

# Silhouette-based Gesture and Action Recognition via Modeling Trajectories on Riemannian shape manifolds<sup>☆</sup>

Mohamed F. Abdelkader<sup>a,\*</sup>, Wael Abd-Almageed<sup>a</sup>, Anuj Srivastava<sup>b</sup>, Rama Chellappa<sup>a</sup>

<sup>a</sup>*Department of Electrical and Computer Engineering and Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 USA.*

<sup>b</sup>*Department of Statistics, Florida State University, 106D OSB, FSU, Tallahassee, FL 32306 USA.*

---

## Abstract

This paper addresses the problem of recognizing human gestures from videos using models that are built from the Riemannian geometry of shape spaces. 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 propose two approaches for modeling these trajectories. In the first template-based approach, we use dynamic time warping (DTW) to align the different trajectories using elastic geodesic distances on the shape space. The gesture templates are then calculated by averaging the aligned trajectories. In the second approach, we use a graphical model approach similar to an exemplar-based hidden Markov model, where we cluster the gesture shapes on the shape space, and build non-parametric statistical models to capture the variations within each cluster. We model each gesture as a Markov model of transitions between these clusters. To evaluate the proposed approaches, an extensive set of experiments was performed using two different data sets representing gesture and action recognition applications. The proposed approaches not only are successfully able to represent the shape and dynamics of the different classes for recognition, but are also robust against some errors resulting from segmentation and background subtraction.

---

## 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 [1], robotics [2], security, and multimedia analysis.

Existing literature in human movement visual analysis (see reviews [3, 4, 5, 6]) uses the terms gesture and action interchangeably to refer to sequences of human poses corresponding to different activities. However, there are slight

---

<sup>☆</sup>This work was partially supported by ONR grant number N00014-09-1-0664.

\*Corresponding author

*Email addresses:* mdfarouk@umd.edu (Mohamed F. Abdelkader), wamageed@umiacs.umd.edu (Wael Abd-Almageed), anuj@stat.fsu.edu (Anuj Srivastava), rama@cfar.umd.edu (Rama Chellappa)

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 recognizing 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 paper can be used for either of the two problems. This is demonstrated by using two different experimental datasets for gesture and action recognition respectively. We will use the term gesture to refer to both gestures and simple actions throughout the paper.

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 [7], optical flow [8] and feature points trajectories [9, 10] 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.

In order to overcome this limitation, silhouette-based approaches have been receiving increasing attention recently [11, 12, 13]. 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 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 [14], Fourier descriptors [15], shape moments [16], and shape context [17]. 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 nonlinear manifolds 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 training data using dimensionality reduction techniques combined with a suitable notion of local discriminative measure between the visual data features. This technique was recently used [18, 19, 13] for human action recognition

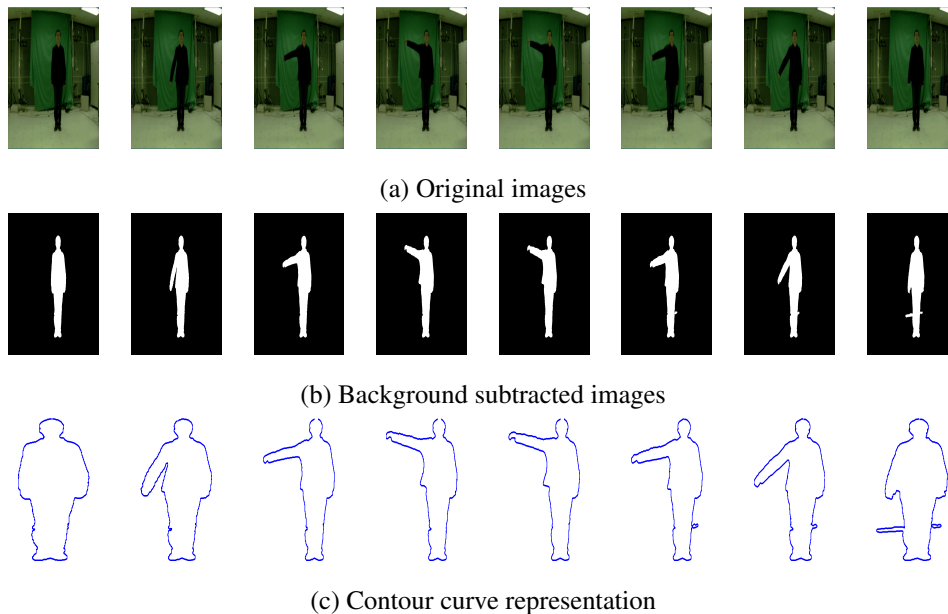


Figure 1: An example of different frames of a humans performing a "Turn Right" control gesture. (a) The original video frames (b) The background subtracted images and (c) The sequence of contours. Most of the information about the gesture can be modeled using only the contour curves

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 differential geometry. 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 [20], affine invariant shape clustering [21], and activity modeling [22]. 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 [23], parameter estimation [24], clustering and dimensionality reduction [25], classification [26], and statistical analysis [27, 22].

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 [28]. Various models were used for modeling these dynamics, ranging between statistical generative models [29, 30, 31, 32], and the more recent discriminative models [33, 34]. The invariance to temporal rate of execution of action in such models is crucial for achieving accurate recognition [35].

### 1.1. Motivation and Overview of Approach

In this paper, 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 [36, 37, 38], that uses differential geometric tools on the shape spaces of closed curves. Similar ideas have also been presented in [39, 40, 41]. 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  $L^2$  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 [42].

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 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 [43], 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.

### 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 variation 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.

### 1.3. Organization

The remainder of this paper is organized as follows: Section 2 reviews some of the work in different areas related to this paper. In Section 3, we review some notations from Riemannian geometry and then 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 4 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.

## 2. Related Work

The problems of gesture and action recognition have received great attention in the literature. Several survey papers [3, 4, 5, 6] have tried to group and analyze the existing body of work. The reader is referred to these review papers for a complete overview of the related work in the field. Meanwhile, we will give a brief review of some of the approaches in three different areas that are of most relevance to this paper.

### 2.1. Gesture and Action Modeling

Wang and Suter recently proposed two approaches [34, 13] for modeling motion subspaces associated with human actions represented by silhouettes. In the first approach [34], 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 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 [13], they use features based on distance transform (DT) and use local preserving projections (LPP) to project the feature data onto the 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 [44, 45], 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 [46], 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 [47, 48] were used in [44] to model body gestures using Chamfer distance as a matching measure between different poses. The same tool was used in [45] for view invariant modeling of actions using 3D exemplars. Our technique for learning the graphical model for trajectories presented in Section 4, 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.

In robotics and human-robot interaction domains, several other aspects of the gesture recognition problem are more demanding. In [49], problems like gesture spotting and interpretation from a continuous sequence of unsegmented video sequence were handled using HMMs. A special transition gesture was modeled to account for transition among different learnt gesture models.

## 2.2. Shape Spaces in Computer Vision

Among the different geometric manifolds that are of interest in computer vision applications, shape spaces have received the most attention in the literature. The goal of these spaces is building tools 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 presented by a discrete sampling of the object contour [50, 51]. This formulation was used to represent shape spaces that are invariant under transformation groups such as similarity [52], affine [53], and projective transformation [54]. 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) [55], that assumes that shape belongs to a Euclidean space and uses principal component analysis (PCA) of the landmark location to model the shape variability 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 [56], 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 the high 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 [57], as well as the

motion of individuals [28]. 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 dynamics of the projected points are then modeled using regular vector space approaches on this single tangent plane. In a more recent work [58], 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 were no experimental results for using this approach for action classification and recognition.

