Statistical and Dynamical Modeling of Riemannian Trajectories with Application to Human Movement Analysis

by

Rushil Anirudh

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved February 2016 by the
Graduate Supervisory Committee:

Pavan Turaga, Chair
Douglas Cochran
George Runger
Thomas Taylor

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

The data explosion in the past decade is in part due to the widespread use of rich sensors that measure various physical phenomenon – gyroscopes that measure orientation in phones and fitness devices, the Microsoft Kinect which measures depth information, etc. A typical application requires inferring the underlying physical phenomenon from data, which is done using machine learning. A fundamental assumption in training models is that the data is Euclidean, i.e. the metric is the standard Euclidean distance governed by the ℓ_2 norm. However in many cases this assumption is violated, when the data lies on non Euclidean spaces such as Riemannian manifolds. While the underlying geometry accounts for the non-linearity, accurate analysis of human activity also requires temporal information to be taken into account. Human movement has a natural interpretation as a trajectory on the underlying feature manifold, as it evolves smoothly in time. A commonly occurring theme in many emerging problems is the need to *represent, compare, and manipulate* such trajectories in a manner that respects the geometric constraints. This dissertation is a comprehensive treatise on modeling Riemannian trajectories to understand and exploit their statistical and dynamical properties. Such properties allow us to formulate novel representations for Riemannian trajectories. For example, the physical constraints on human movement are rarely considered, which results in an unnecessarily large space of features, making search, classification and other applications more complicated. Exploiting statistical properties can help us understand the *true* space of such trajectories. In applications such as stroke rehabilitation where there is a need to differentiate between very similar kinds of movement, dynamical properties can be much more effective. In this regard, we propose a generalization to the Lyapunov exponent to Riemannian manifolds and show its effectiveness for human activity analysis. The theory developed in this thesis naturally leads to several benefits in areas such as data mining, compression, dimensionality reduction, classification, and regression.

Dedicated to my parents Ravikumar and Sheela.

ACKNOWLEDGMENTS

First and foremost this dissertation would not have been possible without the help and support of my advisor Pavan Turaga. I feel especially proud to be his first student to begin his research group at ASU! Pavan has also been very patient and understanding with my utter lack of understanding of differential geometry, and a great mentor. I will always remember life lessons I have learned along the way. I would also like to thank my committee for giving me constructive feedback and showing a lot of enthusiasm for my research. I really appreciate their time and effort.

I would also like to thank the School of Electrical, Computer and Energy Engineering, its faculty, administrative staff and students for being helpful and efficient in all matters. I owe a big thanks to Prof. Andreas Spanias for his class on DSP, which I still remember fondly. I have been fortunate to work for EEE 120 as a teaching assistant, which was a thoroughly enjoyable experience. I am thankful to Prof. Joseph Palais and Mr. Clayton Javurek for having me as a part of the army of teaching assistants (nearly 100 TAs!).

I've been very fortunate to be a part of the School of Arts, Media and Engineering as a concentration student during my PhD. Being a part of such an interdisciplinary crowd of artists, dancers, musicians and engineers has expanded my world view significantly. I fondly recall many discussions I've had with various colleagues trying to understand the interplay between art+engineering, and why it matters. I still don't have an excellent answer but I think I have become much more appreciative of the role of the arts in todays world.

Outside academia, I have had the wonderful opportunity to spend three summers doing internships, making a lot of friends and more importantly spending the summer away from Arizona in cool 75F weather of the bay area. I would like to thank my mentors Vincent Duindam (Intuitive Surgical), Jason Laska (Dropcam), Timo Bremer and Jayaraman Thiagarajan (LLNL) for being patient and teaching me the mysterious ways of the industry.

When Kuldeep and I first joined Pavan's new fledgling group, we were the only two members and we would often discuss how the group should grow in the future. I am happy to report that a large part of that vision has come true, and we have really burgeoned into a nice little research group (in order of joining) - Rushil, Kuldeep, Qiao, Vinay, Suhas, Anirudh and our honorary group member Aaron. A big part of my PhD experience is the (1-2-3-4-?) hour long lunch discussions on various topics - morality, life and death, philosophy, politics, sport, science, engineering, "manifolds" etc. The fact that popular press has often reported on the very same topics only a few days later, is a testament to how effective we were in our procrastination.

The five and a half years I spent in Tempe have been wonderful and eventful to say the least. I have learned a lot of life lessons, which I hope will be of some use in the future. I've been fortunate to make a lot of friends along the way (in order of acquaintance): Apt 11 group- Suhas, Kuldeep, Karthik; C6++ group - Prasad, Suhas, Anu, Hyma, Shashank, Vaibhav; Ramsri, Adithya Rajan, Jayaraman, Karthikeyan, Deeptha, NITK group "The real GCG": Suhas, Sarkar, Sridhar, and friends at AME: Mike and Varsha.

Through all these years, Aparna Unnikrishnan stands out for being my safe place, my comfort zone, and my cushion. I owe her a big thank you – life in grad school would have been much more stressful, and much less eventful if not for her.

Finally, I owe every success of mine to my wonderful family in India, who really shaped me during my formative years and continue to make me the individual I am - My parents, Nikhil, Chittha, Thatha Paaty, Achan Ammamma.

The work in this thesis was partly supported by NSF grant 1320267 and ASU startup grant.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

The last few years have seen a proliferation of sensors in every day life. More than ever, we monitor our health using wearable health trackers, we rely on security cameras to keep our homes safe, we interact with computing systems and play games using only our gestures and movements. In all of these scenarios, a sensor is able to record rich data about its environment which is used to make inferences using machine learning algorithms. Typically, application specific features are extracted from the sensor data, before deploying inference algorithms for tasks such as classification, recognition, detection etc. In many situations the extracted features naturally lie on Riemannian manifolds, which means that traditional inference algorithms need significant generalization by taking the geometry into account, which is often not trivial.

For example, in image analysis features such as contours of objects [60], skeletons from depth sensors [125], the space of $d \times d$ covariance matrices or tensors which appear both in medical imaging [82] as well as texture analysis [120] etc. and in video analysis, video modeling by linear dynamic systems [119], and tensor decomposition [75] etc. For human activity and movement analysis - including recognition, search, exploration and visualization of common everyday activities, some of the popular manifold valued features are include shape silhouettes on the Kendall's shape space [124], pairwise transformations of skeletal joints on $SE(3) \times SE(3) \cdots \times SE(3)$ [125], linear dynamical system on the Grassmann manifold [118], and histogram of oriented optical flow (HOOF) on a hyper-sphere [31].

In many of these cases, typically a feature is extracted per frame ('skeleton', 'shapes', or ' texture covariances') which has a natural geometric interpretation. Further, since the real world phenomenon observed often vary smoothly, the resulting features vary smoothly on the manifold, enabling us to interpret the collection of time varying features as a smooth

curve or a time-series on the Riemannian space. A commonly occurring theme in many applications is the need to *represent, compare, and manipulate* such curves in a manner that respects the geometric constraints. Any computations on these trajectories can easily become overwhelming depending on the sampling-rate (or frame-rate for videos). The task is further complicated by the fact that for data lying on manifolds, standard notions of distance, statistics, quantization etc. need significant modification to account for the non-linearity of the underlying space. As a result, basic computations such as geodesic distance, finding the sample mean etc. are highly involved in terms of computational complexity, and often result in iterative procedures further increasing the computational load making them impractical. Another problem that arises for computations on human movement and activities, is the need for metrics that are invariant to speed of the movement. That is, the distance between two similar movements or actions is expected to be small irrespective of the speed of the subject performing the action. Operating with such complex data can easily become overwhelming for any of the existing machine learning tools, due to the increased computational complexity. Moreover, many of the existing tools cannot work directly with manifold valued data without introducing unwanted artifacts.

In this dissertation, we present a framework to study such high dimensional non-linear trajectories, in the context of human movement and activity analysis. We propose algorithms and tools to efficiently represent, compare, and explore human activities represented as Riemannian trajectories. We first describe these methods briefly, followed by a summary of contributions.

Symbolic Approximation for Riemannian trajectories First, we propose a framework that generalizes a popular indexing technique used to mine and search for vector space time series data known as Symbolic Aggregate Approximation (SAX) [71] to Riemannian manifolds. To the best of our knowledge, we are the first to propose such an indexing scheme for Riemannian trajectories. The main idea is to replace Riemannian trajectories with abstract *symbols* or *prototypes*, that can be learned offline. Symbolic approximation

is a combination of discretization and quantization on manifold spaces, which allows us to approximate distance metrics between trajectories in a quick and efficient manner. Another advantage is extremely fast searching that is possible because the search is limited to the symbolic space. Further, to enable efficient searching techniques, we develop prototypes or symbols which divide the space into equiprobable regions by proposing the first manifold generalization of a conscience based competitive learning algorithm [36]. Using these prototypes, we demonstrate that signals or trajectories on manifolds can be approximated effectively such that the resulting metric remains close to the metric on the original feature space, thereby guaranteeing accurate recognition and search. While this framework is applicable to general high-dimensional feature sequences, we demonstrate its utility on a few common video-analysis problems such as activity analysis and dynamic texture modeling.

Functional codes for human actions Next, we employ a functional interpretation of Riemannian trajectories to obtain metrics that are invariant to temporal re-parameterization (or warping) which can distort distance measures significantly, especially in the context of human actions. The most common way to solve for the mis-alignment problem is to use dynamic time warping (DTW) which originally found its use in speech processing [19]. However, DTW behaves as a similarity measure instead of a true distance metric in that it does not naturally allow the estimation of statistical measures such as mean and variance of action trajectories. We seek a representation that is highly discriminative of different classes while factoring out temporal warping to reduce the variability within classes. Learning such a representation is complicated when the features extracted are non-Euclidean (i.e. they do not obey conventional properties of the Euclidean space). Finally, typical representations for action recognition tend to be extremely high dimensional in part because the features are extracted per-frame and stacked. Any computation on such non-linear trajectories can become very easily involved.

Dynamical analysis of Riemannian trajectories Finally we introduce a new algorithm to extract a Lyapunov feature to understand the dynamical properties of such trajectories. Such features of the dynamical systems have proven to be successful in many applications that require distinguishing between very similar kinds of actions. For example, in movement quality assessment, the level of chaos can be a good proxy for the quality of movement. The proposed algorithm enables the computation of such chaotic measures for Riemannian trajectories.

A summary of contributions

1. We first present a geometry based data-adaptive strategy for indexing Riemannian sequences. We demonstrate the effectiveness on three manifolds namely the hypersphere, the Grassmann manifold and the product space of $SE(3) \times \cdots \times SE(3)$.

2. We propose the first generalization of competitive learning algorithms to Riemannian manifolds for this task, such that they are able learn prototypes which enable efficient searching.

3. The resulting framework allows the comparison between two manifold sequences at speeds nearly $100\times$ faster than geodesic based comparisons in applications such as activity recognition and discovery. The speed up can be achieved with minimal loss of accuracy as compared to the original features.

4. By altering the competitive learning bias, a new algorithm is proposed for online diverse sampling for video summarization that more efficient in terms of memory and speed than exisiting methods.

5. Next, we present the extension of the TSRVF representation for human actions by modeling trajectories on the Grassmann manifold and the product space of $SE(3) \times \cdots \times SE(3)$, and the space of SPD matrices.

6. We propose the first embedding of Riemannian trajectories to lower dimensional spaces in a warp invariant manner which enables faster computations and lower storage. Using multiple different embedding algorithms – PCA, KSVD, LCKSVD, we show how high dimensional Riemannian trajectories can be cast into low dimensional vectors without loss of recognition accuracy, and speed invariance.

7. The embedded features outperform many state-of-the-art approaches in action recognition on three benchmark datasets. Their effectiveness is also demonstrated in action clustering and diverse action sampling.

8. A detailed stability analysis, robustness to factors such as noise, sampling rate etc. are studied for the TSRVF.

9. A new algorithm to extract the Lyapunov feature for Riemannian trajectories is proposed. The computed feature is shown to be a good measure of the amount of chaos within a Riemannian trajectory.

Organization of the dissertation Chapter 2 begins with a formal study of Riemannian geometry, including the exponential and logarithmic maps of manifolds that are dealt with in this dissertation - Grassmann, Hypersphere, and the product space of $SE(3)$. Next in chapter 3, the symbolic approximation strategy to efficiently compute distances on manifold sequences is presented. The conscience based competitive learning algorithm for Riemannian manifolds is described in section 2.2 and algorithm 1. Next, the problem of latent variable models for human actions is introduced in chapter 5. The procedure to perform mfPCA is described in algorithm 4, following which experiments demonstrate its effectiveness in activity analysis. In section 6, a dictionary learning based approach to obtain a generative model for human activities is described, which models the subspace of human actions effectively. Chapter 8 concludes the dissertation, presents a proposal for the further study of questions raised here. This includes presenting several future directions of research ideas and experimental work. This dissertation combines material from several

peer-reviewed publications and manuscripts under review (at the time of writing) by the author, these are listed next for clarity and convenience. Chapter 3 from [9], Chapter 4 from [10], Chapter 5 from [11, 13, 14], and finally Chapter 7 from [15].

Chapter 2

MATHEMATICAL PRELIMINARIES

A topological space is a set $\mathcal{M}$, with a specified class of subsets or neighborhoods ϕ such that 1) ϕ & $\mathcal{M}$ are open, 2) The intersection of any two open sets is open and 3) The union of any number of open sets is open. A topological space is called *Hausdorff* if any distinct two points of $\mathcal{M}$ possess non-intersecting neighborhoods. A function $f : \mathcal{M} \to$ S is said to be continuous if the inverse image of every open set in S, that may or may not be the same as $\mathcal{M}$. If the function f has an inverse that is also continuous then $\mathcal{M}$ & S are said to be homeomorphic.

Finally, a real manifold $\mathcal{M}$ of dimension N, is a topological - Hausdorff space that is locally homeomorphic to $\mathbb{R}^N$ and is second countable. That is, for each $p \in \mathcal{M}$, there exists an open neighborhood U of p and a mapping $\phi : U \to \mathbb{R}^n$ such that $\phi(U)$ is open in $\mathbb{R}^n$ and $\phi : U \to \phi(U)$ is a diffeomorphism [22]. The pair (U, ϕ) is called a *coordinate chart* for the points that fall in U.

The Euclidean space $\mathbb{R}^d$ is studied as a manifold using the identity chart. The complex coordinate space $\mathbb{C}^n$ becomes a real $2n$-dimensional manifold via the chart $\mathbb{C}^n \to \mathbb{R}^{2n}$ replacing every complex coordinate z_j by a pair of real coordinates Re z_j, Im z_j. The sphere $S^n = \{x \in \mathbb{R}^{n+1} : \sum_{i=0}^{n} x_i^2 = 1\}$ is made into a smooth manifold of dimension n, by means of the two stereographic projections onto $\mathbb{R}^n \cong \{x \in \mathbb{R}^{n+1} : x_0 = 0\}$, from the North and South poles $(\pm 1, 0, \ldots, 0)$. The corresponding change of coordinates is given by $(x_1, \ldots, x_n) \to (x_1/|x|^2, \ldots, x_n/|x|^2)$. In computer vision, the Grassmann and the Stiefel manifolds are used in several applications as described earlier. The Grassman manifold is the space of d-dimensional subspaces in $\mathbb{R}^n$ and the Stiefel manifold is the space of orthonormal d-frames in $\mathbb{R}^n$.

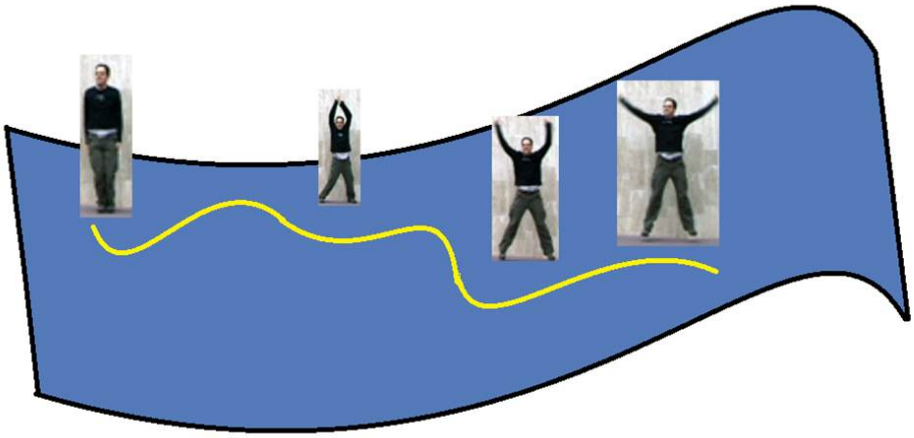


Figure 1: Features are extracted on each image/frame of a video depicting human activity resulting in a sequence of features evolving over time, or a manifold valued time series. The idea is shown here using sample data from the Wiezmann Data set for human action [48]

Manifold Sequences As shown in fig 1, like in Euclidean space, a sequence of points that evolve over time on the manifold can be studied as a time series. To analyze sequences or curves on manifolds, one needs to take recourse to understanding tangent-space and exponential mappings. A tangent-space at a point of a manifold $\mathcal{M}$ is obtained by considering the velocities of differentiable curves passing through the given point. i.e. for a point $p \in \mathcal{M}$, a differentiable curve passing through it is represented as $\beta : (-\delta, \delta) \rightarrow \mathcal{M}$ such that $\beta(0) = p$. The velocity $\dot{\beta}(0)$ refers to the velocity of the curve at p. This vector has the same dimension as the manifold and is a tangent vector to $\mathcal{M}$ at p. The set of all such tangent vectors is called the tangent space to $\mathcal{M}$ at p. The tangent space $T_p(\mathcal{M})$ is always a vector-space.

Riemannian Metric The distance between two points on a manifold is measured by means of the 'length' of the shortest curve connecting the points. The notion of length is formalized by defining a **Riemannian metric**, which is a map $\langle \cdot, \cdot \rangle$ that associates to each

point $p \in \mathcal{M}$ a symmetric, bilinear, positive definite form on the tangent space $T_p(\mathcal{M})$. The Riemannian metric allows one to compute the infinitesimal length of tangent-vectors along a curve. The length of the entire curve is then obtained by integrating the infinitesimal lengths of tangents along the curve. i.e. given $p, q \in \mathcal{M}$, the distance between them is the infimum of the lengths of all smooth paths on $\mathcal{M}$ which start at p and end at q:

$$d(p,q) = \inf_{\{\beta:[0,1]\mapsto\mathcal{M}|\beta(0)=p,\beta(1)=q\}} L[\beta], \text{where,} \tag{2.1}$$

$$L[\beta] = \int_0^1 \sqrt{\left(\left\langle \dot{\beta}(t), \dot{\beta}(t) \right\rangle\right)} dt \tag{2.2}$$

If $\mathcal{M}$ is a Riemannian manifold and $p \in \mathcal{M}$, the **exponential map** $\exp_p : T_p(\mathcal{M}) \to \mathcal{M}$, is defined by $\exp_p(v) = \beta_v(1)$ where β_v is a specific geodesic in the direction of the tangent-vector v. The inverse mapping $\exp_p^{-1} : \mathcal{M} \to T_p$ called the inverse exponential map at a 'pole', takes a point on the manifold and returns a point on the tangent space of the pole.

