer. Learning on lie groups for invariant detection and tracking. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK, June 2008.

[115] O. Tuzel, F.M. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 30:1713–1727, October 2008.

[116] N. Vaswani, A. Roy-Chowdhury, and R. Chellappa. Activity recognition using the dynamics of the configuration of interacting objects. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume 2, pages II – 633– 640, Madison, WI, June 2003.

[117] N. Vaswani, A. Roy-Chowdhury, and R. Chellappa. "Shape Activity": A Continuous State HMM for Moving/Deforming Shapes with Application to Abnormal Activity Detection . *IEEE Trans. on Image Processing*, 14(10):1603–1616, 2005.

[118] A. Veeraraghavan, A. Roy-Chowdhury, and R. Chellappa. Role of shape and kinematics in human movement analysis. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 730 – 737, Washington, DC, June 2004.

[119] A. Veeraraghavan, A. Roy-Chowdhury, and R. Chellappa. Matching shape sequences in video with an application to human movement analysis. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 27:1896–1909, December 2005.

[120] A. Veeraraghavan, A. Srivastava, A.K. Roy-Chowdhury, and R. Chellappa. Rate-invariant recognition of humans and their activities. *IEEE Trans. on Image Processing*, 18:1326 –1339, June 2009.

[121] Ashok Veeraraghavan, Rama Chellappa, and Amit K. Roy-Chowdhury. The function space of an activity. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 959 – 968, Washington, DC, USA, June 2006.

[122] V.T. Vu, F. Bremond, and M. Thonnat. Automatic Video Interpretation: A Recognition Algorithm for Temporal Scenarios Based on Pre-compiled Scenario Models. In *Proc. of IEEE International Conference on Computer Vision Systems*, pages 523 – 533, Graz, Austria, April 2003.

[123] L. Wang and D. Suter. Learning and matching of dynamic shape manifolds for human action recognition. *IEEE Trans. on Image Processing*, 16(6):1646–1661, 2007.

[124] L. Wang and D. Suter. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MN, June 2007.

[125] S.B. Wang, A. Quattoni, L.P. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 1521 – 1527, New York, NY, June 2006.

[126] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 1–7, Anchorage, Alaska, June 2008.

[127] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *Proc. of IEEE International Conf. on Computer Vision*, pages 1 –7, Rio de Janeiro, Brazil,, Oct. 2007.

[128] A. Wilson and A Bobick. Recognition and interpretation of parametric gesture. In *Proc. of IEEE International Conf. on Computer Vision*, pages 329–336, Bombay, India, Jan. 1998.

[129] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *Proc. of European Conference on Computer Vision*, volume 4, pages 573–587, Prague, Czech Republic, May 2004.

[130] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden Markov model. In *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 379–385, Urbana-Champaign, IL, June 1992.

[131] L. Younes. Computable elastic distance between shapes. *SIAM Journal of Applied Mathematics*, 58(2):565–586, 1998.

[132] L. Younes, P. W. Michor, J. Shah, D. Mumford, and R. Lincei. A metric on shape space with explicit geodesics. *Matematica E Applicazioni*, 19(1):25–57, 2008.

[133] C.T. Zahn and R.Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, 21:269–281, March 1972.

[134] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.

[135] Wenyi Zhao and Rama Chellappa. *Face Processing: Advanced Modeling and Methods*. Academic Press, Inc., Orlando, FL, USA, 2005.