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 [59], which uses differential geometry tools for analyzing the space of closed curves using elastic string models. Similar ideas have also been presented in [41]. Using Riemannian analysis of such space, statistical models and efficient methods were built to perform tasks such as shape modeling and clustering [42]. Later, the same methods were combined with a square-root representation [36, 38] 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 [37].

### 3. Manifold Representation of Silhouettes

As mentioned earlier, we will use the square-root elastic representation [36, 37] 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.

For the sake of completeness, we will first review some of the general notations in Riemannian geometry. We will then briefly introduce the reader to the square-root elastic representation of curves and some properties of the the space of curves under such representations.

#### 3.1. Review of Riemannian Geometry

We will briefly review the basics of Riemannian geometry [60]. 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 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 an (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  $\langle \cdot, \cdot \rangle_x$  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  $\|v\| = \langle v, v \rangle_x^{(1/2)}$ . 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.

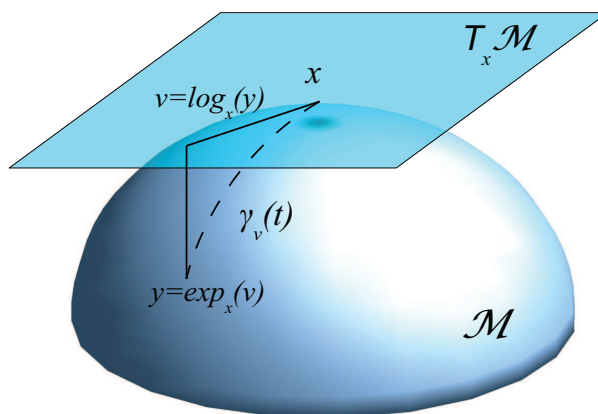


Figure 2: A two-dimensional manifold  $\mathcal{M}$  in  $\mathbb{R}^3$ ,  $T_x\mathcal{M}$  the tangent plane at  $x \in \mathcal{M}$ , and the exp and log maps relating  $x, y \in \mathcal{M}$ .

Given a tangent vector  $v \in T_x\mathcal{M}$ , there exists a locally unique geodesic,  $\gamma_v(t)$ , starting at  $x$  with  $v$  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  $v$  to a point on the manifold that is reached in unit time by the geodesic  $\gamma_v(t)$ . 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  $log_x$ , the geodesic distance between  $x$  and  $y$  is given by

$$d(x, y) = \|log_x(y)\|. \quad (1)$$

As explained in [61, 62], 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  $x_1, \dots, x_n \in \mathcal{M}$  that lie on a sufficiently small neighborhood on  $\mathcal{M}$ .

The intrinsic mean or Karcher mean  $\mu$  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

$$\mu = \operatorname{argmin}_{x \in \mathcal{M}} \sum_{i=1}^N d(x, x_i)^2 \quad (2)$$

where  $d(\cdot, \cdot)$  is the Riemannian geodesic distance defined in (1). A gradient descent approach was proposed in [61, 62] 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  $\mu$ , it was shown in [63] 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).

### 3.2. Representation and Shape Space of Closed Curves

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

$$q(t) = \frac{\dot{\beta}(t)}{\sqrt{\|\dot{\beta}(t)\|}}. \quad (3)$$

where  $\|\cdot\|$  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  $\beta$  can be recovered within a translation, using  $\beta(t) = \int_0^t \|q(s)\| ds$ . Let  $\mathcal{C}$  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  $\mathcal{C}$  is a submanifold of the unit sphere.  $\mathcal{C}$  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 [64], 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  $\mathcal{D} = \{\gamma : \mathbb{S}^1 \rightarrow \mathbb{S}^1\}$  is the space of all orientation-preserving diffeomorphisms. The resulting (transformation invariant) shape space  $\mathcal{S}$  is defined as a quotient space as follows

$$\mathcal{S} = \mathcal{C}/(SO(2) \times \mathcal{D}). \quad (4)$$

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  $C$  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:

$$d([q_0], [q_1]) = \min_{O \in SO(2), \gamma \in \mathcal{D}} d(q_0, O(q \circ \gamma) \sqrt{\dot{\gamma}}). \quad (5)$$

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 [37, 39] 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 [39]. It uses the Karcher mean formulation as presented in (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 [42]. In that approach, all the sample points are projected to the tangent plane  $T_\mu(\mathcal{S}) \in \mathbb{L}^2$ , where  $\mu$  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 [42] for more details and experimental analysis.

#### 4. 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

$$Q = q_1, q_2, \dots, q_m, \quad (6)$$

where  $q_i \in \mathcal{S}$  for  $i = 1, 2, \dots, m$ , and  $\mathcal{S}$  is the quotient shape space described in section 3. 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 [65] 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 [43]. 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 [29], 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  $Q_1, Q_2, \dots, Q_N$  of a gesture, where each realization is a time series of shape points in the form shown in (6), 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?

#### 4.1. 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 algorithm [65], 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 [66], and rate-invariant activity recognition [35]. 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) as a distance function between the shape features at different time instants.

#### 4.1.1. Iteratively learning the template trajectory on the manifold

Given the  $N$  labeled realization  $Q_1, Q_2, \dots, Q_N$  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.

$$Q^* = AVG_{\mathcal{S}}(f_1(Q_1), f_2(Q_2), \dots, f_N(Q_N)). \quad (7)$$

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 (7), 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 3 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.

---

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

---

- 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
- 

#### 4.1.2. 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 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.

#### 4.2. 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 [29]. 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 3. 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 [67]. This ensures that the state transition and observation density functions are optimized in the maximum likelihood sense.

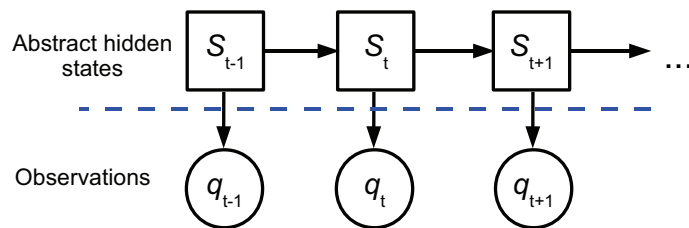


Figure 3: A graphical model of unfolding a standard continuous state HMM

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 4. The resulting model in this case is similar to the decoupled exemplar-based HMM presented in [44]. 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.

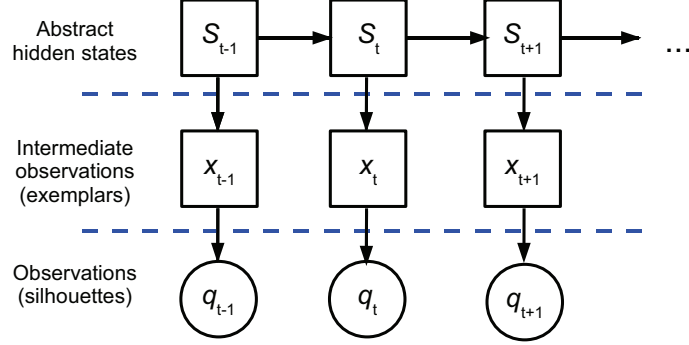


Figure 4: A graphical model of unfolding the exemplar-based HMM used in modeling the gesture dynamics

In our approach we model the dynamics of each gesture as a Markov model of  $L$  hidden abstract states  $H^{(1)}, H^{(2)}, \dots, H^{(L)}$ . 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  $\mathcal{X} = \{x^k | k = 1, \dots, K\}$ , 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 be calculated as

$$f(q_t | s_t) = \sum_{k=1}^K f(q_t | x_t = x^k) P(x_t = x^k | s_t) \quad (8)$$

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.

#### 4.2.1. 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 [42]. 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 [43]. It finds a set of 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 these shapes on the shape space  $\mathcal{S}$ . The output of the algorithm is a set of  $K$  exemplar shapes  $q^1, q^2, \dots, q^K$ , 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. 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.