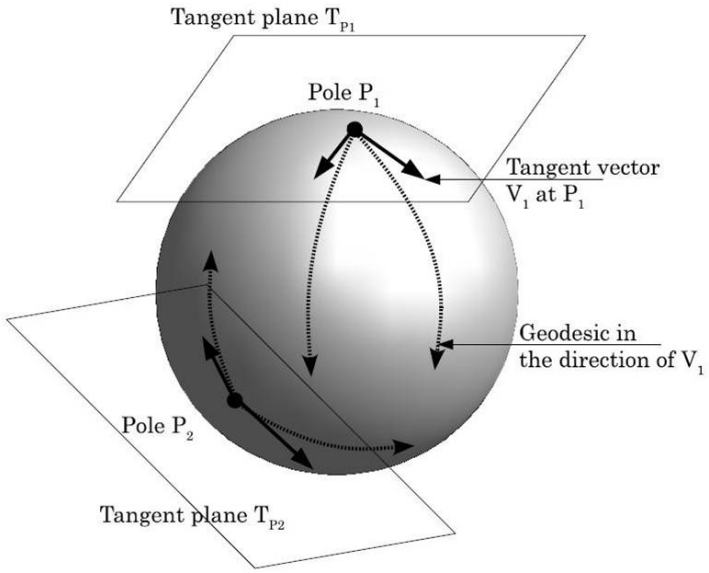


Figure 2: Exponential, Inverse exponential maps and the Tangent Space.

In this section we will outline the geometric properties of the manifolds considered in this work, namely the Grassmannian, hyper-sphere and the space of $SE(3) \times \ldots SE(3)$. For

an overview on Riemannian geometry and topology, we refer the readers to useful resources on the topic [3, 22]. Next we describe the different features and their respective geometric spaces.

1 Grassmann Manifold As A Shape Space

We represent a shape as a $m \times 2$ matrix $L = [(x_1, y_1);(x_2, y_2); \ldots; (x_m, y_m)]$, of the set of m landmarks of the zero-centered shape. The *affine shape space* [47] is useful to remove the effects of small variations in camera location or small changes in the pose of the subject. Affine transforms of the base shape L_{base} can be expressed as $L_{affine}(A) = L_{base} * A^T$, and this multiplication by a full-rank matrix on the right preserves the column-space of the matrix L_{base}. Thus, the 2D subspace of $\mathbb{R}^m$ spanned by the columns of the matrix L_{base} is an *affine-invariant* representation of the shape. i.e. $span(L_{base})$ is invariant to affine transforms of the shape. Subspaces such as these can be identified as points on a Grassmann manifold, $\mathcal{G}$ [119].

An equivalent definition of the Grassmann manifold is as follows: To each k-plane, ν in $\mathcal{G}_{k,m-k}$ corresponds a unique $m \times m$ orthogonal projection matrix, P which is idempotent and of rank k. If the columns of a tall $m \times k$ matrix Y spans ν then $YY^T = P$. Then the set of all possible projection matrices $\mathbb{P}$, is diffeomorphic to $\mathcal{G}$. The identity element of $\mathbb{P}$ is defined as $Q = diag(I_k, 0_{m-k,m-k})$, where $0_{a,b}$ is an $a \times b$ matrix of zeros and I_k is the $k \times k$ identity matrix. The Grassmann manifold $\mathcal{G}$(or $\mathbb{P}$) is a quotient space of the orthogonal group, $O(m)$. Therefore, the geodesic on this manifold can be made explicit by lifting it to a particular geodesic in $O(m)$ [103]. Then the tangent, X, to the lifted geodesic curve in $O(m)$ defines the velocity associated with the curve in $\mathbb{P}$. The tangent space of $O(m)$ at identity is $o(m)$, the space of $m \times m$ skew-symmetric matrices, X. Moreover in $o(m)$, the Riemannian metric is just the inner product of $\langle X_1, X_2 \rangle = \text{trace}(X_1 X_2^T)$ which is inherited by $\mathbb{P}$ as well. This metric topology is induced by the Hilbert-Schmidt norm on the space of

matrices.

$$d^2(P_1, P_2) = tr(P_1 - P_2)^T(P_1 - P_2) \tag{2.3}$$

The distance metric defined in (2.3) is closely related to the Procrustes measure on the Grassmann manifold which has previously been used in [26].

The geodesics in $\mathbb{P}$ passing through the point Q (at time $t = 0$) are of the type $\alpha : (-\epsilon, \epsilon) \mapsto \mathbb{P}, \alpha(t) = \exp(tX)Q\exp(-tX)$, where X is a skew-symmetric matrix belonging to the set M where

$$M = \left\{ \begin{bmatrix} 0 & A \\ -A^T & 0 \end{bmatrix} : A \in \mathbb{R}^{k,n-k} \right\} \subset o(m) \tag{2.4}$$

Therefore the geodesic between Q and any point P is completely specified by an $X \in M$ such that $\exp(X)Q\exp(\text{-}X) = P$. We can construct a geodesic between any two points $P_1, P_2 \in \mathbb{P}$ by rotating them to Q and some $P \in \mathbb{P}$. Readers are referred to [103] for more details on the exponential and logarithmic maps of $\mathcal{G}_{k,m-k}$.

The projection $\Pi : \mathbb{R}^{m \times m} \to \mathbb{P}_{m,d}$ is given by:

$$\Pi(M) = UU^T \tag{2.5}$$

where $M = USV^T$ is the d-rank SVD of M.

Given a set of sample points on the Grassmann manifold represented uniquely by projectors $\{P_1, P_2, ...P_N\}$, we can compute the mean [117] by first computing the mean of the P_i's and then projecting it to the manifold as follows :

$$\mu_{ext} = \Pi(P_{avg}), \text{where } P_{avg} = \frac{1}{N}\sum_{i=1}^{N} P_i \tag{2.6}$$

2 Histograms On The Hypersphere

As described in [31], optical flow is a natural feature for motion sequences. Directions of optical flow vectors are computed for every frame, then binned according to their primary

angle with the horizontal axis and weighted according to their magnitudes. Using magnitudes alone is susceptible to noise and can be very sensitive to scale. Thus all optical flow vectors, $v = [x, y]^T$ with direction $\theta = tan^{-1}(\frac{y}{x})$ in the range

$$-\frac{\pi}{2} + \pi\frac{b-1}{B} \leq \theta < -\frac{\pi}{2} + \pi\frac{b}{B} \tag{2.7}$$

will contribute by $\sqrt{x^2 + y^2}$ to the sum in bin b, $1 \leq b \leq B$, out of a total of B bins. Finally, the histogram is normalized to sum up to 1. Each frame is represented by one histogram and hence a sequence of histograms are used to describe an activity. The histograms $h_t = [h_{t;1}, \ldots, h_{t;B}]$ can be re-parameterized to the *square root representation* for histograms, $\sqrt{\mathbf{h_t}} = [\sqrt{h_{t;1}}, \ldots, \sqrt{h_{t;B}}]$ such that $\sum_{i=1}^{B}(\sqrt{h_{t;i}})^2 = 1$. The Riemannian metric between two points R_1 and R_2 on the hypersphere is $d(R_1, R_2) = cos^{-1}(R_1^T R_2)$. This projects every histogram onto the unit B-dimensional hypersphere or $\mathbb{S}^{B-1}$. From the differential geometry of the sphere, the exponential map is defined as [104]

$$\exp_{\psi_i}(\upsilon) = \cos(||\upsilon||_{\psi_i})\psi_i + \sin(||\upsilon||_{\psi_i})\frac{\upsilon}{||\upsilon||_{\psi_i}} \tag{2.8}$$

Where $\upsilon \in T_{\psi_i}(\Psi)$ is a tangent vector at ψ_i and $||\upsilon||_{\psi_i} = \sqrt{\langle \upsilon, \upsilon \rangle_{\psi_i}} = (\int_0^T \upsilon(s)\upsilon(s)ds)^{\frac{1}{2}}$. In order to ensure that the exponential map is a bijective function, we restrict $||\upsilon||_{\psi_i} \in [0, \pi]$. The truncation of the domain of the the exponential map is made in accordance to the injectivity radius, which is the largest radius for which the exp map is a diffeomorphism. For the sphere, the injectivity radius is π. Points that lie beyond the injectivity radius have a shorter path connecting them to ψ_i, which determines their geodesic distance incorrectly. The logarithmic map from ψ_i to ψ_j is given by

$$\overrightarrow{\psi_i\psi_j} = \log_{\psi_i}(\psi_j) = \frac{\mathbf{u}}{(\int_0^T \mathbf{u}(s)\ \mathbf{u}(s)ds)^{\frac{1}{2}}}\cos^{-1}\langle \psi_i, \psi_j \rangle, \tag{2.9}$$

with $\mathbf{u} = \psi_i - \langle \psi_i, \psi_j \rangle \psi_j$.

For action recognition, we represent a stick figure as a combination of relative transformations between joints, as proposed in [125]. The resulting feature for each skeleton is interpreted as a point on the product space of $SE(3) \times \cdots \times SE(3)$. The skeletal representation explicitly models the 3D geometric relationships between various body parts

using rotations and translations in 3D space [125]. These transformation matrices lie on the curved space known as the Special Euclidean group $SE(3)$. Therefore the set of all transformations lies on the product space of $SE(3) \times \cdots \times SE(3)$.

The special Euclidean group, denoted by $SE(3)$ is a Lie group [78], containing the set of all 4×4 matrices of the form

$$P(R, \overrightarrow{v}) = \begin{bmatrix} R & \overrightarrow{v} \\ 0 & 1 \end{bmatrix}, \tag{2.10}$$

where R denotes the rotation matrix, which is a point on the special orthogonal group $SO(3)$ and $\overrightarrow{v}$ denotes the translation vector, which lies in $\mathbb{R}^3$. The 4×4 identity matrix I_4 is an element of $SE(3)$ and is the identity element of the group. The tangent space of $SE(3)$ at I_4 is called its Lie algebra – denoted here as $\mathfrak{se}(3)$. It can be identified with 4×4 matrices of the form [1]

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & \overrightarrow{v} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega 1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{2.11}$$

where $\widehat{\omega}$ is a 3×3 skew-symmetric matrix and $\overrightarrow{v} \in \mathbb{R}^3$. An equivalent representation is $\xi = [\omega_1, \omega_2, \omega_3, v_1, v_2, v_3]^T \in \mathbb{R}^6$. For the exponential and inverse exponential maps, we use the expressions provided on p. 413-414 in [78], we reproduce them for completeness here. The exponential map is given by

$$\exp \widehat{\xi} = \begin{bmatrix} I & \overrightarrow{v} \\ 0 & 1 \end{bmatrix} \quad \omega = 0 \text{ and } \exp \widehat{\xi} = \begin{bmatrix} e^{\widehat{\omega}} & A\overrightarrow{v} \\ 0 & 1 \end{bmatrix} \quad \omega \neq 0, \tag{2.12}$$

where $e^{\widehat{\omega}}$ is given explicitly by the Rodrigues's formula – $= I + \frac{\widehat{\omega}}{\|\omega\|}\sin\|\omega\| + \frac{\widehat{\omega}^2}{\|\omega\|^2}(1 - \cos\|\omega\|)$, and $A = I + \frac{\widehat{\omega}}{\|\omega\|^2}(1 - \cos\|\omega\|) + \frac{\widehat{\omega}^2}{\|\omega\|^3}(\|\omega\| - \sin\|\omega\|)$.

[1] We are following the notation to denote the vector space ($\xi \in \mathbb{R}^6$) and the equivalent Lie algebra representation ($\widehat{\xi} \in \mathfrak{se}(3)$) as described in p. 411 of [78].

The inverse exponential map is given by

$$\widehat{\xi} = \log \begin{bmatrix} R & \overrightarrow{v} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \widehat{\omega} & A^{-1}\overrightarrow{v} \\ 0 & 0 \end{bmatrix}, \tag{2.13}$$

where $\widehat{\omega} = \log R$, and

$$A^{-1} = I - \frac{1}{2}\widehat{\omega} + \frac{2\sin\|\omega\| - \|\omega\|(1+\cos\|\omega\|)}{2\|\omega\|^2 \sin\|\omega\|}\widehat{\omega}^2 \quad \omega \neq 0,$$

when $\omega = 0$, then $A = I$.

Parallel transport on the product space is the parallel transport of the point on component spaces. Let $T_O(SO(3))$ denote the tangent space at $O \in SO(3)$, then the parallel transport of a $W \in T_O(SO(3))$ from O to $I_{3\times 3}$ is given by $O^T W$. For more details on the properties of the special Euclidean group, we refer the interested reader to [78].

3 The Space Of Symmetric Positive Definite (SPD) Matrices

We utilize the covariance features for the problem of Visual Speech Recognition (VSR). These features first introduced in [120] have become very popular recently due to their ability to model unstructured data from images such as textures and scenes. A covariance matrix of image features such as pixel locations, intensity and their first and second derivatives is constructed to represent the image. As described in [82], for a rectangular region R, let $\{z_k\}_{k=1\ldots n}$ be the d-dimensional feature vector of the points inside R. The sample covariance matrix for R is given by $C_R = \frac{1}{n-1}\sum_{k=1}^{n}(z_k - \mu)(z_k - \mu)^T$. The Riemannian structure of the space of covariance matrices is studied as the space of non-singular, symmetric positive definite matrices [82]. Let $\tilde{\mathcal{P}}(d)$ be the space of $d \times d$ SPD matrices and $\mathcal{P}(d) = \{P | P \in \tilde{\mathcal{P}}(d) \text{ and } \det(P) = 1\}$. The space $\mathcal{P}(d)$ is a well known symmetric Riemannian manifold, it is the quotient of the special linear group $SL(d) = \{G \in GL(d) | \det(G) = 1\}$ by its closed subgroup $SO(d)$ acting on the right and with an $SL(d)$ invariant metric [61]. Although several metrics have been proposed for this space, few qualify as Riemannian metrics, we use the metrics defined in [109] since the expression for parallel transport is readily available. The Lie algebra of $\mathcal{P}(d)$ is $\mathcal{T}_I(\mathcal{P}(d)) = \{A | A^T = A \text{ and } \text{trace}(A) = 0\}$, where I denotes

the $d \times d$ identity matrix and the inner product on $\mathcal{T}_I(\mathcal{P}(d))$ is $\langle A, B\rangle = \text{trace}(AB^T)$. The tangent space at $P \in \mathcal{P}(d)$ is $\mathcal{T}_P(\mathcal{P}(d)) = \{PA | A \in \mathcal{T}_I(\mathcal{P}(d))\}$ and $\langle PA, PB\rangle = \text{trace}(AB^T)$. The exponential map is given as $P \in \mathcal{P}(d)$ and $V \in \mathcal{T}_P(\mathcal{P}(d)), \exp_P(V) = \sqrt{Pe^{2(P^{-1})V}P}$. The inverse exponential map: For any $P_1, P_2 \in \mathcal{P}(d), \exp_{P_1}(P_2) = P_1 log\left(\sqrt{{P_1}^{-1}P_2^2{P_1}^{-1}}\right)$. Finally, for any $P_1, P_2 \in \mathcal{P}(d)$, the parallel transport of $V \in \mathcal{T}_P(\mathcal{P}(d))$ from $P_1 \to P_2$ is $P_2 T_{12}^T B T_{12}$, where $B = P_1^{-1}V, T_{12} = P_{12}^{-1}P_1^{-1}P_2$ and $P_{12} = \sqrt{P_1^{-1}P_2^2P_1^{-1}}$.

Chapter 3

SYMBOLIC APPROXIMATION FOR FAST SEARCH, COMPARISON, AND COMPRESSION

In this chapter we consider the problem of fast comparison of sequences of structured visual representations, which have non-Euclidean geometric properties. Examples of such structured representations include shapes [63, 106], optical flow [31], covariance matrices [120] where underlying distance metrics are highly involved and even simple statistical operations are usually iterative. Generally speaking, the ideal symbolic representation is expected to have two key properties: (1) be able to model the data accurately with a low approximation error, and (2) should enable the efficient use of existing data structures and algorithms, developed for string searching.

1 Related Work

Indexing static points on non-Euclidean spaces Not surprisingly, many standard approaches for sequence modeling and indexing which are designed for vector-spaces need significant generalization to enable application to non-Euclidean spaces. Indexing of static data on manifolds has been addressed recently with hashing based approaches [30]. For data points lying on the space of Symmetric Positive Definite (SPD) matrices, [51] present a dimensionality reduction technique that is geometry aware. Our interest lies in indexing sequences directly instead of individual points. Signal approximation for manifolds using wavelets [85] is a related technique. However, it is non-adaptive to the data and requires observing the entire signal before it can be approximated, while the proposed framework allows for easy real time implementation once the symbols are learned. Recent work also dealt with modeling human activity as a manifold valued random process [141] where the proposed techniques are theoretically and computationally involved due to the requirement

of second-order properties such as parallel transports. Another related line of work in recent years has been advances in Riemannian metrics for sequences on manifolds [106]. These approaches consider a sequence as an equivalent vector-field on the manifold. A distance function is imposed on such vector-fields in a square-root elastic framework. This is applied to the special case of curves in $2D$, nD, and non-Euclidean spaces [106, 60, 110]. While such a distance function could be utilized for the purposes of indexing and approximation of sequences, it is offset by the computational load required in computing the distance function for long sequences.

Computationally efficient representations of images and video In the past decade, there has been significant progress in efficient retrieval and indexing techniques [33] for very large image datasets. There have also been extensions to video retrieval [86] from very large databases. These techniques have made it possible to search accurately through large image and video data bases, but most methods are for high dimensional Euclidean points or time-series, and their generalization for manifold valued data is unclear.

Euclidean time-series indexing A successful approach to tackle the problem of fast indexing of *scalar* sequences has been to discretize and quantize the sequence in a way such that the obtained symbolic form contains most of the information of the original sequence, yet enabling much faster computations. This class of approaches are broadly termed as **S**ymbolic **A**ggregate Approx**i**mation (SAX) [71]. Several problems of indexing and motif discovery from time series have been addressed using this framework [71, 77], however the extension from $1D$ to multidimensional and non Euclidean spaces is not trivial. Multidimensional extensions to SAX have also been proposed such as [121], but these are trivial extensions which perform SAX on every dimension individually without considering the geometry of the ambient space.

Further, for manifolds such as the Grassmannian or the function-space of closed curves, there is no natural embedding into a vector space, thus motivating the need for a geometry-

based intrinsic approach [102, 106]. We show that this class of approaches can be generalized to take into account the geometry of the feature space resulting in several appealing characteristics, as they enable us to replace highly non-linear distance function computations with much faster and simpler symbolic distance computations.

Efficient string searching The biggest advantage of using the proposed indexing method is the the representation of complex feature types using abstract symbols, that are learned offline. This enables the use of string searching algorithms, allowing one to search through very high dimensional, non-linear spaces with a $O(m+n)$ complexity or better, where m and n are the length of a query, and the size of a activity database respectively. A known result in data mining is that the computational complexity can be further reduced to $O(m+n(log_{|\Sigma|}m)/m)$, for an alphabet of size Σ, when the symbols are independent and equiprobable [7]. Other lower bounds have been proposed when symbols are equiprobable [139], and it is known the height of suffix trees is optimized with equiprobable symbols [38]. The vector space SAX [71] proposed to generate symbols by partitioning the Gaussian distribution into bins of equal probability. However, it is not trivial to partition the data space into equiprobable regions on manifolds hence we use a conscience based competitive learning algorithm to learn the codebook.

2 Symbolic Approach For Manifold Sequences

In this section, we describe the proposed representation for manifold sequences which allows efficient algorithms to be deployed for a variety of tasks such as motif discovery, low-complexity activity recognition. We focus on the piece-wise aggregate and Symbolic approximation (PAA, SAX) [27, 71] formulation, and present an intrinsic method to extend it to non Euclidean spaces like manifolds. Briefly, the PAA and SAX formulation consist of the following principal ideas - A given 1D scalar time-series is first divided into windows and the sequence in each window is represented by its mean value. This process is referred to as **piece-wise aggregation**. Then, a set of 'break-points' is chosen which correspond to

dividing the range of the time-series into equi-probable bins. These break-points comprise the symbols using which we translate the time series into its symbolic form. For each window, the mean value is assigned to the closest symbol, this step is referred to as **symbolic approximation**. This representation has been shown to enable efficient solutions to scalar time-series indexing, retrieval, and analysis problems [71].

For manifolds, to enable us to exploit the advantages offered by the symbolic representation of sequences, we need solutions to the following main problems - a) piece-wise aggregation: which can be achieved by appropriate definitions of the mean of a windowed sequence on a manifold, and b) symbolic approximation: which requires choosing a set of points that are able to represent the data well. Here, we discuss how to generalize these concepts to manifolds.

2.1 Piece-wise Aggregation

Denote the manifold of interest by $\mathcal{M}$, given a sequence $\gamma(t) \in \mathcal{M}$, we define its piece-wise approximation in terms of local-averages in small time-windows. To do this, we first need a notion of a mean of points on a manifold. Given a set of points on a manifold, a commonly used definition of their mean is the Riemanian center of mass or the Fréchet mean [49], which is defined as the point μ that minimizes the sum of squared-distance to all other points:

$$\mu = \arg\min_{x \in \mathcal{M}} \sum_{i=1}^{N} d_{\mathcal{M}}(x, x_i)^2, \tag{3.1}$$

where $d_{\mathcal{M}}$ is the geodesic distance on the manifold.

Computing the mean is not usually possible in a closed form, and is unique only for points that are close together [49]. An iterative procedure is popularly used in estimation of means of points on manifolds [81]. Since in local time windows, points are not very far away from each other, the algorithm always converges. Thus, given a manifold-valued time series $\gamma(t)$, and a window of length W, we compute the mean of the points in the window and this gives rise to the piece-wise aggregate approximation for manifold sequences. When we consider vectors in $\mathbb{R}^n$, this reduces to finding the standard mean of W n-dimensional

vectors.

2.2 Symbolic Approximation

As discussed above, one of the key-steps in performing symbolic approximation for manifold-valued time-series is to obtain a set of discrete symbols. An established theoretical result within the data mining literature is that the efficiency of string searching is optimized when the elements of the codebook are equiprobable [7, 38]. The authors of SAX [71] emphasize on using equi-probable symbols because they achieve optimal results for fast searching and retrieval using suffix trees, hashing, and Markov models. However, standard clustering approaches do not necessarily result in equiprobable distributions of their centers [143, 65, 87]. It is also known that when symbols are not equiprobable, there is a possibility of inducing a probabilistic bias in the process [72]. We outline the methods to obtain symbols next.

Geometry Aware K-means For Learning Symbols

As a baseline, we chose K-means because it is the most widely used clustering approach and its extension to non Euclidean spaces is well understood. For a set of points $D = (U_1, U_2, \ldots, U_n)$ we seek to estimate clusters $(C) = (C_1, C_2, \ldots, C_K)$ with centers $(\mu_1, \mu_2, \ldots, \mu_K)$ such that the sum of geodesic-distance squares, $\Sigma_{i=1}^{K} \Sigma_{U_j \in C_i} d^2(U_j, \mu_i)$ is minimized. Here $d^2(U_j, \mu_i) = |\exp_{\mu_i}^{-1}(U_j)|^2$, where $\exp^{-1}$ is the inverse exponential map as described in section 2. We later show that one does not obtain equiprobable symbols using K-means.

Conscience Based Competitive Learning On Manifolds

To generate symbols or prototypes that divide the feature manifold into equiprobable regions, we extend ideas from Desieno's competitive learning mechanism [36] to make it adaptive to the geometry of the space and generate equiprobable symbols. It has been observed that a 'conscience' based competitive learning approach does result in symbols that are much more equiprobable than those obtained from clustering approaches. How-

ever, the algorithm described in [36] is devised only for vector-spaces. Here, we present a generalization of this approach to account for non-Euclidean geometries.

The conscience mechanism starts with a set of initial symbols/prototypes. When an input data-point is presented, a competition is held to determine the symbol closest in distance to the input point. Here, we use the geodesic distance on the manifold for this task. Let us denote the current set of K symbols as $\{S_1, S_2, \ldots, S_K\}$, where each $S_i \in \mathcal{M}$. Let the input data point be denoted as $X \in \mathcal{M}$. The output y_i associated with the i^{th} symbol is described as

$$\begin{aligned} y_i &= 1, \text{ if } d^2(S_i, X) \leq d^2(S_j, X), \forall j \neq i \\ y_i &= 0, \text{otherwise} \end{aligned} \tag{3.2}$$

where, $d()$ is the geodesic distance on the manifold. Since this version of competition does not keep track of the fraction of times each symbols wins, it is modified by means of a bias term to promote more equitable wins among the symbols. A bias b_i is introduced for each symbol based on the number of times it has won in the past. Let p_i denote the fraction of times symbol i wins the competition. This is updated after each competition as

$$p_i^{new} = p_i^{old} + B(y_i - p_i^{old}) \tag{3.3}$$

where $0 < B << 1$. The bias b_i for each symbol is computed as $b_i = C(\frac{1}{K} - p_i)$, where C is a scaling factor chosen to make the bias update significant enough to change the competition (see below). The modified competition is given by

$$\begin{aligned} z_i &= 1, \text{ if } d^2(S_i, X) - b_i \leq d^2(S_j, X) - b_j, \forall j \neq i \\ z_i &= 0, \text{otherwise.} \end{aligned} \tag{3.4}$$

Finally, the winning symbol is adjusted by moving it partially towards the input data point. The key extension of this algorithm from vector space to non Euclidean spaces lies

in this step. In the vector-space version this step is achieved by $S_i^{new} = S_i^{old} + \alpha((X) - S_i^{old})z_i$.The partial movement of a symbol towards a data-point can be achieved by means of the exponential and inverse-exponential map as

$$S_i^{new} = \exp_{S_i^{old}}[\alpha \exp^{-1}_{S_i^{old}}(X)z_i]. \tag{3.5}$$

The proposed algorithm for conscience based equi-probable symbol learning is summarized in algorithm 1.

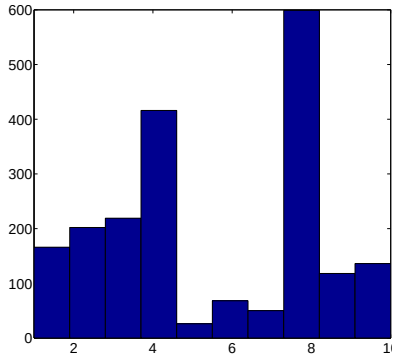
Algorithm 1 Equiprobable Symbol Generation on Manifolds.

Input: Dataset $\{X_1, \ldots, X_n\} \in \mathcal{M}$. Initial set of symbols $\{S_1, \ldots, S_k\}$.
Parameters: Biases $b_i = 0$, learning rate α, win update factor B, conscience factor C.
while $iter \leq maxiter$ **do**
  for $j = 1 \rightarrow n$ **do**
    $\tilde{i} \leftarrow \min_i d^2(X_j, S_i) - b_i$
    $z_{\tilde{i}} = 1,\ z_i = 0, i \neq \tilde{i}$
    $S_i \leftarrow \exp_{S_i}[\alpha \exp^{-1}_{S_i}(X_j)z_i]$
    $p_i \leftarrow p_i + B(z_i - p_i)$
    $b_i \leftarrow C(1/k - p_i)$
  end for
end while

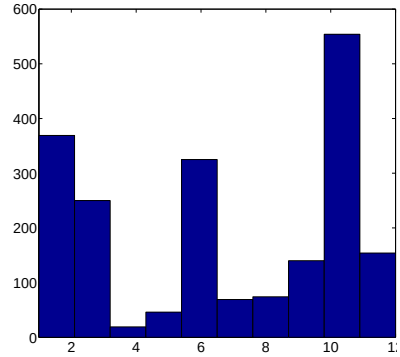
Algorithm 2 Symbolic Approximation for Feature Sequences in Euclidean & Non Euclidean Spaces.

Input: Feature sequence $\{\beta_1, \ldots, \beta_N\} \in \mathcal{M}$, Learned dictionary $\{D_1, \ldots, D_K\}$, Metric $d_{\mathcal{M}}$ defined on $\mathcal{M}$
Parameters: Size of aggregating window W ($<< N$),
Output: Symbolic approximation, $\mathbf{S}$.
$M \leftarrow \lceil \frac{N}{W} \rceil$.
$n = 1$
for $m = 1 \rightarrow M$ **do**
  $\mathrm{A}_m \leftarrow$ **intrinsic mean**$\{\beta_n, \beta_{n+1} \ldots \beta_{n+W-1}\}$
  $\mathbf{S}(m) \leftarrow \underset{1 \leq j \leq K}{\operatorname{argmin}}\ d_{\mathcal{M}}(A_m, D_j)$.
  $n = n + m \times W$
end for

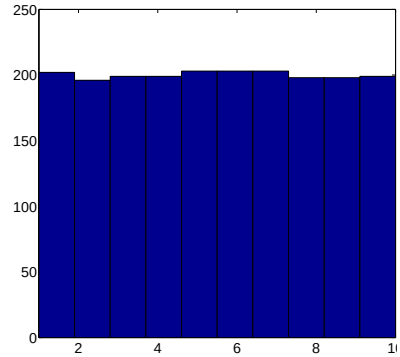
Next, we illustrate the strength of this approach in obtaining equiprobable symbols on manifolds. For this experiment we chose the UMD human activity dataset [123] and pre-processed it such that we obtain the outer contour of the human. A detailed discussion of



(a) K-Means



(b) Affinity Propagation



(c) Equiprobable Clustering

Figure 1: Probability density functions of the labels generated using (a) K-Means clustering, (b) Affinity Propagation and (c) Equi-Probable Clustering are shown, the feature space in this case was the Grassmann manifold as described in the text. As seen above, equiprobable clustering assigns all clusters with almost equal probability.

the dataset, processing, choice of shape metrics etc. appears in the experiments section. Here, we performed clustering of 2000 shapes from the dataset into 10 clusters. We show the histograms of the symbols in fig 1. As seen, both K-means and affinity propagation result in symbols that are far from equiprobable. The proposed approach results in symbols which

are much closer to a uniform distribution. The entropy defined as $-\sum_{i=1}^{N} p_i \log_2(p_i)$, is shown for three different datasets in fig 2. It is seen that the algorithm converges quickly in all cases. Once the symbols are obtained, transforming the feature sequence to its symbolic form is performed using algorithm 2.

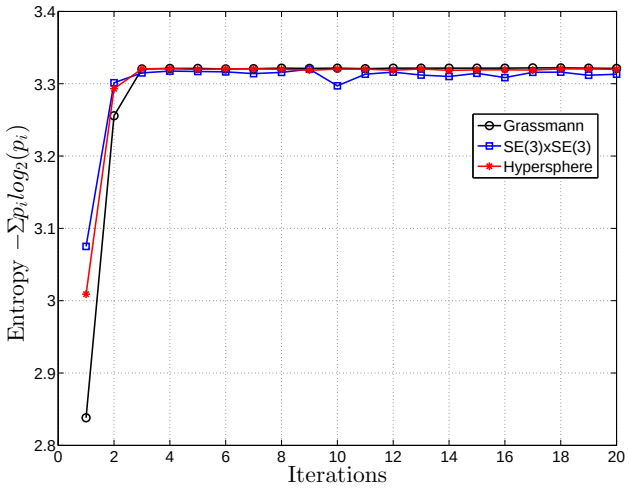


Figure 2: Convergence for the algorithm 1 on different feature manifolds to obtain 10 symbols - Grassmannian (UMD), Hypersphere (Weizmann) and $SE(3) \times .. \times SE(3)$(UTKinect). Entropy is plotted as a measure of equiprobability, higher the better.

In practice, while K-means minimizes approximation error it does not have the favorable property of equiprobability, and competitive learning gives us symbols which are equally likely, while compromising on approximation error. In order to find a trade-off between the two, we use a hybrid approach that first uses K-means and then competitive learning from which equiprobable symbols can be obtained in a two stage process. In the first stage we cluster the data using K-means into a small number of clusters, this ensures most data points are adequately represented. Each of these clusters is further split into smaller, equiprobable *sub-clusters* in the second stage using conscience learning. The number of clusters in the first stage is an empirical choice, we used values in the range of 5 to 10 for each data set. The number of sub-clusters in the second stage varies according to the probability of their parent cluster. For example, if p_s was the probability of the smallest cluster and we decide to split it into r smaller sub-clusters, then the i^{th} cluster with probability p_i would be split

into $\lceil \frac{p_i}{p_s} \times r \rceil$ clusters. The parameter r indirectly controls the size of the final set of symbols, we used values of r in the range of 1 to 5. We chose these values to obtain a codebook of size($\sim 40 - 50$). The training phase is expected to be computationally intensive, however this needs to be done only once and can be performed offline and does not affect the speed of comparisons during testing.

2.3 Limitations And Special Cases

Here, we discuss the limitations and some special cases of the proposed formulation. The overall approach assumes that a training set can be easily obtained from which we can extract the symbols for sequence approximation. In the 1D scalar case, this is not an issue, and one assumes that data distribution is a Gaussian, thus the choice of symbols can be obtained in closed-form without any training. If data is not Gaussian, a simple transformation/normalization of the data can be easily performed. In the manifold case, there is no simple generalization of this idea, and we are left with the option of finding symbols that are adapted for the given dataset. For the special case of $\mathcal{M} = \mathbb{R}^n$, the approach boils down to familiar notions of piece-wise aggregation and symbolic approximation with the additional advantage of obtaining data-adaptive symbols, this ensures that the proposed approach is applicable even to the vast class of traditional features used in video analysis. For the case of manifolds implicitly specified using samples, we suggest the following approach. One can obtain an embedding of the data into a Euclidean space and apply the special case of the algorithm for $\mathcal{M} = \mathbb{R}^n$. The requirement for the embedding here is to preserve geodesic distances between local pairs of points, since we are only interested in ensuring that data in small windows of time are mapped to points that are close together. Any standard dimensionality reduction approach [115, 91] can be used for this task. However, recent advances have resulted in algorithms for estimating exponential and inverse exponential maps numerically from sampled data points [73]. This would make the proposed approach directly applicable for such cases, without significant modifications. Thus the proposed formalism is applicable to manifolds with known geometries as well as to those whose geometry needs

to be estimated.

3 Speed Up In Sequence To Sequence Matching Using Symbols

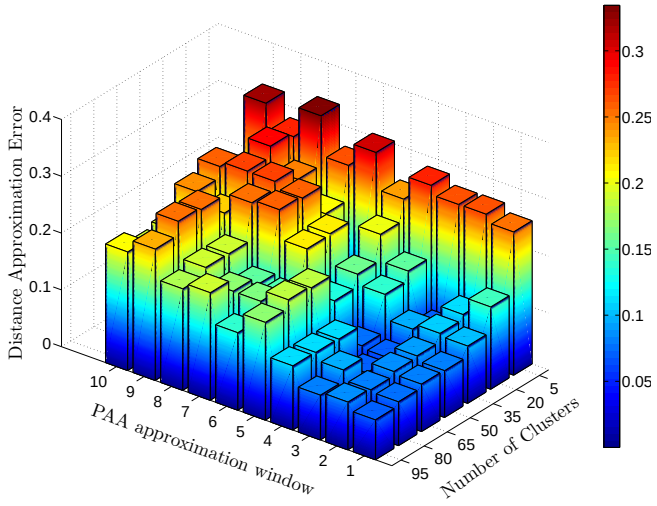


Figure 3: The trade-off between piece-wise aggregation and symbolic approximation is depicted here comparing the error in approximating the distance between two sequences from the Weizmann dataset. A symbol dictionary size of at least 40 and a approximation window size of up to 3 has negligible approximation error.

The applications considered in this dissertation are recognition and discovery of human activities. For recognition, a very commonly used approach involves storing labeled sequences for each activity, and performing recognition using a distance-based classifier, a nearest-neighbor classifier being the simplest one. When activity sequences involve manifold-valued time-series, distance computations are quite intensive depending on the choice of metrics. We explore here the utility of the symbolic approximation as an alternative way for approximate yet fast recognition of activities that can replace the expensive geodesic distance computations during testing. As we will show in the experiments, this is especially applicable in real-time deployments and in cases where recognition occurs remotely and there is a need to reduce the communication requirements between the sensor and the analysis engine. Before getting into the details of our experiments and distance metrics used, we define some of the terms used here:

1. *Activity* - We will consider an activity to be a high dimensional time series consisting of N data points such that each data point is a feature extracted per frame of the original video. The features can be either Euclidean or belong to abstract spaces such as Riemanian manifolds. We consider cases where all activities may not be of equal lengths by using DTW as a distance metric.

2. *Subsequence* - A subsequence is defined as a contiguous subset of the larger time series, i.e. for a time series $T = (t_1, t_2, \ldots, t_n)$ a subsequence of length n is $T_{i,n} = (t_i, t_{i+1}, \ldots, t_{i+n-1})$.

3. *Motif Discovery* - a pattern that repeats often within a larger time series is known as a motif. We say two patterns within the time series are similar if they are at a distance smaller than some threshold.

4. *Trivial Match* - Within a time series T, we say two subsequences P at position p and Q at position q are a trivial match if, $p \in (q - m + 1, \ldots, q, \ldots, q + m - 1)$ i.e p and q are different and within the neighborhood (as specified by m) of each other.

For an Activity of length N, we extract a symbolic representation in windows of size W (where typically $W << N$). To replace geodesic distance computations for recognition, we will consider subsequences in their symbolic representations to calculate the distance between activities. Let p_{sub} (eg: 'bccdea') and q_{sub} (eg: 'afffec') be two such subsequences of length l, then the distance metric d_{symbol}, defined on symbols, is:

$$d_{symbol}(p_{sub}, q_{sub}) = \sum_{i=1}^{l} d_{\mathcal{M}}\Big(D\big(p_{sub}(i)\big), D\big(q_{sub}(i)\big)\Big) \tag{3.6}$$

where $d_{\mathcal{M}}$ is the metric defined on the manifold, D is the set of symbols or dictionary that is previously learned and $D(a)$ is the point on the manifold corresponding to the symbol a. Here we assume that the two sequences are of the same length, in other cases we use DTW as a metric or learn a dynamical model for each sequence and use the distance between them as a metric. Since the symbols are known apriori, the distance between them can

be computed offline as part of training and stored as a look-up table of pairwise distances between symbols. This allows us to compute distances between sequences in near constant time, which is much faster than computing distances each time using DTW on actual features.