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

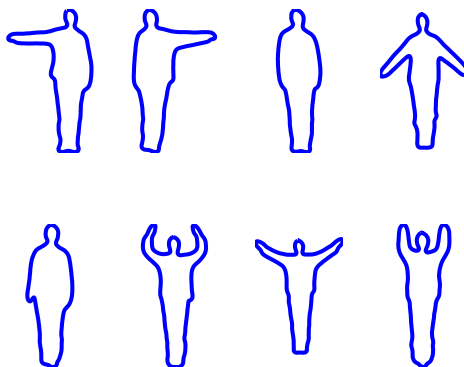


Figure 5: Some exemplar shapes representing clusters computed on the shape manifold using AP.

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.

#### 4.2.2. 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 [42] 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  $\mu^k$  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  $\mu^k$ . Therefore, for every shape  $q \in \mathcal{S}$  in the  $k^{\text{th}}$  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  $\mu_k$  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.

#### 4.2.3. 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

---

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

---

- 1: Compute the intrinsic mean  $\mu^k$  for the  $z$  data points using (2)
  - 2: **for**  $i=1$  to  $z$  **do**
  - 3:   Compute the tangent element  $g_i \in T_{\mu^k}(\mathcal{S})$ , such that geodesics from  $\mu^k$  reaches  $q_i$  in unit time
  - 4: **end for**
  - 5: Perform a local PCA on the data points  $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_z$  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  $\tilde{\mathbf{g}}_i, i = 1 : z$
- 

is a standard problem in HMM and we solve it this using the Baum-Welch method for discrete output HMM [29].

Given  $N$  realization of each gesture, where each  $m$ -point realization is in the form  $Q = q_1, q_2, \dots, q_m, \quad q_i \in \mathcal{S}$ , we assign each of these points to one of the learnt clusters to obtain a sequence of cluster assignments  $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m, \quad \hat{x} \in \mathcal{X}$ . 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

$$\hat{x}_i = \min_{j=1}^K d(q_i, \mu^j), \quad \text{for } i = 1 : m \quad (9)$$

where  $\mu^j$  is the intrinsic mean for the  $j$ th 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 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

$$\hat{x}_i = \max_{j=1}^K f(q_i|x_t = x^j), \quad \text{for } i = 1 : m \quad (10)$$

#### 4.2.4. Classification of a test sequence

Given a sequence of observation shapes  $Q = q_1, q_2, \dots, q_m$ ,  $q_i \in \mathcal{S}$  and a set of  $M$  HMM models corresponding to  $M$  different gestures, we first compute the conditional observation likelihood  $f(q_i|s_t)$  using the learnt cluster density functions  $f(q_i|x_t = x^k)$  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 (8).

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 [29]. The test sequence is then classified using the maximum likelihood rule.

## 5. 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. As in all silhouette based methods, we assume the availability of a relatively clean background subtracted images representing the silhouettes. In order to avoid dealing with low-level processing steps such as background subtraction and object segmentation, the existence of such images was an important criteria in choosing the datasets used in these experiments. For both datasets, background subtracted images 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.

### 5.1. UMD Common Actions Dataset

In the first set of experiments we use the UMD common activity dataset [68] 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.

#### 5.1.1. Template-Based approach

We first use the dataset to evaluate the template based approach presented in Section 4.1. 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 6-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

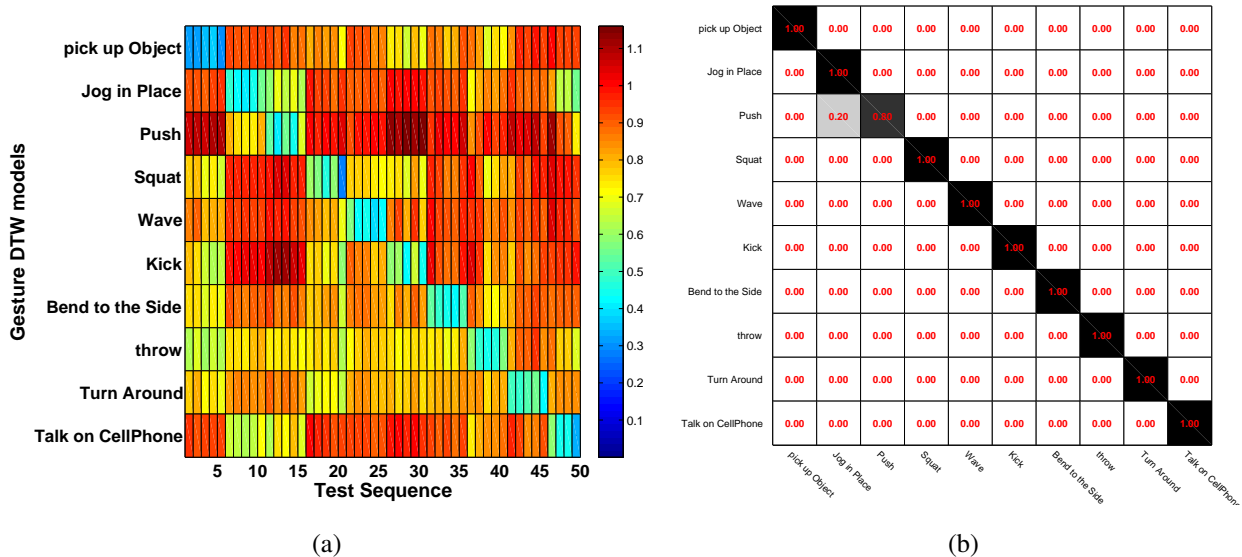


Figure 6: Results of using the template based method for classification on the UMD action dataset (a) The 10 x 50 similarity matrix between the 50 test sequences and the 10 action models learnt (better viewed in color) (b) The confusion matrix for action classification

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.

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 6-b, showing an correct recognition rate of 98%.

### 5.1.2. Graphical-Model Based Approach

We used the same dataset to test the graphical model approach presented in Section 4.2. 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 [43], 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 4.2. 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 4.2. 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 7 shows the correct classification rates for two validation experiments under different number of clusters and different assignment methods. The first experiment, shown in Figure 7-a, is similar to the template-based experiment,

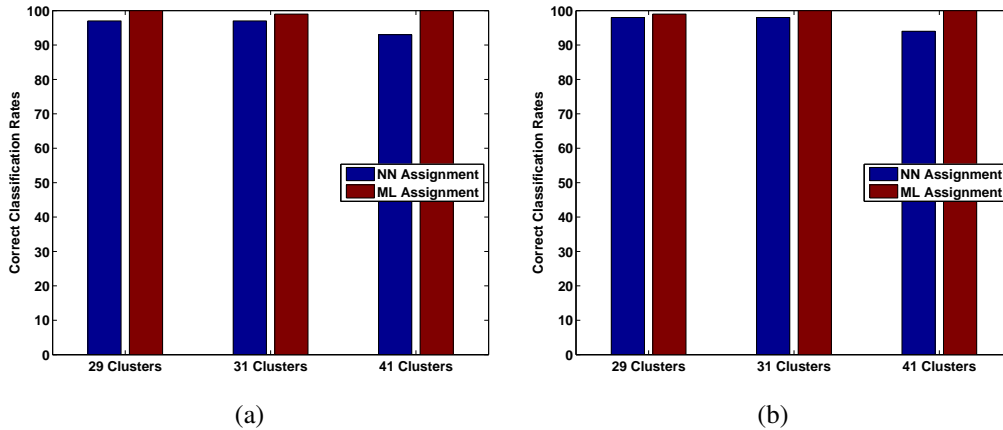


Figure 7: UMD common activity dataset action classification results. (a) Classification rates for splitting the data into training and testing sets. (b) Classification rates for leaving-one-out experiment.