Before considering applications for the simplified distance measure, one must consider the trade-off between piecewise aggregation, number of symbols versus the error of approximation, this is shown in figure 3.

For activity discovery, we consider the problem as one of mining for motifs in time-series. In finding motifs, it is important to consider only non-trivial matches, for every such match we store its location and find the top k motifs. For each of the k motifs, we define a *center* for the motif as the sequence which is at minimum distance to all the sequences similar to it. These centers are the k most recurring patterns in the multidimensional time series. We use the brute-force algorithm given in [80] to extract our motifs.

4 Experimental Evaluation

In this section, we demonstrate the utility of the proposed algorithms for symbolic approximation and its application to activity recognition and discovery. We also study the complexity advantage in using these symbols as compared to original feature sequences. We first describe the datasets and choice of features.

UTKinect dataset [136] contains 10 activities by 10 subjects, where each activity is repeated twice. There are a total of 199 action sequences. Here we use the feature proposed recently in [125], which models each skeleton as a point on the cross product space of $SE(3) \times \cdots \times SE(3)$.

The UMD database consists of 10 different activities like bend, jog, push, squat etc.[124], each activity was repeated 10 times, so there were a total of 100 sequences in the dataset. The background within the UMD Dataset is relatively static which allows us to perform background subtraction. From the extracted foreground, we perform morphological operations and extract the outer contour of the human. We sampled a fixed number of points

Figure 4: Sample images from the various data sets used for validation. The UTKinect [136], UMD [124], the Wiezmann [48], and the UCSD traffic [29] data sets are shown here from top to bottom in that order.

on the outer contour of the silhouette to yield landmarks, which are represented as points on the Grassmann manifold.

The Weizmann Dataset consists of 93 videos of 10 different actions each performed by 9 different persons [48]. The classes of actions include running, jumping, walking, side walking etc. Here, the HOOF features [31] are represented as points on a hyper-spherical manifold.

The UCSD traffic database consists of 254 video sequences of daytime highway traffic in Seattle in three patterns i.e. heavy, medium and light traffic [29]. It was collected from a single stationary traffic camera over two days.

4.1 Speed Up And Compression Achieved Using Symbols

A theoretical complexity analysis of the algorithm is shown in table 1. We also consider three metrics to study the time-complexity of the proposed framework. Namely 1)

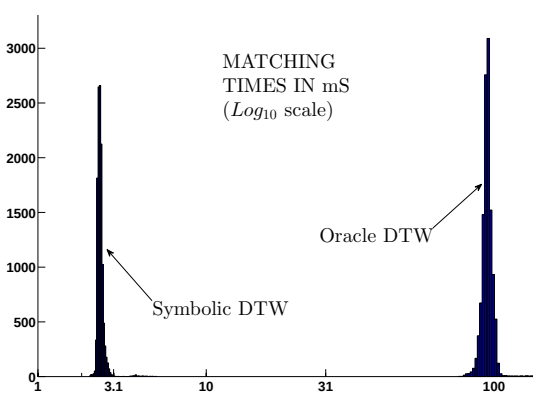
Step	Complexity
Exponential map for $\mathcal{M}$ (manifold specific)	O(ν)
Inverse exponential map for $\mathcal{M}$ (manifold specific)	O(χ)
Intrinsic K-means clustering	O((NK + K^2)χ+Kν) Γ)
Equi-probable clustering	O((NKχ+Nν)Γ)
Approximation of N-length activity to M symbols	O(M(wχ+ν)Γ+MKχ)
Symbolic DTW	O($M^2\delta$), δ is the look up time
Geodesic distance DTW	O($M^2\chi$), $\chi >>> \delta$

Table 1: Theoretical complexity analysis for the proposed algorithms. Notations used: N - number of data points, K - number of symbols, with O(δ) the time required to read from memory, Γ maximum number of iterations, M and w are as defined in algorithm 2 and are usually much lesser than N. It can be seen that a huge complexity gain is achieved in using symbols over original features.

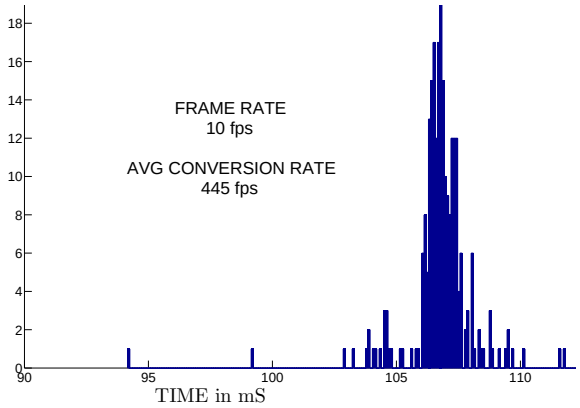
Time complexity of matching using symbols vs original feature sequences, 2) Time required to transform a given activity into a symbolic form, and 3) Number of bits required to store/transmit symbols as compared to feature sequences. Ideally, we require that the matching time be several orders of magnitude faster than using the original sequences, the transformation time to be small enough to enable real-time approximation, and very small bit-rate/storage requirement compared to original feature sequences. We show in the following that the proposed framework successfully satisfies all these criteria. We performed the experiments using MATLAB, on a PC with an i7 processor operating at 3.40Ghz with 16GB memory on Windows 7.

KNN Search And Sequence Matching Time Analysis

In this experiment we show the gain in speed and compression achieved using symbols compared to using the original high-dimensional features with accompanying metrics. For the gain in speed, we measured the run-time of matching sequences using DTW on symbols vs geodesic DTW. As shown in fig 5a, the time taken to match two activity sequences using symbols is just 3.1ms which is two orders of magnitude faster than 100ms that it



(a) Matching times



(b) Converting features to symbols

Manifold	manifold SAX	Geodesic
Grassmann	0.0082 ± 0.0002s	1.54 ± 0.10s
Sphere	0.24 ± 0.08s	1.22 ± 0.39s
$SE(3) \times SE(3)$..	0.50 ± 0.04s	57.80 ± 1.58s

(c) kNN searching times

Feature	Original Data	manifold SAX	Compression
Shape	500 Kb	0.468 Kb	**99.90**%
HOOF feature	65.625 Kb	0.410 Kb	**99.37**%
Skeleton feature	4,617 Kb	111.67 Kb	**97.58**%

(d) Bit budgets for original vs manifold SAX

Figure 5: Comparison of histograms for matching times when using symbolic v/s original feature sequences are shown in fig 5a for the UCSD traffic dataset. The times are shown in milliseconds on a log scale. As it can be seen, using symbols speeds up the process by nearly two orders of magnitude. Fig 5b shows a histogram of times taken to translate entire activities of 50 frames into symbols from the UCSD dataset. Table 5c shows the improvements in performing a k-NN search on different feature manifolds. Finally table 5d shows the reduced storage requirements for different features.

takes using the actual features. Next, we compare the times taken to perform a k-nearest neighbor (kNN) search on different manifolds in table 5c. Similar to the sequence matching speed, the search speed is improved by nearly two orders of magnitude.

Analysis Of Approximation Time

Fig 5b shows the distribution of times taken over various activities to transform them into their respective symbolic forms. The average conversion time for an entire activity video is about 107ms. In other words, we can process the video at a speed of 445 frames per second (fps) which allows for easy real time implementation since most videos are recorded at 10-30fps.

Bit-rate Analysis

Next, to demonstrate the gain in compression we compared our representation to a baseline using the original feature sequence. Assuming each dimension of the feature is coded as a 32-bit float number, we calculated the bits it would take to represent each feature and its symbolic representation. As shown in table 5d, on nearly all the feature types, the compression ratios are 97% or higher. For a dictionary of size K, the number of bits required to represent each symbol is $Log_2(K)$. This provides enough flexibility for the user to choose the size of the codebook and pick features of their choice without significantly affecting the bit-rate.

4.2 Activity Discovery Experiment

Having learned the symbols, we test their effectiveness in activity discovery. For this experiment, we randomly concatenated 10 repetitions of 5 different activities of the UMD dataset to create a sequence that was 50 activities long. Each activity consists of 80 frames which were sampled by a sliding window of size 20 frames with step size of 10 frames. After symbolic approximation, this resulted in 6 symbols per activity, chosen from an alphabet of 25 symbols. The *motifs* or repeating patterns, in five activities - *Jogging, Squatting,*

Bending Knees, Waving and Throwing were discovered automatically using the proposed method. Each of the discovered motifs was validated manually to obtain a confusion matrix shown in table 2. As can be seen, it shows a strong diagonal structure, which indicates that the algorithm works fairly well. Even though all executions of the same activity are not found, we do not find any false matches either.

Activity Type	1	2	3	4	5
1	7	0	0	0	0
2	0	7	0	0	0
3	0	0	8	0	0
4	0	0	0	9	0
5	0	0	0	0	8

Table 2: Confusion matrix for the discovered motifs on the UMD database using the manifold SAX representation of the shape feature. Due to the symbolic representation, search can be performed very quickly. Actions discovered are - *jogging, squatting, bending, waving and throwing* respectively.

4.3 Activity Recognition Using Symbols

Symbolic approximation plays a significant role in reducing computational complexity since it allows us to work with symbols instead of working with high dimensional feature sets. In this experiment, we test the utility of the proposed symbolic approximation method for fast and approximate recognition of activities over three datasets. For each data set picking the number of symbols, K is an empirical choice, typically we picked $K = K_{min}$ where, for all $K > K_{min}$ the recognition performance shows no improvement. We also picked a window size of $W = 1$ in our recognition experiments to achieve best performance. A detailed comparison between the window size, number of symbols and performance is seen in figure 3, which shows the the error in the geodesic distance vs symbolic distance. To effectively demonstrate the quality of the approximation, we use the classifiers that were reported in the papers that proposed the features. For example, for the shape and the HOOF features, we use the nearest neighbor classifiers, and for the LARP features, we use the SVM. As a baseline, we compare the recognition accuracy of principal geodesic analysis

(PGA) [44], for diffenrent manifolds.

Activity	Accuracy (%)	Relative bit budget
Shape + manifold SAX	98	1
Shape + PGA [44]	90	6.012
Shape [124]	100	1202.6

Table 3: Recognition experiment for the UMD database with a shape silhouette feature. Here we see the performance achieved with symbolic approximation compared to an oracle geodesic distance based nearest neighbor classifier.

For the UMD dataset, we learned a dictionary of 60 symbols using algorithm 1. Then, we performed a recognition experiment using a leave one-execution-out test in which we trained on 9 executions and tested on the remaining execution, the results are shown in Table 3. It can be seen that the recognition performance using symbols is very close to that obtained by using an oracle geodesic distance DTW based algorithm. We achieve this performance with matching times that are significantly faster, as will be described in section 4.1.

For the UTKinect dataset, we learn a codebook of size 20 symbols for all the relative joints from actions corresponding to the training subjects. The approximated LARP features are then mapped to their corresponding Lie algebra and classified using a one-vs-all SVM classifier following the protocol of [125]. Here, our results are reported without any post-processing using Fourier Temporal Pyramids (FTP) as done in [125], which improves performance further by providing robustness to noise. Results show that even with a small codebook, the approximated features perform extremely well in action recognition, while drastically reducing the search speed 5c by a factor of nearly 50. Even though we approximate the actions, we obtain a better recognition performance than the original features which is explained by the fact that the Lie algebra, $\mathfrak{se}(3) \in \mathbb{R}^6$, which is much lower than the other features considered here and therefore can be appropximated much better with fewer symbols. The approximated LARP features also provide robustness to noise, which is common in features extracted using Kinect.

For the Weizmann dataset, we demonstrate the flexibility of the approximation strategy

Feature	Accuracy(%)	Relative bit budget
LARP+ manifold SAX	**94.77**	1
LARP+PGA [44]	92.46	20.428
LARP [125]	92.97	40.856
HOJ3D [136]	90.92	NA

Table 4: Results on the UTKinect Dataset.

by learning linear dynamical models over the approximated sequences, which also serves as a fair comparison to the state of the art techniques. We performed the recognition experiment on all the 9 subjects performing 10 activities each with a total of 90 activities. The dictionary learned had 55 symbols which were used to map the activities to the approximated sequences. Next, we fit a linear dynamical model to the approximately reconstructed actions and perform recognition with a nearest neighbor classifier using the Martin metric on LDS parameters [101]. The results for the leave-one-execution-out recognition test are shown in Table 5 and it can be seen there is almost no loss in performance in comparison to state of the art techniques. Better results have been reported on this dataset by Gorelick *et al.* [48] etc., but there are no common grounds between their technique or feature and ours for it to be a fair comparison.

Feature	Accuracy(%)	Relative bit budget
LDS+ manifold SAX	**92.22**	1
HOOF+DTW+manifold SAX	88.87	1
HOOF+DTW+PGA [44]	74.44	10.67
HOOF+DTW [31]	90.00	160
χ^2-Kernel [31]	95.66	160
Chaotic measures [6]	92.60	NA

Table 5: Recognition Performance for the Weizmann Dataset.

Finally we show that the proposed framework can be used easily with Euclidean features on the Traffic Database. We stack every other pixel in the rows and columns of each frame to form our feature vector. We learned 45 symbols from the training set using these features.

	Manifold SAX (%)	CS LDS(%)	Oracle LDS(%)
Expt 1	84.13	85.71	77.77
Expt 2	82.81	73.43	82.81
Expt 3	79.69	78.10	91.18
Expt 4	79.37	76.10	80.95
Average	**81.50**	78.33	83.25

Table 6: Recognition performance for UCSD traffic data set. The results for Oracle LDS and CS LDS are from [93].

We performed the recognition experiment on 4 different test sets which contained 25% of the total videos. We used a 1-NN classifier with a DTW metric on the symbols. The results are shown in Table 6. We compare our results to [93], which also performed recognition using lower dimensional feature representation using compressive sensing. As it can be seen, recognition performance is clearly better when the feature is in its symbolic form as compared to when it was compressively sensed, given that both are significantly reduced versions of the original feature. We also perform nearly as well as the performance achieved using the original feature itself.

5 Discussion And Future Work

In this chapter we presented a formalization of high dimensional time-series approximation for efficient and low-complexity activity discovery and activity recognition. We presented geometry and data adaptive strategies for symbolic approximation, which enables these techniques for new classes of non-Euclidean visual representations, for instance in activity analysis. The results show that it is possible to significantly reduce Riemannian computations during run-time by an intrinsic indexing and approximation algorithm which allows for easy and efficient real time implementation. This opens several avenues for future work like an integrated approach of temporal segmentation of human activities and symbolic approximation. A theoretical and empirical analysis of the advantages of the proposed formalism on resource-constrained systems such as robotic platforms would be

another avenue of research.

Finally, the framework in this paper is general enough to deal with more abstract forms of information such as graphs [59] or bag-of-words [45]. In fact, any system that is sequential can be used within this framework, the key is to have a good understanding of metrics on these abstract models. Existing works have defined kernels for data on manifolds [66], for graphs [128] and a good starting point would be to use these to develop a kernel version of this framework that would allow us to learn symbols.

Chapter 4

COMPETITIVE LEARNING FOR DIVERSE SAMPLING

In the previous chapter, we noted how introducing a 'conscience' bias into the competitive learning framework can influence the algorithm to pick samples that can divide the feature space into equally likely regions. The choice of a bias can give rise to interesting sampling mechanisms. In this chapter we will look at one such sampling algorithm that is obtained by introducing a *diversity* bias. Smart sampling algorithms are useful in applications where computational or memory resources are limited. In such scenarios, a small number of well chosen samples can be used to generalize properties of an entire dataset for training [96], labeling [12], or other learning problems [98, 138]. We are interested in video summarization, which can be broadly defined as the problem of picking the K best frames/shots/segments of a video. The challenge in summarizing a video is to define an appropriate cost function, since it can be very subjective based on the application. Almost all video summarization algorithms today work after the fact, i.e. they assume access to the entire video at a time. However, there are many emerging applications with high definition streaming video, where there is a need to perform summarization with little or no memory overhead such as videos on mobile platforms etc. In this work we propose a online generalization of the video summarization problem so that it can work while accessing a single frame at a time, as shown in figure 1. We formulate summarization as a diverse sampling problem, which picks the most *diverse* set of samples from a dataset. This approach is inspired by Video Precis [98], a batch-mode algorithm, that modifies the K-means clustering cost to include the *diversity* of centers in addition to the standard ℓ_2 clustering error. The additional diversity term improves sampling by making the algorithm less sensitive to large and dense clusters, unlike K-means. In the context of summarization, this results in a summary that samples from all key events. An effective video summarization algorithm trades-off between

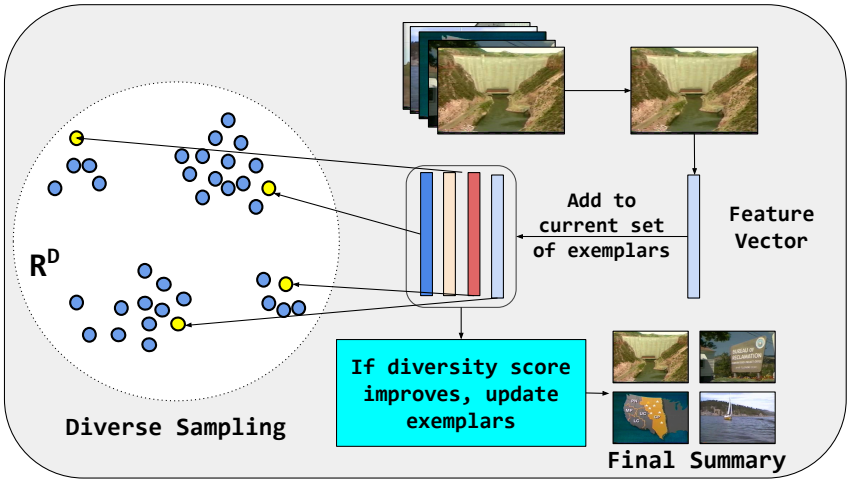


Figure 1: **Overview of our system** for online video summarization.

representing most of the video and picking unique and/or interesting frames that may occur sparsely. Our algorithm has memory requirements in the order of $\mathcal{O}(K)$, where K is the length of the summary, typically in the range of 10-100. This is much better than existing approaches, which require *at least* $\mathcal{O}(N)$, the computational complexity is also linear in N, compared to quadratic complexity for comparable approaches.