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 7-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 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 1. We also include the results reported in [35] on the same data set. We can note that our HMM approach has a comparable result to the perfect recognition rate reported in [35].

## 5.2. 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 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 [46]. The dataset consists of 14 arm control gestures performed by three different actors with different clothes and body

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 [35]	leave one out	100

Table 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 [35] on the same dataset.

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 8 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 9. 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 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.

### 5.2.1. Template-Based approach

We ran our template-based approach for learning a template for each gesture and used that template for recognition. Figure 10 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%.

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

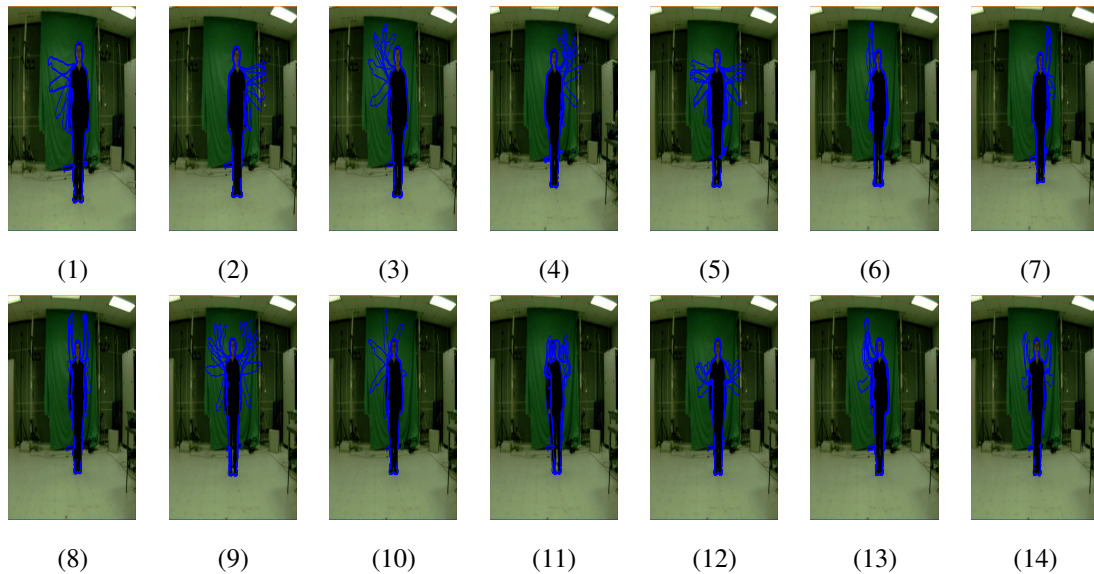


Figure 8: The different control gestures in the data set used in our gesture recognition experiment [46]. (1) Turn Right, (2) Turn Left, (3) Attention Right, (4) Attention Left, (5) Flap, (6) Stop Right, (7) Stop Left, (8) Stop Both, (9) Attention Both, (10) Start Engine, (11) Go back, (12) Close Distance, (13) Speed up, and (14) Come Near.



Figure 9: Some of the errors in the background subtraction errors in the control gesture dataset resulting in inaccurate silhouette extraction.

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 11 and 12 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 13-a and 13-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.

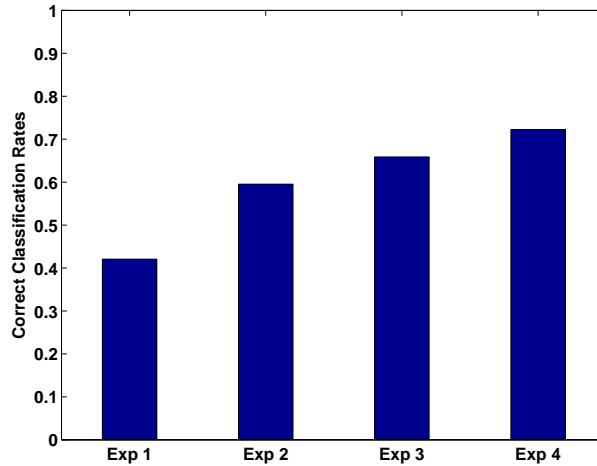


Figure 10: The correct classification results for using the template-based approach on the gesture dataset. Experiment 1: Bad segmentation of sequences and random initialization. Experiment 2: Correct segmentation of sequences and random initialization. Experiment 3: Initialization using the best sequence, all sequence used for training. Experiment 4: Initialization using the best Sequence, leave-one-out cross validation

### 5.2.2. 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 14 and 15 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 16 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

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 [46], shape only	leave one out	92.86
Lin et al [46], shape and motion	leave one out	95.24

Table 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 [46] on the same dataset.

of 91%, 85%, and 70% for the NN assignment method.

2. A careful study of the confusion matrices in Figure 16 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 [46]. Table 2 group our results for different approaches and evaluation methods as compared to [46]. We can note that our method managed to achieve comparable performance with 94% maximum recognition rate as compared 95.24% recognition rate reported in [46]. However, we were able to achieve this results using only the silhouette of the background subtracted sequences, while [46] was combining both the shape and motion information obtained using the color images and optical flow computations. In fact, our results exceed the performance reported in the same paper using only the shape cues for recognition.

## 6. Conclusion

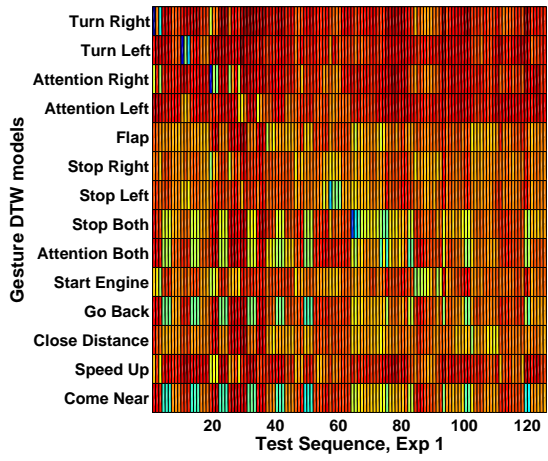
In this paper, we presented a novel gesture recognition technique using shape manifolds. Contours of the silhouette are extracted and represented as 2D closed elastic curves parameterized using the square-root parametrization. This representation is intrinsically invariant to both translation, scale, 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.

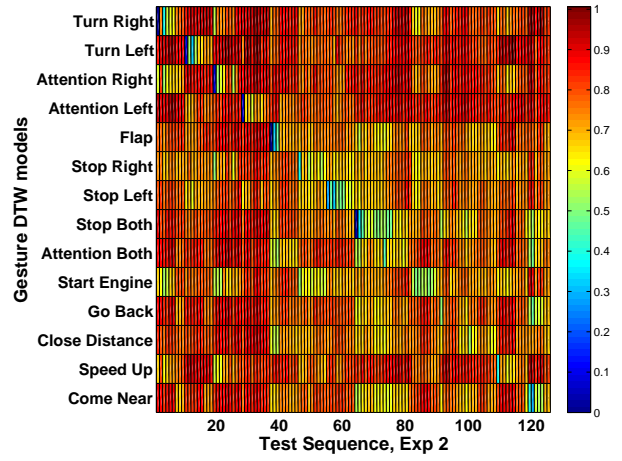
- [1] M. Hasanuzzaman, T. Zhang, V. Ampornaramveth, H. Gotoda, Y. Shirai, H. Ueno, Adaptive visual gesture recognition for human-robot interaction using a knowledge-based software platform, *Robot. Auton. Syst.* 55 (2007) 643–657.
- [2] M.-A. Bhuiyan, M. Islam, N. Begum, M. Hasanuzzaman, C. H. Liu, H. Ueno, Vision based gesture recognition for human-robot symbiosis, *Tenth Intl. Conf. on Computer and information technology, ICCIT 2007 (2007)* 1–6.
- [3] J. K. Aggarwal, S. 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, Washington, DC, USA*, pp. 640–647.
- [4] D. Gavrilu, The visual analysis of human movement: A survey, *Computer Vision and Image Understanding* 73 (1999) 82–98.
- [5] W. Hu, T. Tan, L. Wang, 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 (2004) 334–352.
- [6] T. B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, *Comput. Vis. Image Underst.* 104 (2006) 90–126.
- [7] D. Weinland, E. Boyer, Action recognition using exemplar-based embedding, in: *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Anchorage, Alaska*, pp. 1–7.
- [8] A. Efros, A. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: *Proc. of IEEE International Conf. on Computer Vision, Nice, France*, pp. 726–733.
- [9] Y. Sheikh, M. Sheikh, M. Shah, Exploring the space of a human action, in: *Proc. of IEEE International Conf. on Computer Vision, volume 1, Beijing, China*, pp. 144–149.
- [10] C. Bregler, Learning and recognizing human dynamics in video sequences, in: *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, San Juan, Puerto Rico*, pp. 568–574.
- [11] A. Bobick, J. Davis, The recognition of human movement using temporal templates, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 23 (2001) 257–267.
- [12] A. Veeraraghavan, A. Roy-Chowdhury, R. Chellappa, Matching shape sequences in video with an application to human movement analysis, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 27 (2005) 1896–1909.
- [13] L. Wang, D. Suter, Learning and matching of dynamic shape manifolds for human action recognition, *IEEE Trans. on Image Processing* 16 (2007) 1646–1661.
- [14] H. Freeman, On the encoding of arbitrary geometric configurations, *IRE Trans. Electron. Comput.* 10 (1961) 260–268.
- [15] C. Zahn, R. Roskies, Fourier descriptors for plane closed curves, *IEEE Transactions on Computers* 21 (1972) 269–281.
- [16] M. Hu, Visual pattern recognition by moment invariants, *IEEE Trans. on Information Theory* 8 (1962) 179–187.
- [17] S. Belongie, J. Malik, J. Puzicha, Shape Matching and Object Recognition Using Shape Contexts, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 24 (2002) 509–522.
- [18] A. Elgammal, C. Lee, Inferring 3D body pose from silhouettes using activity manifold learning, in: *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Washington D.C.*, pp. II: 681–688.
- [19] M. Black, Y. Yacoob, Parameterized modeling and recognition of activities, in: *Proc. of IEEE International Conf. on Computer Vision, Bombay*, pp. 120–127.
- [20] O. Tuzel, F. Porikli, P. Meer, Learning on lie groups for invariant detection and tracking, in: *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Anchorage, AK*, pp. 1–8.
- [21] E. Beggel, M. Werman, Affine invariance revisited, in: *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition,*

- Miami, FL, pp. II: 2087–2094.
- [22] P. Turaga, A. Veeraraghavan, 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, Anchorage, Alaska, pp. 1–8.
- [23] U. Grenander, M. I. Miller, A. Srivastava, Hilbert-Schmidt lower bounds for estimators on matrix Lie groups for ATR, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 20 (1998) 790–802.
- [24] A. Srivastava, E. Klassen, Monte Carlo extrinsic estimators for manifold-valued parameters, *IEEE Trans. on Signal Processing* 50 (2001) 299–308.
- [25] A. Goh, R. Vidal, Clustering and dimensionality reduction on riemannian manifolds, in: Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Anchorage, Alaska., pp. 1–7.
- [26] O. Tuzel, F. Porikli, P. Meer, Pedestrian detection via classification on riemannian manifolds, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 30 (2008) 1713–1727.
- [27] P. Fletcher, S. Venkatasubramanian, S. Joshi, Robust statistics on riemannian manifolds via the geometric median, in: Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Anchorage, Alaska, pp. 1–8.
- [28] A. Veeraraghavan, A. Roy-Chowdhury, 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, Washington, DC, pp. 730 – 737.
- [29] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. of The IEEE* 77 (1989) 257–286.
- [30] J. Yamato, J. Ohya, 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, Urbana-Champaign, IL, pp. 379–385.
- [31] M. Brand, N. Oliver, A. Pentland, Coupled hidden Markov models for complex action recognition, in: Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, pp. 994 –999.
- [32] S. Hongeng, R. Nevatia, Large-scale event detection using semi-hidden Markov models, in: Proc. of IEEE International Conf. on Computer Vision, Nice, France., pp. 1455–1462.
- [33] S. Wang, A. Quattoni, L. Morency, D. Demirdjian, T. Darrell, Hidden conditional random fields for gesture recognition, in: Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, volume 2, New York, NY, pp. 1521 – 1527.
- [34] L. Wang, 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, Minneapolis, MN, pp. 1–8.
- [35] A. Veeraraghavan, A. Srivastava, A. Roy-Chowdhury, R. Chellappa, Rate-invariant recognition of humans and their activities, *IEEE Trans. on Image Processing* 18 (2009) 1326 –1339.
- [36] S. Joshi, E. Klassen, A. Srivastava, I. Jermyn, A novel representation for riemannian analysis of elastic curves in  $R^n$ , in: Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Minneapolis, MN, pp. 1–7.
- [37] S. Joshi, E. Klassen, A. Srivastava, 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), Ezhou, China, pp. 387–398.
- [38] A. Srivastava, E. Klassen, S. H. Joshi, I. H. Jermyn, Shape analysis of elastic curves in euclidean spaces, *IEEE Trans. on Patt. Anal. and Mach. Intell.* in review (2010).
- [39] E. Klassen, A. Srivastava, W. Mio, Analysis of planar shapes using geodesic paths on shape spaces, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 26 (2004) 372–383.
- [40] L. Younes, Computable elastic distance between shapes, *SIAM Journal of Applied Mathematics* 58 (1998) 565–586.
- [41] L. Younes, P. W. Michor, J. Shah, D. Mumford, R. Lincei, A metric on shape space with explicit geodesics, *Matematica E Applicazioni* 19 (2008) 25–57.
- [42] A. Srivastava, S. H. Joshi, W. Mio, X. Liu, Statistical shape analysis: Clustering, learning, and testing, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 27 (2005) 590–602.
- [43] B. J. Frey, D. Dueck, Clustering by passing messages between data points, *Science* 315 (2007) 972–976.