Existing approaches for batch-mode summarization have used different strategies to define importance scores for events in a video. For example, the work in [46] focuses on ego-centric video and uses visual cues that humans often use such as the position of the object within the frame. As a result, any object in the focus of the user is given high importance. The idea of important objects from a single view point, has also been generalized for generic videos [64]. In many videos, there is a lot of content in video transitions, which can be omitted using priors learned from the web [64]. Adaptive or dynamic video summarization does not enforce a fixed summary length and adapts the length of the summary based on the information within the video [28]. Online summarization for videos has remained largely unexplored – the work in [8] proposes to use a user-customizable summarization which allows the user to specify quality of the summary and also the time available for the process. This technique enhances the user experience and speeds up the process by creating

the summary as an online task, saving time. In contrast, we propose an online algorithm that can work with any kind of image/video features, while having access to a single frame at a time. We propose a generalization to the online K-means clustering algorithm, that also includes a *diversity bias*. This ensures that each sample is assigned to a center that is close to it while also satisfying the diversity constraint. In a special case, our algorithm reduces to the online K-medoids clustering algorithm. We show that the proposed algorithm is able to summarize videos significantly better than several comparable baselines, at significantly lesser computational cost. We show extensive evaluation on a dataset of 50 videos [1, 35] and perform a comparison with human-user generated summaries.

1 Problem Formulation

The summarization problem can be stated as follows: given a set of frames from a video $X = \{x_1, x_2, \ldots, x_n\}, x_i \in \mathbb{R}^D$, pick the most *representative* K points, $\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots, \mu_K\}$ from the set. We will refer to these representatives as exemplars. The x_i's can be a feature or a set of features extracted per frame from the video. Summarization or diverse sampling is similar to clustering in many ways, and the clustering analogy is useful to illustrate our algorithm. For example, K-means (or K medoids) is a sampling algorithm when the centers are the samples, chosen by minimizing the ℓ_2 clustering error. In online K-means, a *competition* is held between centers to determine who 'wins' the current sample, determined by which center is the closest to the current sample in the sense of the Euclidean norm. The winning center is moved in the direction of the sample, by a small amount governed by the learning rate, $\alpha \in [0, 1]$. That is, for a winning center μ_k and the i^{th} point x_i, the updated center is given by $\widehat{\mu}_k = \mu_k + \alpha\ (x_i - \mu_k)$.

However K-means can be very biased towards larger clusters, leading to poor summaries. To overcome this, we modify the clustering error term to include a notion of diversity bias which forces the centers apart, instead of having several centers in a single large cluster. The diversity bias is similar to the conscience bias [36] that can be used to generate equi-probable clusters, where the bias discourages a center from winning too often. Instead, the

diversity bias promotes updating centers that improve the overall diversity. The modified cost function resembles the one used in Video Precis [98] for batch-mode summarization. In our algorithm the criterion to determine the winning center for the i^{th} round is given by $\hat{k} = \arg\min_k d(k)$, where $d(k)$ is given by:

$$d(k) = \beta \, ||x_i - \mu_k||^2 + C(1-\beta) \, \texttt{divscore}(\boldsymbol{\mu}_{k \leftarrow i}) - \zeta, \tag{4.1}$$

where $(\boldsymbol{\mu}_{k \leftarrow i})$ denotes the set of centers, when the k^{th} center is replaced by the current data point x_i, C is a normalizing factor that ensures all data points are given the same importance, and ζ is the previous maximum diversity score computed using the function – `divscore( )`.

1.1 Diversity Measure

The choice of the function `divscore( )`, in equation (4.1) is important since it significantly influences the final summary. Diversity can be measured using dispersion measures such as the sample variance of the centers, as in [98]. However, we observed that it can encourage a grouping behaviour, where a pair of centers is close to each other but far away from the rest of the centers.

Volume of the convex hull: We propose to use the volume of the convex polytope formed by the centroids, as our diversity score. A convex polytope P is the convex hull $conv(\boldsymbol{\mu})$ for a finite set of centers. Computing the volume is hard in general and computationally expensive [24], especially when the points are in higher dimensions [16]. Fortunately in lower dimensions its time efficient, and there are several standard implementations. We use the `qHull, convexhulln` functions in MATLAB [16]. For high dimensional features, we map the centers to $\mathbb{R}^d, d << D$. and then compute the volume of the convex-hull in $\mathbb{R}^d$. Although this may not reflect the true volume, it is an approximation that works well in practice.

Algorithm 3 describes the procedure to generate diverse samples in an online fashion. We initialize the exemplars with the first K data points. Following this, we compute the diversity score for the current set of exemplars, denoted as `divscore`($\boldsymbol{\mu}$) in algorithm

Algorithm 3 Online Diverse Sampling

Input: Currrent frame $x_i \in \mathbb{R}^D$, Number of exemplars K
Output: Exemplars $\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots, \mu_K\}$
if $i < K$ **then**
  $\boldsymbol{\mu}(i) = x_i$ // Initialization
  exemplar_idx$(i) = i$
else
  $C = 10\ i$ //normalizing factor
  $\zeta =$ `divscore`$(\boldsymbol{\mu})$) (see sec 1.1)
  for $k \leftarrow [1 \ldots K]$ **do**
    $div(k) =$ `divscore`$(\boldsymbol{\mu}_{k \leftarrow i}) - \zeta$
    $\mathrm{d}(k) = \beta \ \|x_i - \mu_k\|_2 - \frac{1-\beta}{C} \ (div(k))$
  end for
  $j = \arg\min_k d(k)$
  if $div(j) > \zeta$ **then**
    exemplar_idx$(j) = i, \boldsymbol{\mu}(j) = x_i$ //update
    $\zeta = div(idx)$
    divcost$(i) = \zeta$
  end if
end if

3. Next, we begin the competition to find out which center has won the current round. Here winning is determined by a modified cost function that includes a diversity cost. The importance given to clustering error versus the diversity cost is governed by β, which is a user defined parameter. When $\beta = 1$, this expression reduces to the cost used in the online K-means algorithm. The effect of β is shown in figure 2, the right choice of β can vary depending on the dataset and the features. Finally, we update the winning center only if it improves the overall diversity compared to the previous set. In some cases the centers may get stuck in local minimas, which can lead to poor exemplars. To avoid such cases, we add some noise, by updating centers even when they do not meet the diversity criterion in $1 - 10\%$ of the samples.

Complexity: One of the main advantages of an online algorithm is that it can function with very low memory and computational resources. For the task of picking K exemplars from dataset of N points, our algorithm requires $\mathcal{O}(K)$ for storage, compared to *at least* $\mathcal{O}(N)$ for batch-mode summarization algorithms such as Precis [98]. Typically, N can be of the order of 10^5 frames for an hour long video, whereas K is typically around $10 - 50$. In terms of computational complexity, our algorithm takes $\mathcal{O}(NK)$ as compared to Precis [98],

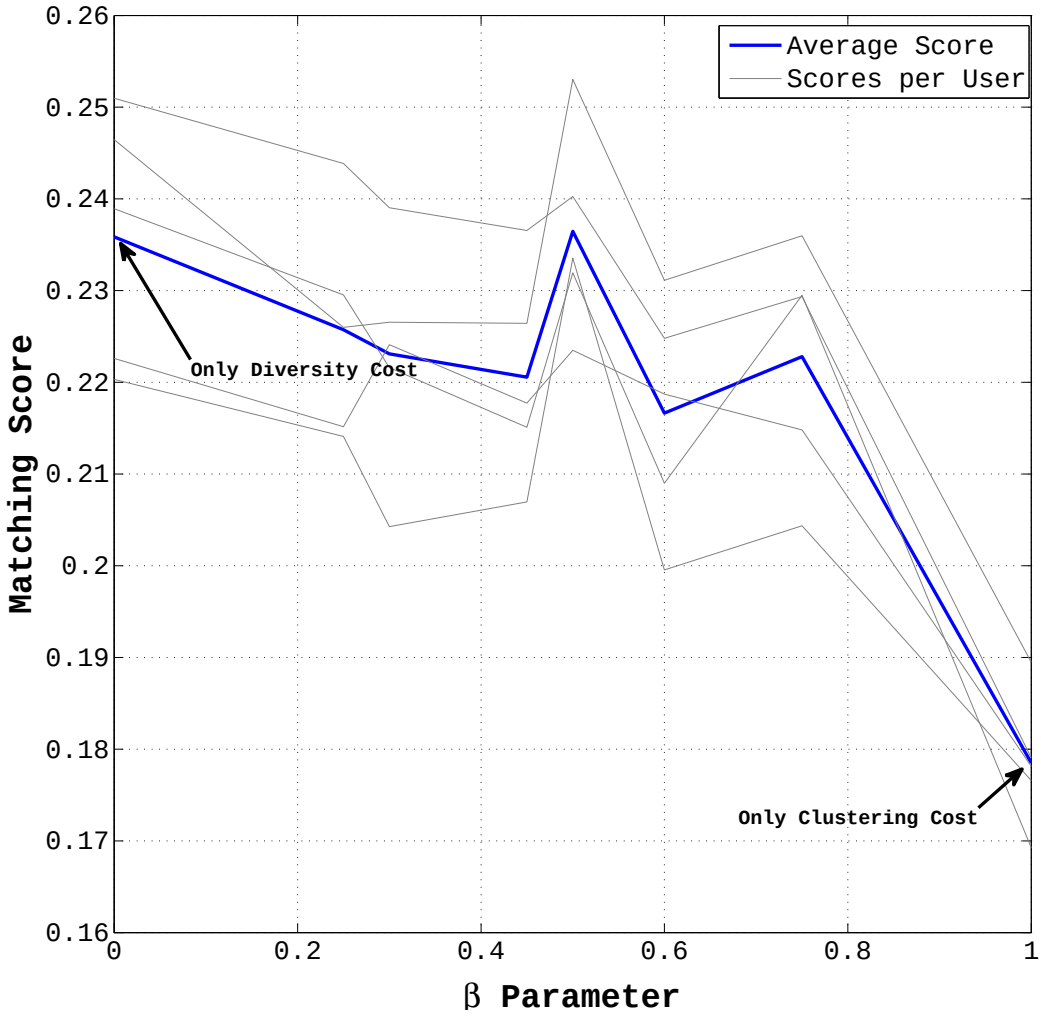


Figure 2: **Effect of the β parameter:** on summarization performance. It is interesting to note that when we make $\beta = 1$, there is a significant drop in the score since diversity is not considered at all. See algorithm 3 for more details. Here results for 5 different users at different βs are shown. The average is also depicted in bold.

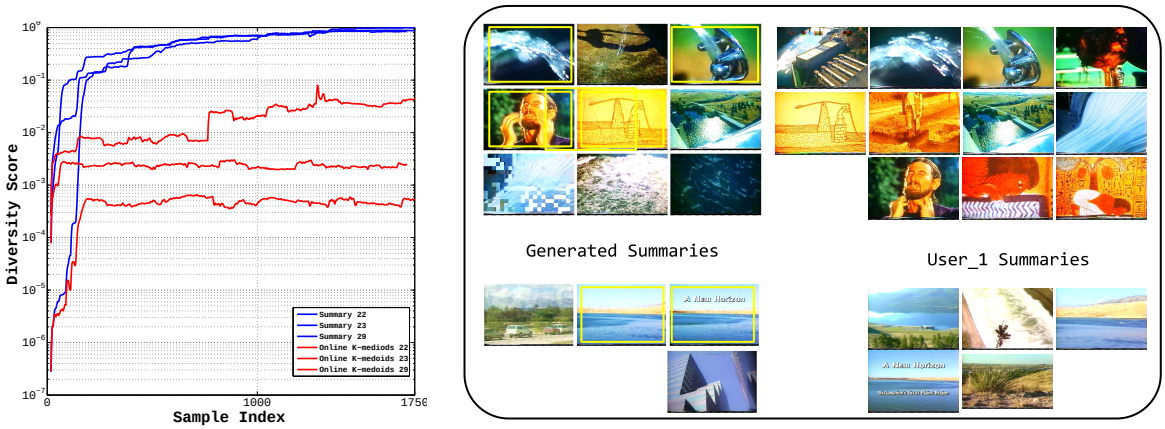
$\mathcal{O}(N(N-K)T)$ for T iterations. When $N >> K$, which is typical in summarization, the computational complexity of our algorithm approximates to $\mathcal{O}(N)$ while Precis increases to $\mathcal{O}(N^2T)$. As a result, we are able to process features extracted from a video at about 14.3 fps, in MATLAB on a standard Intel i7 PC.

2 Experiments

We perform experiments on the VSUMM dataset [35], which contains 50 videos in MPEG-1 format (30 fps, 352 x 240 pixels), distributed across several genres (documentary,

educational, ephemeral, historical, lecture) and their duration varies from 1 to 4 minutes and approximately 75 minutes of video in total [1]. The dataset also contains 5 different user evaluations per video, which are what human users have considered the best summary for the video. In order to exaggerate the advantage of using summarization over traditional sampling, we skew the dataset by replacing the last 500 frames of the video with a single *frozen* frame. Such artifacts can be expected to occur, but more importantly they demonstrate the effectiveness of summarization.

Feature Extraction: The video summarization problem is to pick the K best exemplars



(a) Diversity score for online K-medoids(b) Sample summaries generated for two different videos, the matches and the proposed algorithm, over 3 dif-are marked in yellow.

ferent videos. It is evident that our algorithm promotes diversity between exemplars much better than K-medoids.

from a set of N points, $X = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^D$. The choice of x_i is open to the application and the proposed algorithm can work with any kind of Euclidean features. We used deep features from the penultimate layer of a pre-trained neural network – the VGG "very deep" network [100] trained on the ImageNet dataset [92]. These pre-trained networks are available through the MatConvNet toolbox [122].

Defining a match: In order to accurately obtain the match score, we first filter the exemplars to remove similar frames. This is done by computing the $K \times K$ similarity matrix

a set of exemplars, followed by picking only those points that have a distance greater than a fixed threshold, γ. The value for γ needs to be chosen heuristically, and depends on the feature space. In our experiments with the deep features, we found that $\gamma = 70$, worked effectively in removing redundant exemplars. A weakness of using a fixed γ is it may result in false positives and false negatives and better schemes maybe used to choose γ. To make a fair comparison, we use the same value of γ across all our baselines.

Evaluation: Evaluating a summary is hard in general because there is no *ground truth*. In many cases, the evaluation is done in comparison to human user generated summaries to find the highest "matching" score. In VSUMM [35], a new evaluation metric is proposed that measures the number of matching frames, and the number of non matching frames. The user generated summaries are of arbitrary lengths, as deemed suitable by the user. However, since our algorithm requires K, the number of desired exemplars as an input we modify the evaluation score to simply be the number of matches between each user generated summary and the summary generated by our algorithm. We choose K to be equal the length of the largest summary set generated by a user per video, if $K < 5$, then we set $K := 2 * K$. This can be easily automated and chosen to be relative to the size of each video without affecting the results. Finally, we normalize the number of matches by the length of that user's summary.

2.1 Alternative Sampling Strategies And Results

As a comparison to the proposed approach, we perform sampling using the following different baselines.

Batch-mode Video Precis: [98] Our main comparison is with the Video Precis algorithm that optimizes between the representational error of the chosen samples and the diversity cost between any set of samples. The proposed algorithm can be considered an online version of Precis.

Online K-medoids clustering: We use the competitive learning algorithm used for on-

line K-means (see 1), as another comparison with comparable computational and memory complexity. Here, we set $\alpha = 1$, which is expected to be noisy since the learning rate is high. An alternative formulation could involve computing centers using a smaller α, then assigning each center to the nearest data point. However, this violates the assumption of an online algorithm that does not have access to the entire dataset.

In addition we also report results using batch-mode K-medoids, random sampling and uniform sampling. Random and uniform sampling require knowledge of the number of frames or length of a video, which is unrealistic for streaming video. The performance of different sampling algorithms are reported in table 1, and it can be seen that the proposed diversity sampling performs better than batch mode summarization algorithm Precis. We are also significantly better than the online K-medoids algorithm and other baselines. Sample summaries are shown in figure 3b, and the diversity score for our algorithm is compared to the diversity score obtained by the online K-medoids algorithm in figure 3a.

Sampling Algorithm	U1	U2	U3	U4	U5	Online?
K-medoids	0.191	0.199	0.179	0.199	0.193	✗
Random	0.173	0.165	0.176	0.186	0.179	✗
Uniform	0.190	0.196	0.188	0.200	0.193	✗
Precis [98]	0.227	0.219	0.225	0.240	**0.245**	✗
Online K-medoids	0.141	0.129	0.131	0.146	0.143	✓
Proposed	**0.240**	**0.224**	**0.234**	**0.253**	0.232	✓

Table 1: Average mean scores denoting the percentage match with 5 different users across 50 videos. The proposed online sampling scheme performs as well if not better than batch-mode Precis, and significantly outperforms comparable baselines.

3 Conclusion & Future Work

We presented the a novel online algorithm to perform streaming video summarization which can work with access to just a single frame at a time and does not need to know in advance the number of frames to allocate memory. We showed that the proposed online diverse sampling algorithm performs summarization as well as its batch-mode counter-parts,

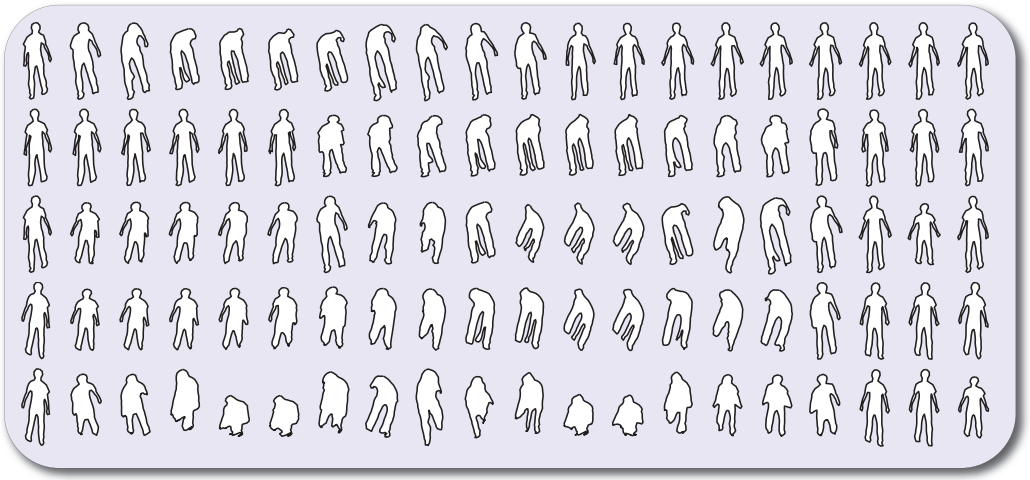
while being significantly more efficient. By generalizing aspects of competitive learning[36], and Video Precis [98], we are able to force the exemplars to be as *diverse* as possible. We used PCA to map the centers to a lower dimensional space and then measured the volume of the convex hull in the PCA space as a measure of diversity. In the future, the dimensionality reduction step can be replaced with more advanced tools, that preserve topological properties and can potentially improve the robustness of the diversity measure. Another interesting extension is to generalize this algorithm to non Euclidean spaces such as Riemannian manifolds.