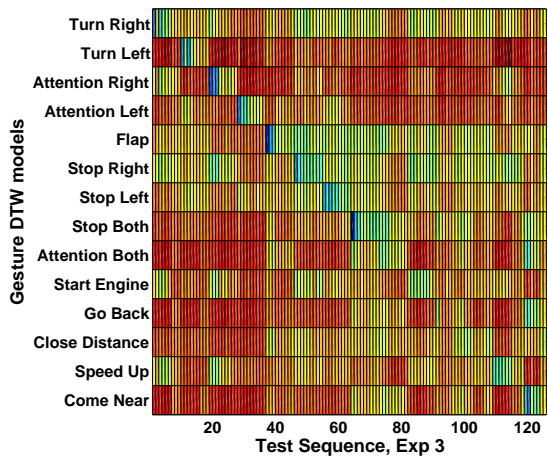
- [44] A. Elgammal, V. Shet, Y. Yacoob, L. Davis, Learning dynamics for exemplar-based gesture recognition, in: Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, volume 1, Madison, WI, pp. 571 – 578.
- [45] D. Weinland, E. Boyer, R. Ronfard, Action recognition from arbitrary views using 3d exemplars, in: Proc. of IEEE International Conf. on Computer Vision, Rio de Janeiro, Brazil, pp. 1 –7.
- [46] Z. Lin, Z. Jiang, L. Davis, Recognizing Actions by Shape-Motion Prototype Trees, in: Proc. of IEEE International Conf. on Computer Vision, Kyoto, Japan, pp. 444–451.
- [47] B. Frey, N. Jojic, Flexible models: A powerful alternative to exemplars and explicit models, in: IEEE Computer Society Workshop on Models vs. Exemplars in Computer Vision, Honolulu, Hawaii, pp. 34–41.
- [48] K. Toyama, A. Blake, Probabilistic tracking with exemplars in a metric space, *International Journal of Computer Vision* 48 (2002) 9–19.
- [49] H. Yang, A. Park, S. Lee, Gesture spotting and recognition for human-robot interaction, *IEEE transactions on Robotics* 23 (2007) 256–270.
- [50] D. Kendall, D. Barden, T. Carne, H. Le, *Shape and Shape Theory*, John Wiley and Sons, 1999.
- [51] I. Dryden, K. Mardia, *Statistical Shape Analysis*, John Wiley and Sons, 1998.
- [52] D. Kendall, D. Barden, T. Carne, H. Le, *Sape and Shape Theory*, John Wiley and Sons, 1999.
- [53] V. Patrangenaru, K. Mardia, Affine shape analysis and image analysis, *Stochastic Geometry, Biological Structure and Images*, Dept. of Statistics, University of Leeds (2003) 56–62.
- [54] C. Goodall, K. Mardia, Projective shape analysis, *Journal of Computational and Graphical Statistics* (1999) 143–168.
- [55] T. Cootes, C. Taylor, D. Cooper, J. Graham, Active Shape Models: Their Training and Application, *Computer Vision and Image Understanding* 61 (1995) 38–59.
- [56] U. Grenander, *General Pattern Theory*, Oxford university Press, 1993.
- [57] N. Vaswani, A. Roy-Chowdhury, 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, Madison, WI, pp. II – 633– 640.
- [58] S. Das, N. Vaswani, Nonstationary shape activities: Dynamic models for landmark shape change and applications, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 32 (2009) 579 – 592.
- [59] W. Mio, A. Srivastava, S. Joshi, On shape of plane elastic curves, *International Journal of Computer Vision* 73 (2007) 307–324.
- [60] M. do Carmo, *Riemannian Geometry*, Birkhuser, Englewoods Cliffs, 1992.
- [61] X. Pennec, I. P. Epidaure, Probabilities and statistics on riemannian manifolds: Basic tools for geometric measurements, in: *IEEE Workshop on Nonlinear Signal and Image Processing*, Antalya, Turkey, pp. 194–198.
- [62] H. Le, Locating frechet means with application to shape spaces, *Advances in Applied Probability* 33 (2001) 324–338.
- [63] P. Fletcher, C. Lu, S. Pizer, S. Joshi, Principal geodesic analysis for the study of nonlinear statistics of shape, *IEEE Transactions on Medical Imaging* 23 (2004) 995–1005.
- [64] E. Klassen, A. Srivastava, Geodesics between 3d closed curves using path-straightening, in: *Proc. of European Conference on Computer Vision*, Graz, Austria, pp. I: 95–106.
- [65] L. Rabiner, B.-H. Juang, *Fundamentals of speech recognition*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
- [66] P. Maurel, G. Sapiro, Dynamic shapes average, in: *Proc. Second IEEE Workshop Variational, Geometric and Level Set Methods in Computer Vision*, Nice, France.
- [67] L. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *The Annals of Mathematical Statistics* 41 (1970) 164–171.
- [68] A. Veeraraghavan, R. Chellappa, A. K. Roy-Chowdhury, The function space of an activity, in: *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Washington, DC, USA, pp. 959 – 968.



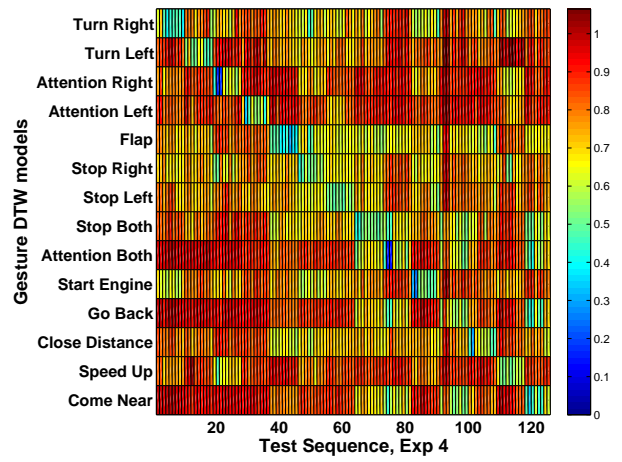
(a)



(b)



(c)



(d)

Figure 11: 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)

Turn Right	0.22	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.33
Turn Left	0.00	0.33	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33
Attention Right	0.00	0.00	0.44	0.00	0.00	0.00	0.00	0.00	0.22	0.00	0.00	0.00	0.00	0.33
Attention Left	0.00	0.00	0.11	0.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33
Flap	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.56	0.00	0.00	0.00	0.00	0.33
Stop Right	0.00	0.00	0.00	0.00	0.00	0.22	0.22	0.00	0.00	0.00	0.00	0.00	0.22	0.33
Stop Left	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Stop Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Both	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.33	0.44	0.00	0.00	0.00	0.00	0.11
Start Engine	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.22	0.44	0.00	0.00	0.00	0.00
Go Back	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.33	0.00	0.22	0.00	0.11
Close Distance	0.00	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.22
Speed Up	0.00	0.00	0.00	0.00	0.22	0.44	0.00	0.00	0.00	0.00	0.00	0.22	0.11	0.00
Come Near	0.00	0.00	0.00	0.00	0.11	0.11	0.00	0.22	0.00	0.00	0.00	0.11	0.11	0.33
	Turn Right	Turn Left	Attention Right	Attention Left	Flap	Stop Right	Stop Left	Stop Both	Attention Both	Start Engine	Go Back	Close Distance	Speed Up	Come Near

(a)

Turn Right	0.44	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.22	0.00	0.00	0.00	0.22
Turn Left	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.00
Attention Right	0.00	0.00	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.56	0.00	0.00	0.00	0.00
Attention Left	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Flap	0.00	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.44	0.00	0.00	0.00	0.00	0.00
Stop Right	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.22	0.00	0.00	0.00	0.00
Stop Left	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Stop Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Both	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.44	0.33	0.00	0.00	0.00	0.00	0.11
Start Engine	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.69	0.00	0.00	0.00
Go Back	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.56	0.00	0.22	0.11	0.11	0.00
Close Distance	0.00	0.00	0.00	0.00	0.11	0.11	0.00	0.00	0.00	0.11	0.00	0.22	0.00	0.00
Speed Up	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.22	0.00	0.00	0.44	0.00
Come Near	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.22	0.00	0.00	0.11	0.00	0.00	0.56
	Turn Right	Turn Left	Attention Right	Attention Left	Flap	Stop Right	Stop Left	Stop Both	Attention Both	Start Engine	Go Back	Close Distance	Speed Up	Come Near

(b)

Turn Right	0.56	0.00	0.00	0.00	0.00	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Turn Left	0.00	0.56	0.00	0.00	0.00	0.33	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Right	0.00	0.00	0.44	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Left	0.00	0.00	0.00	0.69	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Flap	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Stop Right	0.00	0.00	0.00	0.00	0.11	0.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Stop Left	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Stop Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Both	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.33	0.56	0.00	0.00	0.00	0.00	0.00
Start Engine	0.00	0.00	0.00	0.00	0.00	0.22	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.00
Go Back	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.44	0.00	0.11	0.11	0.00	0.00	0.00
Close Distance	0.00	0.00	0.00	0.00	0.44	0.11	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00
Speed Up	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.22	0.11	0.00
Come Near	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.22	0.11	0.56	0.00
	Turn Right	Turn Left	Attention Right	Attention Left	Flap	Stop Right	Stop Left	Stop Both	Attention Both	Start Engine	Go Back	Close Distance	Speed Up	Come Near

(c)

Turn Right	0.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00
Turn Left	0.00	0.89	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Right	0.11	0.00	0.56	0.00	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.00
Attention Left	0.00	0.00	0.00	0.73	0.00	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Flap	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Stop Right	0.22	0.00	0.00	0.00	0.00	0.56	0.00	0.00	0.00	0.22	0.00	0.00	0.00	0.00
Stop Left	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Stop Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Both	0.00	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.69	0.00	0.00	0.00	0.00	0.00
Start Engine	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.89	0.00	0.00	0.00	0.00
Go Back	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.22	0.00	0.11	0.00	0.00	0.33
Close Distance	0.00	0.00	0.00	0.00	0.44	0.00	0.00	0.11	0.00	0.00	0.00	0.44	0.00	0.00
Speed Up	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.22	0.00
Come Near	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.11	0.00	0.00	0.11	0.00	0.56
	Turn Right	Turn Left	Attention Right	Attention Left	Flap	Stop Right	Stop Left	Stop Both	Attention Both	Start Engine	Go Back	Close Distance	Speed Up	Come Near

(d)

Figure 12: The confusion matrix for gesture classification using the template-based method for experiments 1-4 are shown in a-d respectively

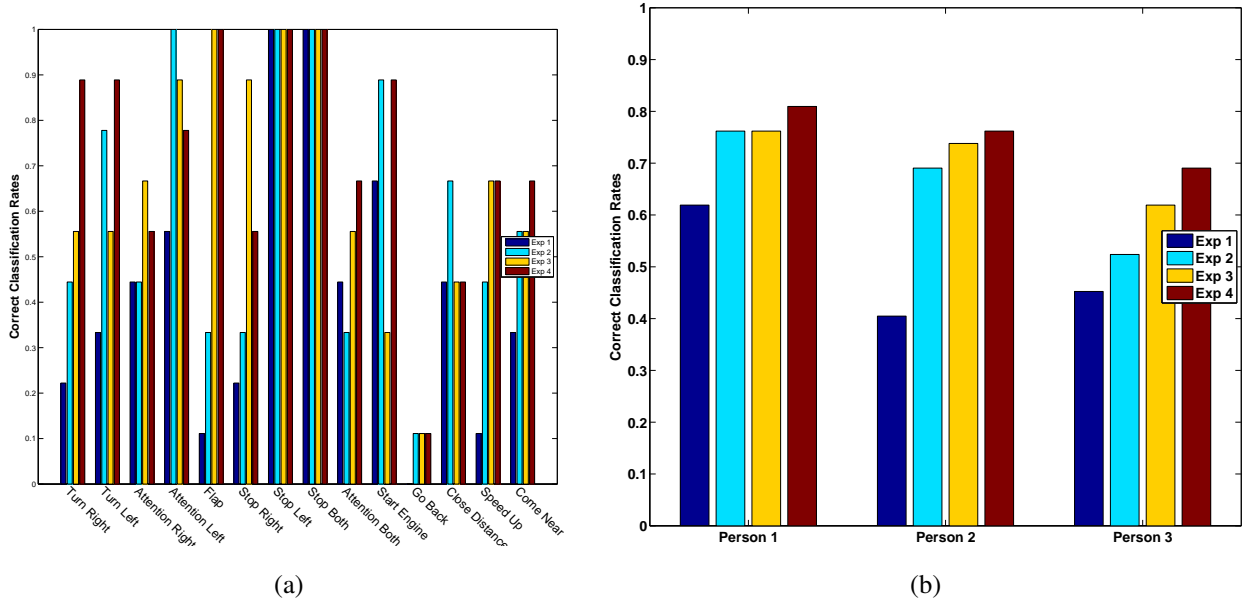


Figure 13: Gesture dataset classification results using the template-based approach (a) The correct classification rates for each gestures in different experiments (b) The correct classification rates for each person in different experiments

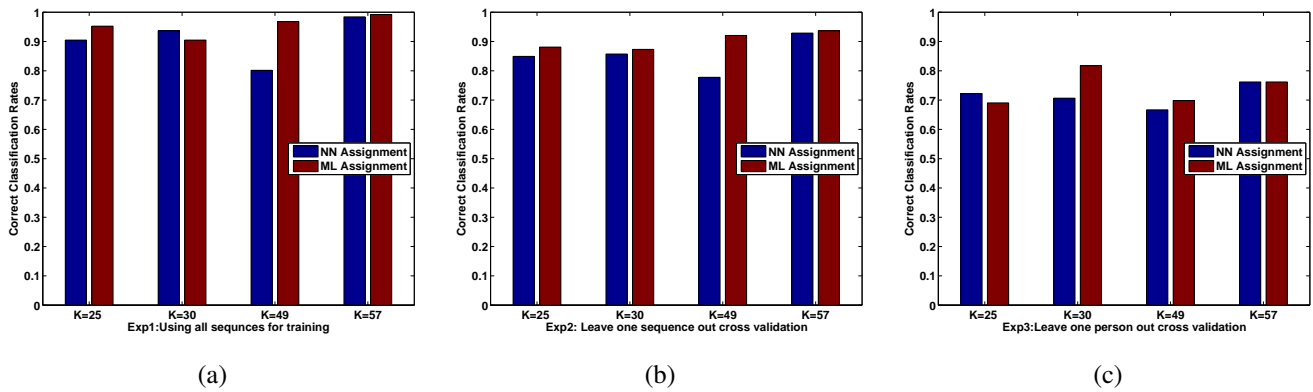


Figure 14: The gesture classification results using the graphical model approach, with 5 states HMM. It show the correct classification rates using each of ML and NN assignment and with varying number of clusters (a) Using all the sequences for training (b) Leaving one sequence out cross validation and (c) Leaving one person out cross validation

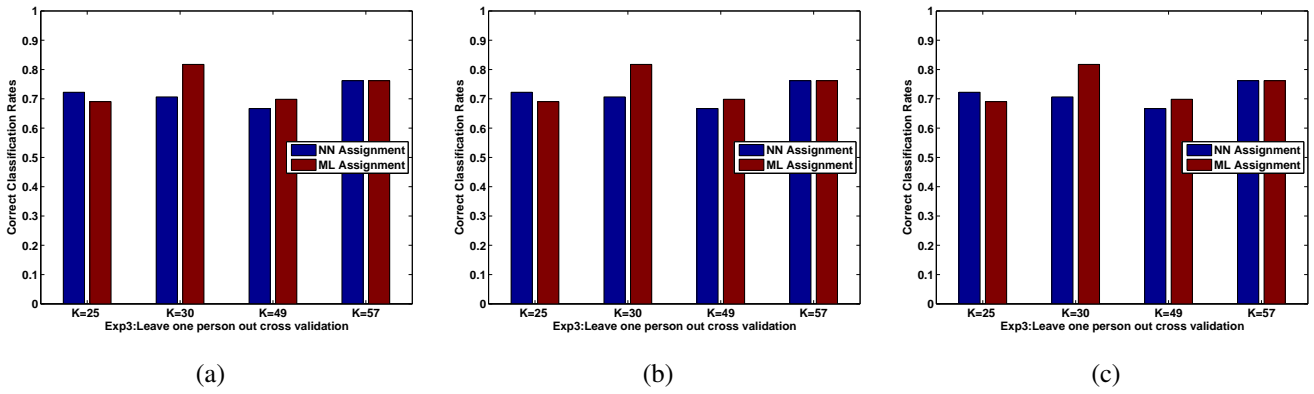


Figure 15: The gesture classification results using the graphical model approach, with 7 states HMM. It show the correct classification rates using each of ML and NN assignment and with varying number of clusters (a) Using all the sequences for training (b) Leaving one sequence out cross validation and (c) Leaving one person out cross validation