Chapter 5

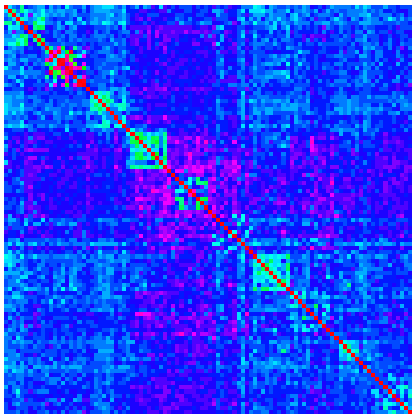
ELASTIC FUNCTIONAL CODES FOR REPRESENTATION AND RECOGNITION

There have been significant advances in understanding differential geometric properties of image and video features in vision and robotics. Examples include activity recognition [118, 31, 125], medical image analysis [44], and shape analysis [105]. Some of the popular non-Euclidean features used for activity analysis include shape silhouettes on the Kendall's shape space [124], pairwise transformations of skeletal joints on $SE(3) \times SE(3) \cdots \times SE(3)$ [125], representing the parameters of a linear dynamical system as points on the Grassmann manifold [118], and histogram of oriented optical flow (HOOF) on a hyper-sphere [31]. A commonly occurring theme in many applications is the need to *represent, compare, and manipulate* such representations in a manner that respects certain constraints.

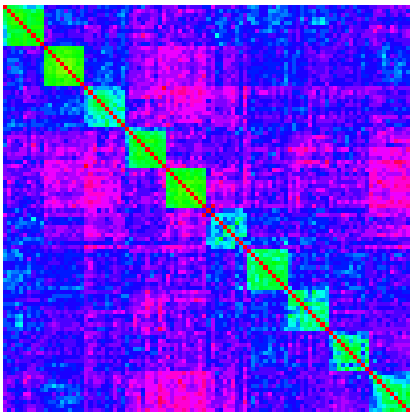
One such constraint is the geometry of such features, since they do not obey conventional Euclidean properties. Another constraint for temporal data such as human actions is the need for speed invariance or *warping*, which causes two sequences to be mis-aligned in time inducing unwanted distortions in the distance metric. Figure 1 shows the effects of ignoring warping, in the context of human actions. Accounting for warping reduces the intra-class distance and improves the inter-class distance. Consequently, statistical quantities such as the *mean sequence* are distorted as seen in figure 1 for two actions S_1 and S_2. Such effects can cause significant performance losses when using building class templates, without accounting for the changes in speed. The most common way to solve for the mis-alignment problem is to use dynamic time warping (DTW) which originally found its use in speech processing [19]. For human actions, [123, 146] address this problem using different strategies for features in the Euclidean space. However, DTW behaves as a similarity measure instead of a true distance metric in that it does not naturally allow the estimation of statistical measures such as mean and variance of action trajectories. We seek a representation that



(a) Row wise from top – S_1, S_2, Warped action $\tilde{S}_2$, Warped mean, Unwarped mean



(b) Unwarped actions



(c) Warped actions

Figure 1: Row wise from top – S_1, S_2, Warped action $\tilde{S}_2$, Warped mean, Unwarped mean. The TSRVF can enable more accurate estimation of statistical quantities such as average of two actions S_1, S_2.

is highly discriminative of different classes while factoring out temporal warping to reduce the variability within classes, while also enabling low dimensional coding at the sequence level.

Learning such a representation is complicated when the features extracted are non-Euclidean (i.e. they do not obey conventional properties of the Euclidean space). Finally, typical representations for action recognition tend to be extremely high dimensional in part because the features are extracted per-frame and stacked. Any computation on such non-

Euclidean trajectories can become very easily involved. For example, a recently proposed skeletal representation [125] results in a 38220 dimensional vector for a 15 joint skeletal system when observed for 35 frames. Such features do not take into account, the physical constraints of the human body, which translates to giving varying degrees of freedom to different joints. It is therefore a reasonable assumption to make that the *true* space of actions is much lower dimensional. This is similar to the argument that motivated manifold learning for image data, where the number of observed image pixels maybe extremely high dimensional, but the object or scene is often considered to lie on a lower dimensional manifold. A lower dimensional embedding will provide a robust, computationally efficient, and intuitive framework for analysis of actions. In this paper, we address these issues by studying the statistical properties of trajectories on Riemannian manifolds to extract lower dimensional representations or codes. We propose a general framework to *code* Riemannian trajectories in a speed invariant fashion that generalizes to many manifolds, the general idea is presented in figure 2. We validate our work on three different manifolds - the Grassmann manifold, the product space $SE(3) \times \cdots \times SE(3)$, and the space of SPD matrices.

Elastic representations for Riemannian trajectories is relatively new and the lower dimensional embedding of such sequences has remained unexplored. We employ the transport square-root velocity function (TSRVF) representation – a recent development in statistics [110], to provide a warp invariant representation to the Riemannian trajectories. The TSRVF is also advantageous as it provides a functional representation that is Euclidean. Exploiting this we propose to learn the low dimensional embedding with a Riemannian functional variant of popular coding techniques. In other words, we are interested in parameterization of Riemannian trajectories, i.e. for N actions $A_i(t), i = 1 \dots N$, our goal is to learn $\mathcal{F}$ such that $\mathcal{F}(x) = A_i$ where $x \in \mathbb{R}^k$ is the set of parameters. Such a model will allow us to compare actions by simply comparing them in their parametric space with respect to $\mathcal{F}$, with significantly faster distance computations, while being able to reconstruct the original actions. In this work, we learn two different kinds of functions using PCA and

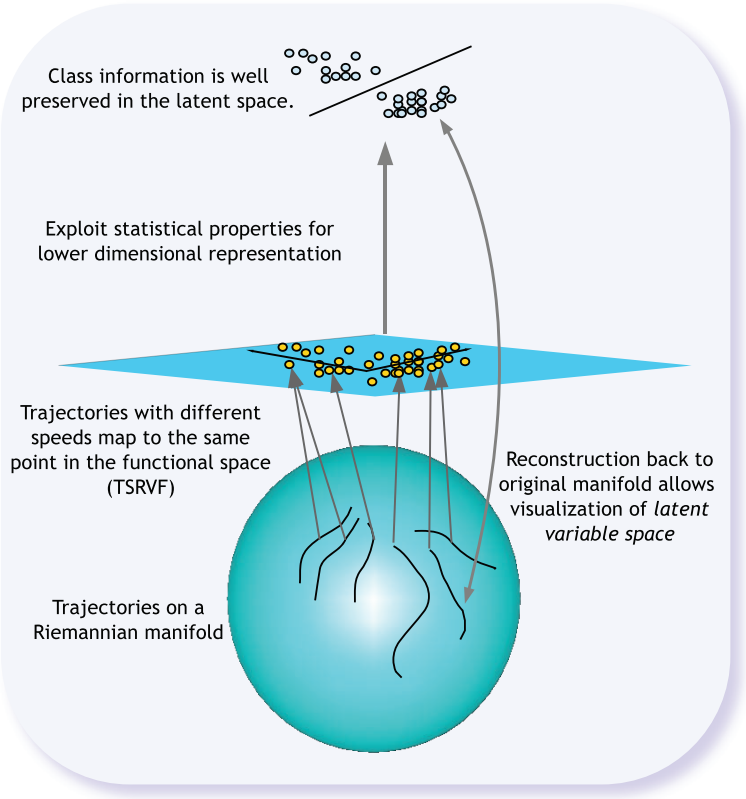


Figure 2: Dimensionality Reduction for Riemannian Trajectories

dictionary learning, which have attractive properties for recognition and visualization.

Broader impact: While one advantage of embedding Riemannian trajectories into a lower dimensional space is the low cost of storage and transmission, perhaps the biggest advantage is the reduction in complexity of search and retrieval in the latent spaces. Although this work concerns itself primarily with recognition and reconstruction, it is easy to see the opportunities these embeddings present in search applications given that the search space dimension is now $\sim 250\times$ smaller. We conclusively demonstrate that the embeddings are as discriminative as their original features, therefore guaranteeing an accurate and fast search. The proposed coding scheme also enables visualization of highly abstract properties of human movement in an intuitive manner. We show results on a stroke rehabilitation project which allows us to visualize the *quality* of movement for stroke survivors. These

ideas present a lot of opportunity towards building applications that provide users with feedback, while facilitating rehabilitation. We summarize our contributions next.

Contributions

1. An elastic vector-field representation for Riemannian trajectories by modeling the TSRVF on the Grassmann manifold, the product space of $SE(3) \times .. \times SE(3)$ and the space of symmetric positive definite matrices (SPD).

2. Dimensionality reduction for Riemannian trajectories in a speed invariant manner, such that each trajectory is mapped to a single point in the low dimensional space.

3. We present results on three coding techniques that have been generalized for Riemannian Functionals (RF) - PCA, KSVD [4] and Label Consistent KSVD [58].

4. We show the application of such embedded features or codes in three applications - action recognition, visual speech recognition, and stroke rehabilitation outperforming all comparable baselines, while being nearly $100 - 250\times$ more compressed. Their effectiveness is also demonstrated in action clustering and diverse action sampling.

5. The low dimensional codes can be used for visualization of Riemannian trajectories to explore the *latent space* of human movement. We show that these present interesting opportunities for stroke rehabilitation.

6. We perform a thorough analysis of the TSRVF representation testing its stability under different conditions such as noise, length of trajectories and its impact on convergence.

1 Related Work

1.1 Elastic Metrics For Trajectories

The TSRVF is a recent development in statistics [110] that provides a way to represent trajectories on Riemannian manifolds such that the distance between two trajectories is

invariant to identical time-warpings. The representation itself lies on a tangent space and is therefore Euclidean, this is discussed further in section 2. The representation was then applied to the problem of visual speech recognition by warping trajectories on the space of SPD matrices [111]. A more recent work [144] has addressed the arbitrariness of the reference point in the TSRVF representation, by developing a purely intrinsic approach that redefines the TSRVF at the starting point of each trajectory. A version of the representation for Euclidean trajectories - known as the Square-Root Velocity Function (SRVF), was recently applied to skeletal action recognition using joint locations in $\mathbb{R}^3$ with promising results [37]. We differentiate our contribution as the first to use the TSRVF representation by representing actions as trajectories in high dimensional non-linear spaces. We use the skeletal feature recently proposed in [125], which models each skeleton as a point on the space of $SE(3) \times \cdots \times SE(3)$. Rate invariance for activities has been addressed before [123, 146], for example [123] models the space of all possible warpings of an action sequence. Such techniques can align sequences correctly, even when features are multi-modal [146]. However, most of the techniques are used for recognition which can be achieved with a similarity measure, but we are interested in a representation which serves a more general purpose to 1) provide an effective metric for comparison, recognition, retrieval, etc. and 2) provide a framework for efficient lower dimensional coding which also enables recovery back to the original feature space.

1.2 Low Dimensional Data Embedding

Principal component analysis has been used extensively in statistics for dimensionality reduction of linear data. It has also been extended to model a wide variety of data types. For high dimensional data in $\mathbb{R}^n$, manifold learning (or non-linear dimensionality reduction) techniques [115, 91] attempt to identify the underlying low dimensional manifold while preserving specific properties of the original space. Using a robust metric, one could theoretically use such techniques for coding, but the algorithms have impractical memory requirements for very high dimensional data of the order of $\sim 10^4 - 10^5$, they also do not pro-

vide a way of reconstructing the original manifold data. For data already lying on a known manifold, geometry aware mapping of SPD matrices [51] constructs a lower-dimensional SPD manifold, and principal geodesic analysis (PGA) [44] identifies the primary geodesics along which there is maximum variability of data points. We are interested in identifying the variability of sequences instead. Recently, dictionary learning methods for data lying on Riemannian manifolds have been proposed [57, 53] and could potentially be used to code sequential data but they can be expected to be computationally more intensive. Coding data on Riemannian manifolds is still a new idea with some progress in the past few years, for example recently the Vector of Locally Aggregated Descriptors (VLAD) has also been extended recently to Riemannian manifolds [43]. However, to the best of our knowledge, coding Riemannian trajectories has received little or no attention, but has several attractive advantages.

Manifold learning of Trajectories: Dimensionality reduction for high dimensional time series is still a relatively new area of research, some recent works have addressed the issue of defining spatial and temporal neighborhoods. For example, [69] recently proposed a generalization of Laplacian eigenmaps to incorporate temporal information. Here, the neighborhoods are also a function of time, but the final reduction step still involves mapping a single point in the high dimensional space to a single point in the lower dimensional space. Next, the Gaussian process latent variable model (GPLVM) [67] and its variants, are a set of techniques that perform non-linear dimensionality reduction for data in $\mathbb{R}^N$, while allowing for reconstruction back to the original space. However, its generalization to non-linear Riemannian trajectories is unclear, which is the primary concern of this work. Quantization of Riemannian trajectories has been addressed in [9], which reduces dimensionality but does not enable visualization. Further, there is loss of information which can cause reduction in recognition performance, whereas we propose to reduce dimensionality by exploiting the latent variable structure of the data. Comparing actions in the latent variable space is similar in concept to learning a linear dynamical system [118] for Euclidean

data, where different actions can be compared in the parametric space of the model.

1.3 Visualization In Biomedical Applications

A promising application for the ideas proposed here, is in systems for rehabilitation of patients suffering from impairment of their motor function. Typically visual sensors are used to record and analyze the movement, which drives feedback. An essential aspect of the feedback is the idea of decomposing human motion into its individual components. For example, they can be used to understand abstract ideas such as movement quality [34], gender styles [42] etc. Troje [42] proposed to use PCA on individual body joints in $\mathbb{R}^3$, to model different styles of the walking motion. However, they work with data in the Euclidean space, and explicitly model the temporality of movement using a combination of sinusoids at different frequencies. More recently, a study in neuroscience [34] showed that the perceived space of movement in the brain is inherently non-linear and that visualization of different movement attributes can help achieve the *most efficient* movement between two poses. This efficient movement is known to be the geodesic in the *pose space* [20]. The study was validated on finger tapping, which is a much simpler motion than most human actions. In this work, we generalize these ideas by visualizing entire trajectories of much more complicated systems such as human skeletons and show results on the movement data of stroke-patients obtained from a motion-capture based hospital system [32].

2 Rate Invariant Sequence Comparison

In this section we describe the Transport Square Root Velocity Function (TSRVF), recently proposed in [110] as a representation to perform warp invariant comparison between multiple Riemannian trajectories. Using the TSRVF representation for human actions, we propose to learn the latent function space of these Riemannian trajectories in a much lower dimensional space. As we demonstrate in our experiments, such a mapping also provides some robustness to noise which is essential when dealing with noisy sensors.

Let α denote a smooth trajectory on $\mathcal{M}$ and let $\mathbb{M}$ denote the set of all such trajectories:

$\mathbb{M} = \{\alpha : [0,1] \mapsto \mathcal{M}|, \alpha \text{ is smooth}\}$. Also define Γ to be the set of all orientation preserving diffeomorphisms of [0,1]: $\Gamma = \{\gamma \mapsto [0,1]|\gamma(0) = 0, \gamma(1) = 1, \gamma \text{ is a diffeomorphism}\}$. It is important to note that γ forms a group under the composition operation. If α is a trajectory on $\mathcal{M}$, then $\alpha \circ \gamma$ is a trajectory that follows the same sequence of points as α but at the evolution rate governed by γ. The group Γ acts on $\mathbb{M}$, $\mathbb{M} \times \Gamma \to \mathbb{M}$, according to $(\alpha, \gamma) = \alpha \circ \gamma$. To construct the TSRVF representation, we require a formulation for parallel transporting a vector between two points $p, q \in \mathcal{M}$, denoted by $(v)_{p \to q}$. For cases where p and q do not fall in the cut loci of each other, the geodesic remains unique, and therefore the parallel transport is well defined.

The TSRVF [110] for a smooth trajectory $\alpha \in \mathbb{M}$ is the parallel transport of a scaled velocity vector field of α to a reference point $c \in M$ according to:

$$h_\alpha(t) = \begin{cases} \frac{\dot{\alpha}(t)_{\alpha(t) \mapsto c}}{\sqrt{|\dot{\alpha}(t)|}} \in T_c(\mathcal{M}), & |\dot{\alpha}(t)| \neq 0 \\ 0 \in T_c(\mathcal{M}) & |\dot{\alpha}(t)| = 0 \end{cases} \tag{5.1}$$

where $|\ .\ |$ denotes the norm related to the Riemannian metric on $\mathcal{M}$ and $T_c(\mathcal{M})$ denotes the tangent space of $\mathcal{M}$ at c. Since α is smooth, so is the vector field h_α. Let $\mathcal{H} \subset T_c(\mathcal{M})^{[0,1]}$ be the set of smooth curves in $T_c(\mathcal{M})$ obtained as TSRVFs of trajectories in $\mathcal{M}$, $\mathcal{H} = \{h_\alpha | \alpha \in \mathcal{M}\}$.

Distance between TSRVFs: Since the TSRVFs lie on $T_c(\mathcal{M})$, the distance is measured by the standard $\mathbb{L}^2$ norm given by

$$d_h(h_{\alpha_1}, h_{\alpha_2}) = \left(\int_0^1 |h_{\alpha_1}(t) - h_{\alpha_2}(t)|^2 \right)^{\frac{1}{2}}. \tag{5.2}$$

If a trajectory α is warped by γ, to result in $\alpha \circ \gamma$, the TSRVF of the warped trajectory is given by:

$$h_{\alpha \circ \gamma}(t) = h_\alpha(\gamma(t)) \sqrt{\dot{\gamma(t)}} \tag{5.3}$$

The distance between TSRVFs remains unchanged to warping, i.e.

$$d_h(h_{\alpha_1}, h_{\alpha_2}) = d_h(h_{\alpha_1 \circ \gamma}, h_{\alpha_2 \circ \gamma}). \tag{5.4}$$

The invariance to group action is important as it allows us to compare two trajectories using the optimization problem stated next.