Turn Right	0.86	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.00	
Turn Left	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Attention Right	0.00	0.00	0.97	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Attention Left	0.00	0.00	0.00	0.93	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	
Flap	0.08	0.00	0.00	0.00	0.73	0.00	0.00	0.00	0.08	0.00	0.04	0.00	0.00	
Stop Right	0.00	0.00	0.01	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Stop Left	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	
Stop Both	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.01	0.00	0.00	0.00	0.00	0.01	
Attention Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	
Start Engine	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	
Go Back	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.75	0.00	0.00	0.19	
Close Distance	0.07	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.03	0.00	0.85	0.03	0.00	
Speed Up	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.85	0.00	
Come Near	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.06	0.79	
	Turn Right	Turn Left	Attention Right	Attention Left	Flap	Stop Right	Stop Left	Stop Both	Attention Both	Start Engine	Go Back	Close Distance	Speed Up	Come Near

(a)

Turn Right	0.81	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00
Turn Left	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Right	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Left	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Flap	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
Stop Right	0.00	0.01	0.00	0.00	0.00	0.97	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
Stop Left	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Stop Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Start Engine	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Go Back	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.82	0.00	0.00	0.17
Close Distance	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Speed Up	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Come Near	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.01	0.00	0.00	0.96
	Turn Right	Turn Left	Attention Right	Attention Left	Flap	Stop Right	Stop Left	Stop Both	Attention Both	Start Engine	Go Back	Close Distance	Speed Up	Come Near

(b)

Turn Right	0.73	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.15	0.00	0.00	0.00	0.00	0.00
Turn Left	0.00	0.85	0.00	0.14	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Right	0.00	0.00	0.96	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00
Attention Left	0.00	0.01	0.00	0.85	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Flap	0.04	0.00	0.00	0.00	0.82	0.00	0.00	0.00	0.12	0.00	0.01	0.00	0.00	0.00
Stop Right	0.00	0.00	0.04	0.00	0.00	0.93	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00
Stop Left	0.00	0.00	0.00	0.03	0.00	0.00	0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Stop Both	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.00	0.03	0.00	0.00	0.00	0.01
Attention Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Start Engine	0.01	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.00	0.00	0.00	0.00
Go Back	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.87	0.00	0.00	0.00	0.24
Close Distance	0.06	0.00	0.00	0.00	0.06	0.00	0.01	0.00	0.04	0.01	0.73	0.03	0.00	0.00
Speed Up	0.00	0.00	0.24	0.00	0.00	0.01	0.00	0.00	0.03	0.00	0.00	0.72	0.00	0.00
Come Near	0.03	0.00	0.01	0.00	0.00	0.03	0.00	0.10	0.00	0.00	0.21	0.00	0.04	0.58
	Turn Right	Turn Left	Attention Right	Attention Left	Flap	Stop Right	Stop Left	Stop Both	Attention Both	Start Engine	Go Back	Close Distance	Speed Up	Come Near

(c)

Turn Right	0.85	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00
Turn Left	0.00	0.93	0.00	0.06	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Attention Right	0.00	0.00	0.94	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00
Attention Left	0.00	0.04	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Flap	0.00	0.00	0.00	0.00	0.93	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00
Stop Right	0.00	0.00	0.01	0.00	0.01	0.94	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
Stop Left	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Stop Both	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.04
Attention Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Start Engine	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Go Back	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.81	0.00	0.31
Close Distance	0.01	0.00	0.00	0.00	0.10	0.00	0.00	0.01	0.00	0.00	0.88	0.00	0.00	0.00
Speed Up	0.00	0.00	0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.83	0.00	0.00
Come Near	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.03	0.00	0.12	0.01	0.00
	Turn Right	Turn Left	Attention Right	Attention Left	Flap	Stop Right	Stop Left	Stop Both	Attention Both	Start Engine	Go Back	Close Distance	Speed Up	Come Near

(d)

Turn Right	0.47	0.00	0.21	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.00
Turn Left	0.00	0.75	0.00	0.17	0.00	0.00	0.04	0.00	0.06	0.00	0.01	0.00	0.00	0.00
Attention Right	0.00	0.00	0.88	0.00	0.00	0.11	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
Attention Left	0.00	0.00	0.00	0.78	0.00	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Flap	0.08	0.00	0.01	0.00	0.87	0.00	0.00	0.15	0.06	0.00	0.01	0.01	0.00	0.00
Stop Right	0.00	0.00	0.14	0.00	0.00	0.81	0.00	0.01	0.00	0.19	0.00	0.01	0.03	0.00
Stop Left	0.00	0.00	0.00	0.01	0.00	0.00	0.97	0.00	0.00	0.01	0.00	0.00	0.00	0.00
Stop Both	0.00	0.00	0.00	0.00	0.00	0.00	0.82	0.04	0.00	0.12	0.00	0.00	0.00	0.01
Attention Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Start Engine	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.94	0.00	0.00	0.01	0.00	0.00
Go Back	0.03	0.00	0.03	0.03	0.00	0.00	0.03	0.10	0.00	0.03	0.50	0.00	0.00	0.26
Close Distance	0.08	0.01	0.00	0.01	0.19	0.01	0.00	0.03	0.03	0.03	0.00	0.78	0.00	0.01
Speed Up	0.00	0.00	0.53	0.00	0.00	0.06	0.00	0.00	0.00	0.01	0.00	0.00	0.39	0.00
Come Near	0.06	0.00	0.04	0.00	0.00	0.00	0.03	0.11	0.01	0.00	0.51	0.00	0.01	0.22
	Turn Right	Turn Left	Attention Right	Attention Left	Flap	Stop Right	Stop Left	Stop Both	Attention Both	Start Engine	Go Back	Close Distance	Speed Up	Come Near

(e)

Turn Right	0.73	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.14	0.00	0.03	0.00	0.00	0.00
Turn Left	0.00	0.75	0.00	0.14	0.00	0.00	0.06	0.00	0.00	0.00	0.06	0.00	0.00	0.00
Attention Right	0.00	0.00	0.88	0.00	0.00	0.06	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.00
Attention Left	0.00	0.00	0.00	0.78	0.00	0.00	0.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Flap	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.07	0.00	0.03	0.14	0.00	0.00
Stop Right	0.00	0.00	0.17	0.00	0.00	0.87	0.03	0.00	0.00	0.10	0.00	0.01	0.03	0.00
Stop Left	0.00	0.00	0.00	0.01	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Stop Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.92	0.01	0.00	0.04	0.00	0.00	0.03
Attention Both	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.00	0.00	0.00	0.00	0.03
Start Engine	0.07	0.00	0.03	0.00	0.00	0.01	0.00	0.00	0.00	0.89	0.00	0.00	0.00	0.00
Go Back	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.03	0.00	0.49	0.03	0.00	0.35
Close Distance	0.03	0.00	0.00	0.00	0.29	0.01	0.00	0.03	0.00	0.03	0.00	0.71	0.00	0.00
Speed Up	0.00	0.00	0.46	0.00	0.00	0.14	0.00	0.00	0.00	0.03	0.00	0.00	0.38	0.00
Come Near	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.12	0.07	0.00	0.33	0.03	0.00</	