Metric invariant to temporal variability: Next, we will use d_h to define a metric between trajectories that is invariant to their time warpings. The basic idea is to partition $\mathbb{M}$ using an equivalence relation using the action of Γ and then to inherit d_h on to the quotient space of this equivalence relation. Any two trajectories α_1, α_2 are set to be equivalent if there is a warping function $\gamma \in \Gamma$ such that $\alpha_1 = \alpha_2 \circ \gamma$. The distance d_h can be inherited as a metric between the orbits if two conditions are satisfied: (1) the action of Γ on $\mathbb{M}$ is by isometries, and (2) the equivalence classes are closed sets. While the first condition has already been verified (see Eqn. 5.4), the second condition needs more consideration. In fact, since Γ is an open set (under the standard norm), its equivalence classes are also consequently open. This issue is resolved in [110] using a larger, *closed* set of time-warping functions as follows. Define $\tilde{\Gamma}$ to the set of all non-decreasing, absolutely continuous functions, $\gamma : [0,1] \to [0,1]$ such that $\gamma(0) = 0$ and $\gamma(1) = 1$. This $\tilde{\Gamma}$ is a semi-group with the composition operation. More importantly, the original warping group Γ is a *dense* subset of $\tilde{\Gamma}$ and the elements of $\tilde{\Gamma}$ warp the trajectories in the same way as Γ, except that they allow for singularities [110]. If we define the equivalence relation using $\tilde{\Gamma}$, instead of Γ, then orbits are closed and the second condition is satisfied as well. This equivalence relation takes the following form. Any two trajectories α_1, α_2 are said to be equivalent, if there exists a $\gamma \in \tilde{\Gamma}$ such that $\alpha_1 = \alpha_2 \circ \gamma$. Since Γ is dense in $\tilde{\Gamma}$, and since the mapping $\alpha \mapsto (\alpha(0), h_\alpha)$ is bijective, we can rewrite this equivalence relation in terms of TSRVF as $\alpha_1 \sim \alpha_2$, if **(a.)** $\alpha_1(0) = \alpha_1(0)$, and **(b.)** there exists a sequence $\{\gamma_k\} \in \Gamma$ such that $\lim_{k \mapsto \infty} h_{\alpha_1 \circ \gamma_k} = h_{\alpha_2}$, this convergence is measured under the $\mathbb{L}^2$ metric. In other words two trajectories are said to be equivalent if they have the same starting point, and the TSRVF of one can be time-warped into the TSRVF of the other using a sequence of warpings. We will use the notation $[\alpha]$ to denote the set of all trajectories that are equivalent to a given $\alpha \in \mathbb{M}$. Now, the distance d_h can be inherited on the quotient space, with the result d_s on

$\mathbb{M}/\sim$ (or equivalently $\mathcal{H}/\sim$) given by:

$$d_s([\alpha_1],[\alpha_2]) \equiv \inf_{\gamma_1,\gamma_2\in\tilde{\Gamma}} d_h((h_{\alpha_1},\gamma_1),(h_{\alpha_2,\gamma_2}))$$
$$= \inf_{\gamma_1,\gamma_2\in\tilde{\Gamma}} \left(\int_0^1 \left|h_{\alpha_1}(\gamma_1(t))\sqrt{\dot{\gamma}_1(t)} - h_{\alpha_2}(\gamma_2(t))\sqrt{\dot{\gamma}_2(t)}\right|^2 dt\right)^{\frac{1}{2}} \tag{5.5}$$

The interesting part is that we do not have to solve for the optimizers in $\tilde{\Gamma}$ since Γ is dense in $\tilde{\Gamma}$ and, for any $\delta > 0$, there exists a γ^* such that

$$|d_h(h_{\alpha_1}, h_{\alpha_2} o \gamma^*) - d_s([h_{\alpha_1}],[h_{\alpha_2}])| < \delta. \tag{5.6}$$

This γ^* may not be unique but any such γ^* is sufficient for our purpose. Further, since $\gamma^* \in \Gamma$, it has an inverse that can be used in further analysis. The minimization over Γ is solved for using dynamic programming. Here one samples the interval $[0,1]$ using T discrete points and then restricts to only piecewise linear γ that passes through the $T \times T$ grid. Further properties of the metric d_s are provided in [110].

Warping human actions: In the original formulation of the TSRVF [110], a set of trajectories were all warped together to produce the mean trajectory. In the context of analyzing skeletal human actions, several design choices are available to warp different actions and maybe chosen to potentially improve performance. For example, warping actions per class may work better for certain kinds of actions that have a very different speed profile, this can be achieved by modifying (5.5), to use class information. Next, since the work here is concerned with skeletal representations of humans, different joints have varying degrees of freedom for all actions. Therefore, in the context of skeletal representations, it is reasonable to assume that different joints require different constraints on warping functions. While it may be harder to explicitly impose different constraints to solve for γ, it can be easily achieved by solving for γ per joint trajectory instead of the entire skeleton.

3 Riemannian Functional Coding

A state of the art feature for skeletal action recognition – the Lie Algebra Relative Pairs (LARP) features [125] uses the relative configurations of every joint to every other

joints, which provides a very robust representation, but also ends up being extremely high dimensional. For example, for a 15 joint skeletal system, the LARP representation lies in a 182×6 dimensional space, therefore an action sequence with 35 frames has a final representation that has 38220 dimensions. Such features do not encode the physical constraints on the human body while performing different movements because explicitly encoding such constraints may require hand tuning specific configurations for different applications, which may not always be obvious, and is labor intensive. Therefore, for a given set of human actions, if one can identify a lower dimensional *latent variable* space, which automatically encodes the physical constraints, while removing the redundancy in the original feature representation - one can theoretically represent entire actions as lower dimensional points. This is an extension to existing manifold learning techniques to Riemannian trajectories. It is useful to distinguish the lower dimensional manifold of sequences that is being learned from the Riemannian manifold that represents the individual features such as LARP on $SE(3) \times .. \times SE(3)$ etc. Our goal is to exploit the redundancy in these high dimensional features to learn a lower dimensional embedding without significant information loss. Further, the TSRVF representation, provides us speed invariance which is essential for human actions, this results in an embedding where trajectories that only differ in their rates of execution will map to the same point or to points that are very close in the lower dimensional space.

We study two main applications of coding - 1) visualization of high dimensional Riemannian trajectories, and 2) classification. For visualization, one key property is to be able to reconstruct back from the low dimensional space, which is easily done using principal component analysis (PCA). For classification, we show results on discriminative coding methods such as K-SVD, LC-KSVD, in additional to PCA, that learn a dictionary where each atom is a trajectory. More generally, common manifold learning techniques such as Isomap [115], and LLE [91] can also be used to perform coding, while keeping in mind that it is not easy to obtain the original feature from the low dimensional code. Further, the

trajectories tend to be extremely high dimensional (of the order of $10^4 - 10^5$), therefore most manifold learning techniques require massive memory requirements.

Next we describe the algorithm to obtain low dimensional codes using PCA and dictionary learning algorithms.

Algorithm 4 Riemannian Functional Coding

Input: $\alpha_1(t), \alpha_2(t) \dots \alpha_N(t) \in \mathbb{M}$
Output: Codes $C \in \mathbb{R}^{d \times N}$, in a basis $B \in \mathbb{R}^{D \times d}, d << D$
Compute the Riemannian center of mass $\mu(t)$, which also aligns $\tilde{\alpha}_1(t), \tilde{\alpha}_2(t) \dots \tilde{\alpha}_N(t)$ [110].
for $i \leftarrow [1 \dots N]$ **do**
    for $t \leftarrow [1 \dots T]$ **do**
        Compute shooting vectors $v(i,t) \in T_{\mu(t)}(M)$ as $v(i,t) = \mathbf{exp}^{-1}_{\mu(t)}(\tilde{\alpha}_i(t))$
    end for
    Define $V(i) = [v(i,1)^T \; v(i,2)^T \; \dots \; v(i,T)^T]^T$
end for
$[C, B] = \mathcal{F}(V)$. // $\mathcal{F}$ can be any Euclidean coding scheme

3.1 Representing An Elastic Trajectory As A Vector Field

The TSRVF representation allows the evaluation of first and second order statistics on *entire sequences of actions* and define quantities such as the variability of actions, which we can use to estimate the redundancy in the data similar to the Euclidean space. We utilize the TSRVF to obtain the ideal warping between sequences, such that the warped sequence is equivalent to its TSRVF. To obtain a low dimensional embedding, first we represent the sequences as deviations from a reference sequence using tangent vectors. For manifolds such as $SE(3)$ the natural "origin" I_4 can be used, in other cases the sequence mean [110] by definition lies equi-distant from all the points and therefore is a suitable candidate. In all our experiments, we found the tangent vectors obtained from the mean sequence to be much more robust and discriminative. Next, we obtain the *shooting vectors*, which are the tangent vectors one would travel along, starting from the average sequence $\mu(t)$ at $\tau = 0$ to reach the i^{th} action $\tilde{\alpha}_i(t)$ at time $\tau = 1$, this is depicted in figure 3. Note here that τ is the time in the sequence space which is different from t, which is time in the original manifold space. The combined shooting vectors can be interpreted as a *sequence tangent* that takes us from one point to another in sequence space, in unit time. Since we are representing each trajectory

as a vector field, we can use existing algorithms to perform coding treating the sequences as points, because we have accounted for the temporal information. The algorithm 4 describes the process to perform coding using a generic coding function represented as $\mathcal{F} : \mathbb{R}^D \rightarrow \mathbb{R}^d$, where $d << D$. In the algorithm, C represents the low dimensional representation in the basis/dictionary B that is learned using $\mathcal{F}$.

Complexity: Computing the mean trajectory and simultaneously warping N trajectories for a single iteration can be done in $\mathcal{O}(N(T^2 + \nu))$, where the cost to compute the TSRVF is $\mathcal{O}(\nu)$. If we assume the cost of computing the exponential map is $\mathcal{O}(m)$, algorithm 4 has a time complexity of $\mathcal{O}(mNT)$. This can be a computational bottle neck for manifolds that do not have a closed form solution for the exponential and logarithmic maps. However, the warping needs to be done once offline, as test trajectories can be warped to the computed mean sequence in $\mathcal{O}(T^2 + \nu)$. Further, both the mean and shooting vector computation can be parallelized to improve speed.

Reconstructing trajectories from codes: If the $\mathcal{F}$ is chosen such that it can be easily inverted, i.e. we can find an appropriate $\mathcal{F}^{-1} : \mathbb{R}^d \rightarrow \mathbb{R}^D$, then the lower dimensional embedding can be used to reconstruct a trajectory on the manifold, $\mathcal{M}$, by traveling along the reconstructed tangents from the mean, $\mu(t)$. This is described in algorithm 5.

Algorithm 5 Reconstructing Non Euclidean Trajectories

Input: $C \in \mathbb{R}^{d \times N}, d << D$, $B \in \mathbb{R}^{D \times d}$, $\mu(t)$.
Output: $\widehat{\alpha}(t) \in \mathbb{M}$
for $i \leftarrow [1 \dots N]$ **do**
    $\widehat{V}_i = \mathcal{F}^{-1}(B, C)$
    Rearrange $\widehat{V}_i$ as an $m \times T$ matrix, where T is the length of each sequence.
    for $t \leftarrow [1 \dots T]$ **do**
        $\widehat{\alpha}_i(t) = \mathbf{exp}_{\mu(t)}(\widehat{V}_i(t), 1)$
    end for
end for

3.2 Choices Of Coding Techniques

Since the final representation before dimensionality reduction lies in a vector space, any Euclidean coding scheme can be chosen depending on the application. We focus on two

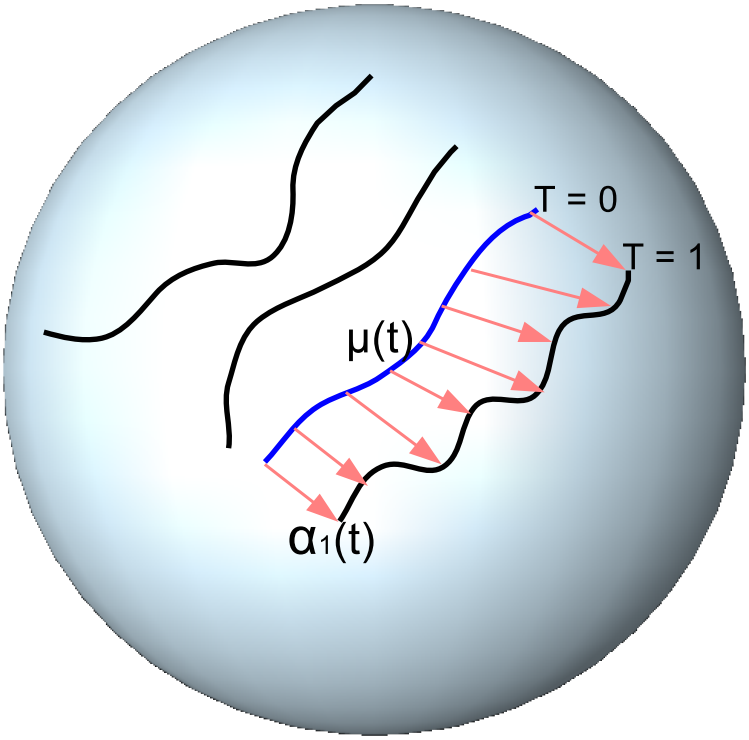


Figure 3: Representing the warped trajectories on a manifold as a vector field, allows us to use existing algorithms to perform dimensionality reduction efficiently, while also respecting the geometric and temporal constraints.

main techniques to demonstrate the ideas. First we perform principal component analysis (PCA) since it can be computed efficiently for extremely high dimensional data, it allows reconstruction by which we can obtain the original features, and it also provides an intuitive interpretation to visualize the high dimensional data in $2D$ or $3D$. This version of Riemannian Functional PCA (RF-PCA, previously referred to as mfPCA in [13]), generalizes functional PCA to Riemannian manifolds, and also generalizes principal geodesic analysis (PGA)[44] to sequential data. Next, we use dictionary learning algorithms, allowing us to exploit sparsity. K-SVD [4] is one of the most popular dictionary learning algorithms that has been influential in a wide variety of problems. Recently, label consistent - KSVD (LCKSVD) [58] improved the results for recognition. K-Hyperline clustering [55] is a special

case of K-SVD where the sparsity is forced to be 1, i.e. each point is approximated by a single dictionary atom. It is expected that since K-SVD relaxes the need for the bases to be orthogonal, it achieves much more efficient codes, that are much more compact, have the additional desirable property of sparsity and perform nearly as well as the original features themselves.

Eigenvalue decay using RF-PCA: To first corroborate our hypothesis that Riemannian trajectories are often far lower dimensional than the feature spaces in which they are observed, we show the eigenvalue decay in figure 4, after performing RF-PCA on three commonly used datasets in skeletal action recognition. It is evident that most of the variation in the datasets is captured by 10-20 eigenvectors of the covariance matrix. It is also interesting to note that that RF-PCA does a good job of approximating the different classes in the product space of $SE(3) \times \cdots \times SE(3)$. The MSRActions dataset [70] contains 20 classes and correspondingly the eigenvalue decay flattens around 20. In comparison the UTKinect [136] and Florence3D [95] datasets contain 10 and 9 classes of actions respectively, which is reflected in the eigenvalue decay that flattens closer to around 10. Features in the RF-PCA tend to be lower dimensional and more robust to noise, which is helpful in reducing the amount of pre/post processing required for optimal performance.

4 Experimental Evaluation

We evaluate our low dimensional Riemannian coding approach in several applications and show their advantages over conventional techniques that take geometry into account as well as other Euclidean approaches. First we address the problem of activity recognition from depth sensors such as the Microsoft Kinect. We show that a low dimensional embedding can perform as well or better than the high dimensional features on benchmark datasets. Next we evaluate our framework on the problem of visual speech recognition (VSR), or also known as lip-reading from videos. We show that, all other factors remaining the same, our low dimensional codes outperform many baselines. Finally, we also address

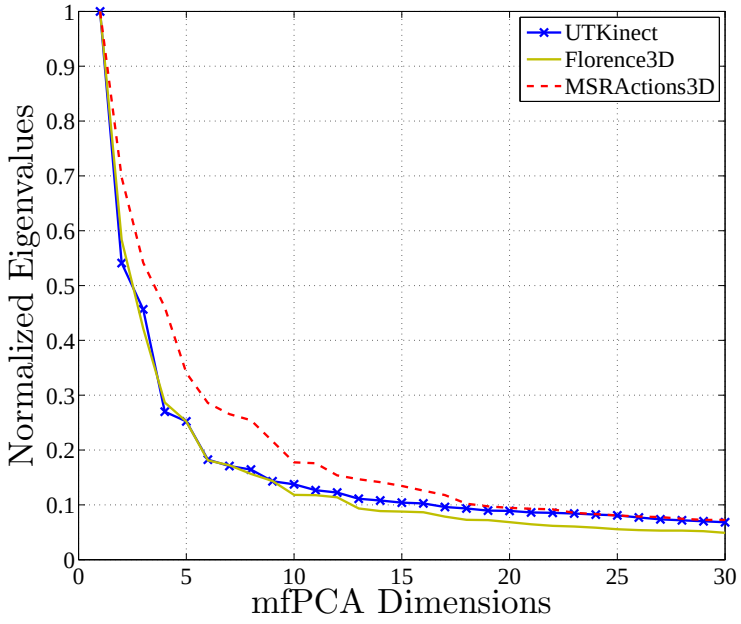


Figure 4: Eigenvalue decay for MSRActions3D [70], UTKinect [136], and Florence3D [95] datasets obtained with RF-PCA. UTKinect and Florence3D have 10 and 9 different classes respectively, as a result the corresponding eigenvalue decay saturates at around 10 dimensions. MSRActions consists of 20 classes and the decay saturates later at around 20.

the problem of movement quality assessment in the context of stroke rehabilitation with state-of-the-art results. We also show that low dimensional mapping provides an intuitive visual interpretation to understand quality of movement in stroke rehabilitation.

4.1 Action Recognition

We use a recently proposed feature called Lie algebra relative pairs (LARP) [125] for skeleton action recognition. This feature maps each skeleton to a point on the product space of $SE(3) \times SE(3) \cdots \times SE(3)$, where it is modeled using transformations between joint pairs. It was shown to be very effective on three benchmark datasets - UTKinect [136], Florence3D [95], and MSR Actions3D [70]. We show that using geometry aware warping results in significant improvement in recognition. Further, we show that it is possible to do so with a representational feature dimension that is 250$\times$ smaller than state-of-the-art.

Florence3D actions dataset [95] contains 9 actions performed by 10 different subjects repeated two or three times by each actor. There are 15 joints on the skeleton data collected using the Kinect sensor. There are a total of 182 relative joint interactions which are encoded in the features.

UTKinect actions dataset [136] contains 10 actions performed by 10 subjects, each action is repeated twice by the actor. Totally, there are 199 action sequences. Information regarding 20 different joints is provided. There are a total of 342 relative joint interactions.

MSRActions3D dataset [70] contains a total of 557 labeled action sequences consisting of 20 actions performed by 10 subjects. There are 20 joint locations provided on the skeletal data, which gives 342 relative joint interactions.

Alternative Representations

We compare the performance of our representation with various other recently proposed related methods:

Lie Algebra Relative Pairs (LARP): Recently proposed in [125], this feature is shown to model skeletons effectively. We will compare our results to those obtained using the LARP feature with warping obtained from DTW and unwarped sequences as baselines.

Body Parts + SquareRoot Velocity Function (BP + SRVF) : A skeleton is a collection of body parts where each skeletal sequence is represented as a combination of multiple body part sequences, proposed in [37]. It is also relevant to our work because the authors use the SRVF for ideal warping, which is the vector space version of the representation used in this paper. The representational dimension is calculated assuming the number of body parts $N_{jp} = 10$, per skeleton[37].

Principal Geodesic Analysis (PGA)[44]: Performs PCA on the tangent space of static points on a manifold. We code individual points using this technique and concatenate the final feature vector before classification.

Feature	Representational Dimension	Accuracy
BP+SRVF [37]	30000	87.04
LARP [125]	38220	86.27
DTW [125]	38220	86.74
PGA [44]	6370	79.01
TSRVF	38200	89.50
RF-KSVD	**45** (sparse)	**88.55**
RF- LCKSVD	**60** (sparse)	**89.02**
RF-PCA	**110**	**89.67**

Table 1: Recognition performance on the Florence3D actions dataset [95] for different feature spaces.

Feature	Representational Dimension	Accuracy
BP+SRVF [37]	60000	91.10
HOJ3D [136]	N/A	90.92
LARP [125]	151,848	93.57
DTW [125]	151,848	92.17
PGA [44]	25308	91.26
TSRVF	151,848	94.47
RF-KSVD	**50** (sparse)	92.67
RF-LCKSVD	**50** (sparse)	**94.87**
RF-PCA	**105**	**94.87**

Table 2: Recognition performance on the UTKinect actions dataset [136].

Evaluation Settings

The skeletal joint data obtained from low cost sensors are often noisy, as a result of which post-processing methods such as Fourier Temporal Pyramid (FTP) [130] have been shown to be very effective for recognition in the presence of noise. FTP is also a powerful tool to work around alignment issues, as it transforms a time series into the Fourier domain and discards the high frequency components. By the nature of FTP, the final feature is

Feature	Representational Dimension	Accuracy
BP + SRVF [37]	60000	$\mathbf{87.28 \pm 2.99}$
HON4D [79]	N/A	82.15 ± 4.18
LARP[125]	155,952	75.57 ± 3.43
DTW[125]	155,952	78.75 ± 3.08
PGA [44]	25,992	72.06 ± 3.12
TSRVF	155,952	84.62 ± 3.08
RF-KSVD	**120** (sparse)	84.45 ± 3.15
RF-LCKSVD	**50** (sparse)	83.60 ± 3.14
RF-PCA	**250**	$\mathbf{85.16 \pm 3.13}$

Table 3: Recognition performance on the MSRActions3D dataset [70] following the protocol of [79] by testing on 20 classes, with all possible combinations of test train subjects.

invariant to any form of warping. One of the contributions of this work is to demonstrate the effectiveness of geometry aware warping over conventional methods, and then explore the space of these warped sequences, which is not easily possible with FTP. Therefore, we perform our recognition experiments on the non-Euclidean features sequences without FTP. We computed the mean on $SE(3)$ extrinsically for the sake of computation, since the Riemannian center of mass for the manifold is iterative. In general this can lead to errors since the log map for $SE(3)$ is not unique, however we found this to work well enough to model skeletal movement in our experiments. This can easily be replaced with the more stable intrinsic version, for details on implementations we refer the reader to [40]. For classification, we use a one-vs-all SVM classifier following the protocol of [125], and set the C parameter to 1 in all our experiments. For the Florence3D and UTKinect datasets we use five different combinations of test-train scenarios and average the results. For the MSRActions dataset, we follow the train-test protocol of [79] by performing recognition on all 242 scenarios of 10 subjects of which half are used for training, and the rest for testing.

Recognition Results

The recognition rates for Florence 3D, UTKinect, and MSRActions3D are shown in tables 1, 2 and 3 respectively. It is clear from the results that using TSRVF on a Riemannian feature, leads to significant improvement in performance. Further, using RF-PCA improves the results slightly, perhaps due to robustness to noise, but more importantly, reduces the representational dimension of each action by a factor of nearly 250. Sparse codes obtained by K-SVD, and LC-KSVD further reduce the data requirement on the features, where LC-KSVD performs as well as RF-PCA while also inducing sparsity in the codes. The improvements are significant compared to using DTW as a baseline; the performance is around 3% better on Florence3D, 2% on UTKinect, and 7% averaged over all test train variations on MSR Actions 3D. Although BP+SRVF [37] has higher recognition numbers on the MSRActions3D, our contribution lies in the significant advantage obtained using the LARP features with RF-PCA (over 7% on average). We observed that simple features in $\mathbb{R}^N$ performed exceedingly well on MSRActions3D, for example using relative joint positions (given by $\overrightarrow{v} = J_1 - J_2$, where J_1 and J_2 are 3D coordinates joints 1 and 2.) on the MSRActions3D with SRVF and PCA we obtain $87.17 \pm 3.08\%$ by embedding every action into $\mathbb{R}^{250}\times$, which is similar to [37], but in a much lower dimensional space. The performance of LCKSVD on MSRActions3D is lower than state-of-the-art because it requires a large number of samples per action class to learn a robust dictionary. There are $\sim$ 20 action classes in the dataset, but only 557 actions, therefore we are restricted to learn a much smaller dictionary. In other datasets with enough samples per class, LCKSVD performs as well as RF-PCA while also generating sparse codes.

We also show that performing PCA on the shooting vectors is significantly better than performing PCA on individual time samples using Principal Geodesic Analysis. The dimensions for LARP features are calculated as $6 \times J \times T$, where J is the number of relative joint pairs per skeleton, and T is the number of frames per video. We learn the RF-PCA basis using the training data for each dataset, and project the test data onto the orthogonal

basis.

4.2 Visual Speech Recognition

Next we evaluate our method Visual Speech Recognition (VSR) on the OuluVS database [145] and show that the proposed coding framework outperforms comparable techniques at a significantly reduced dimensionality. VSR is the problem of understanding speech as observed from videos. The dataset contains audio and video clues, but we will use only the videos to perform recognition, this problem is also known as automatic lipreading. Speech is a dynamic process, and very much like human movement. It is also subject to significant variation in speed, as a result of which accounting for speed becomes important before choosing a metric between two samples of speech [111].

Figure 5: Samples from the OuluVS database [145], used to perform visual speech recognition (VSR) by extracting region covariance matrices which are symmetric positive definite matrices (SPD).

OuluVS database [145]: This includes 20 speakers uttering 10 phrases: *Hello, Excuse me, I am sorry, Thank you, Good bye, See you, Nice to meet you, You are welcome, How are you, Have a good time.* Each phrase is repeated 5 times. All the videos in the database are segmented, with the mouth regions determined by the manually labeled eye positions in each frame. We compare our results to those reported in [111], who used covariance descriptors on the space of SPD matrices to model the visual speech using TSRVF. There are two protocols of evaluation for VSR typically, speaker independent test and speaker dependent test (SDT). We report results on the latter following [111].

Feature Descriptor And Evaluation Settings

We use the covariance descriptor [120] which has proven to be very effective in modeling unstructured data such as textures, materials etc. We follow the feature extraction process as described in [111], to show the effectiveness of our framework. For the covariance descriptor, seven features are extracted including $\{x, y, I(x,y), |\frac{\partial I}{\partial x}|, |\frac{\partial I}{\partial x}|, |\frac{\partial^2 I}{\partial x}|, |\frac{\partial^2 I}{\partial x}|\}$, where x, y are the pixel locations, $I(x,y)$ is the intensity of the pixel, and the remaining terms are the first & second partial derivatives of the image with respect to x, y. This is extracted at each pixel, within a bounded region around the mouth. These covariance matrices are summed up to obtain a single 7×7 region covariance descriptor per frame. These form a trajectory of such matrices per video, which we use to calculate its TSRVF and subsequently the low dimensional codes.

We show improved results are achieved while also providing highly compressed feature representations as shown in Table 4. We train a one-vs-all SVM similar to the previous experiment, on the shooting vectors directly, by training on 60% of the subjects for each spoken phrase, this is repeated for all train/test combinations. We obtain an accuracy of 74.05% on uncompressed shooting vectors, as compared to 66.0% using a 1-NN classifier on all the 1000 videos proposed in [111]. The functional codes using different coding schemes outperform even the SVM results by around 1.5%. While the improvement is not significant, it is important to note that there is a reduction in the feature representation by a factor of nearly 100×.

4.3 Movement Quality For Stroke Rehabilitation

Each year stroke leaves millions of patients disabled with reduced motor function, which severely restricts a person's ability to perform activities of daily living. Fortunately, the recent decade has seen the development of rehabilitation systems with varying degrees of automated guidance, to help the patients regain a part of their motor function. A typical system is shown in figure 6, which was developed by Chen et al. [32]. The system uses

Feature	Representational Dimension	Accuracy
Cov SPD [120]	2450	31.9
TSRVF + NN [111]	2450	66.0
Spatio-temporal[145]	N/A	70.2 (800 videos)
PGA [44]	1000	72.42 ± 3.14
TSRVF + SVM	2450	74.05 ± 4.14
RF - LCKSVD	**20** (sparse)	74.04 ± 3.5
RF - KSVD	**20** (sparse)	$\mathbf{75.63 \pm 4.45}$
RF - PCA	30	$\mathbf{75.3 \pm 5.41}$

Table 4: Visual speech recognition performance on the OuluVS database [145] on 1000 videos using the subject dependent testing (SDT). Results show that the functional coding representation outperforms previous state-of-the-art with similar features, while significantly reducing dimensionality.

Figure 6: The stroke rehabilitation system [32], that uses a 14 marker configuration to provide feedback on motor function for stroke patients. A typical evaluation protocol requires a therapist to observe a specified movement to give a score indicating the quality of movement.

14 markers to analyze and study the patient's movement (eg. reach and grasp), usually in the presence of a therapist who then provides a *movement quality score*, such as the Wolf

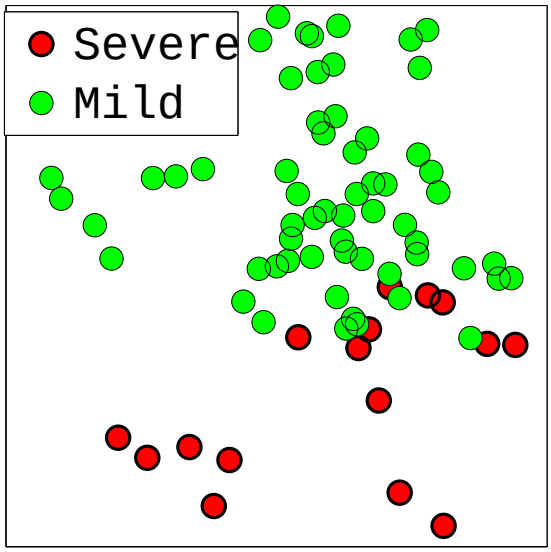
Motor Function Test (WMFT) [134].

Our goal in this experiment is to predict the quality of the stroke survivor's movement as well as the therapist, so that such systems can be home-based with fewer therapist interventions. There are 14 markers on the right hand, arm and torso in a hospital setting. A total of 19 impaired subjects perform multiple repetitions of reach and grasp movements, both on-table and elevated (with the additional force of gravity acting against their movement). Each subject performs 4 sets of reach and grasp movements to different target locations, with each set having 10 repetitions.

Feature Description And Evaluation Settings

We choose 4 joints – back, shoulder, elbow, and wrist. This is used to represent them in relative configurations to each other as is done in LARP [125] resulting in each *hand skeleton* that lies in $SE(3) \times \cdots \times SE(3)$ as earlier. The problem now reduces to performing logistic regression on trajectories that lie in $SE(3) \times \cdots \times SE(3)$. The stroke survivors were also evaluated by the WMFT [134] on the day of recording, where a therapist evaluates the subject's ability on a scale of 1 - 5 (with 5 being least impaired to 1 being most impaired). We use these scores as the ground truth, and predict the quality scores using the LARP features extracted from the hand markers. The dataset is small in size due to the difficulty in obtaining data from stroke survivors, therefore we use the evaluation protocol of [126], where we train on all but one test sample for regression. We compare our results to Shape of Phase Space (SoPS) [126], who perform a reconstruction of the phase space from individual motion trajectories in each dimension of each joint.

Table 7b shows the results for different features. The baseline, using the features as it is, gives a correlation score of 92.27 to the therapist's WMFT evaluation. Adding elasticity to the curves in the $SE(3)$ product space improves the correlation score to 93.53. The functional codes improves the score significantly to 97.84, while using only 70 dimensions giving state of the art performance. We also compare our score to the kinematic based features proposed by [126]. **Visualizing quality:** Next, figure 7a shows the different



(a) Easily visualizing quality of movement in RFPCA space

Feature	Dimension	Score
SoPS* [126]	2100	88.6
KIM* [32]	NA	85.2
LARP [125]	79200	92.27
LARP + TSRVF [110]	79200	93.53
RF-PCA	**70**	**97.84**
RF-KSVD	**25** (sparse)	75.76

(b) Predicting the quality of movement in the rehabilitation of stroke survivors.

Figure 7: The RF-PCA is able to accurately predict movement quality as compared to an expert therapist which can improve home-based systems for stroke rehabilitation.

movements in the lower dimensional space. Visualizing the movements in RF-PCA space, it is evident that even in $\mathbb{R}^2$, information about the quality of movement is captured. Movements which are indicative of high impairment in the motor function appear to be physically separated from the movements which indicate mild or less impairment. It is easy to see the opportunities such visualizations present for rehabilitation, for example a recent

study in neuroscience [34] showed that real-time visual feedback can help *learn* the most efficient control strategies for certain movements.

4.4 Reconstruction And Visualization Of Actions

We also show results on visualization and exploration of human actions as Riemannian trajectories. Since shapes are easy to visualize, we use the silhouette feature as a point on the Grassmann manifold.

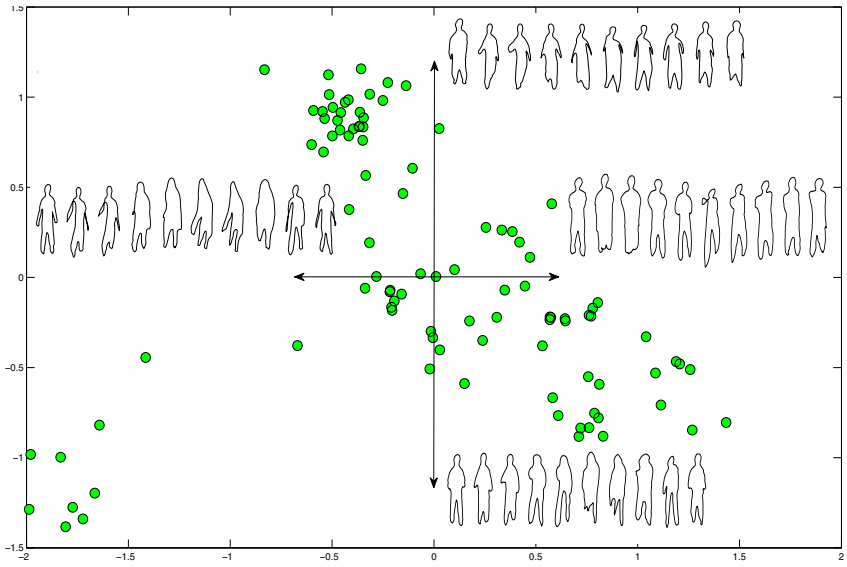
UMD actions dataset [123]: This is a relatively constrained dataset, which has a static background allowing us to easily extract shape silhouettes. It contains 100 sequences consisting of 10 different actions repeated 10 times by the same actor. For this dataset, we use the shape silhouette of the actor as our feature, because of its easy visualization as compared to other non-linear features.

RECONSTRUCTION RESULTS

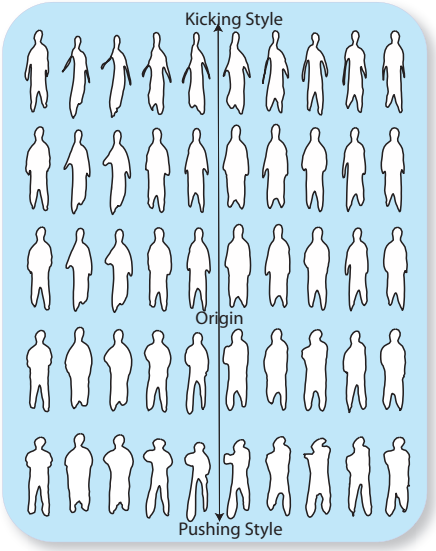
Once we have mapped the actions onto their lower dimensional space using RF-PCA, we can reconstruct them back easily using algorithm 5. We show that high dimensional action sequences that lie in non-Euclidean spaces can be effectively embedded into a lower dimensional latent variable space. Figure 8b shows the sampling of one axis at different points. As expected, the "origin" of the dataset contains no information about any action, but moving in the positive or negative direction of the axis results in different *styles* as shown. Note, that since we are only using 2 dimensions, there is a loss of information, but the variations are still visually discernible.

4.5 DIVERSE SEQUENCE SAMPLING

Next, we show that applications such as clustering can also benefit from a robust distance metric that the TSRVF provides. Further, performing clustering is significantly faster in the lower dimensional vector space, such as the one obtained with RF-PCA. We perform these experiments on the UMD Actions data with actions as trajectories on the Grassmann



(a) Exploring the action space in $\mathbb{R}^2$.



(b) Exploring the 2D latent action space

Figure 8: **Exploring the latent variable space of actions** in the UMD actions dataset using RF-PCA. Notice the "origin" contains no information about any action, and moving along an axis provides different abstract style information.

manifold. K-means for data on manifolds involves generalizing the notion of distance to the geodesic distance and the mean to the Riemannian center of mass. We can further generalize this to sequences on manifolds by replacing the geodesic distance with the TSRVF distance

and the mean by the RCM of sequences as defined in [110]. A variant of this problem is to identify the different kinds of groups within a population, i.e. clustering *diversly*, which is a harder problem in general and cannot be optimally solved using K-means. Instead we use manifold Precis which is a *diverse sampling* method [99]. Precis is an unsupervised exemplar selection algorithm for points on a Riemannian manifold, i.e. it picks a set of K most representative points S from a data set X. The algorithm works by jointly optimizing between approximation error and diversity of the exemplars, i.e. forcing the exemplars to be as different as possible while covering all the points.

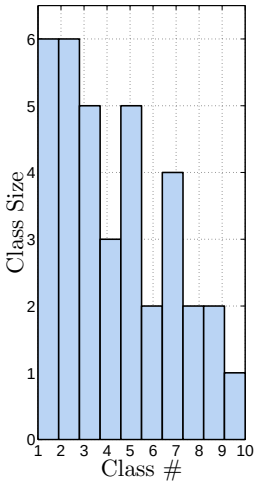
To demonstrate the generalizability of our functional codes, we perform an experiment to perform K-means clustering and diverse clustering of *entire sequences*. In the experiment on the UMD actions dataset, we constructed a collection of actions that were chosen such that different classes had significantly different populations in the collection. Action centers obtained with K-medoids is shown in figure 9b and as expected classes which have a higher population are over represented in the chosen samples as compared to Precis (figure 9c) which is invariant to the distribution. Due to the low dimensional Euclidean representation, these techniques can be easily extended to suit sequential data in a speed invariant fashion due to the TSRVF and at speeds $\sim 500\times$ faster due to RF-PCA.

5 Analysis Of The Tsrvf Representation

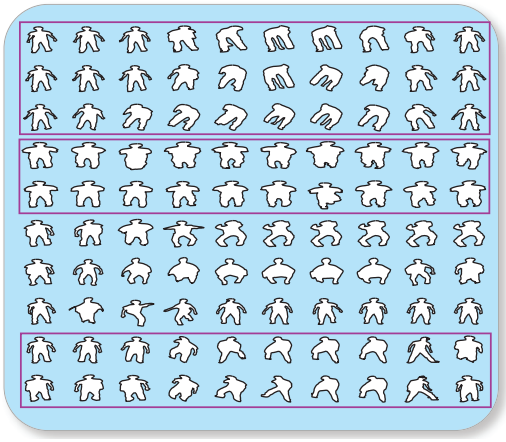
In this section, we consider different factors that influence the stability and robustness of the TSRVF representation, thereby affecting its coding performance. Factors such as (a) it's stability for different choices of the reference point, (b) the effect of noise on functional coding, and (c) arbitrary length of a trajectory, are realistic scenarios that occur in many applications.

5.1 STABILITY TO THE CHOICE OF REFERENCE POINT

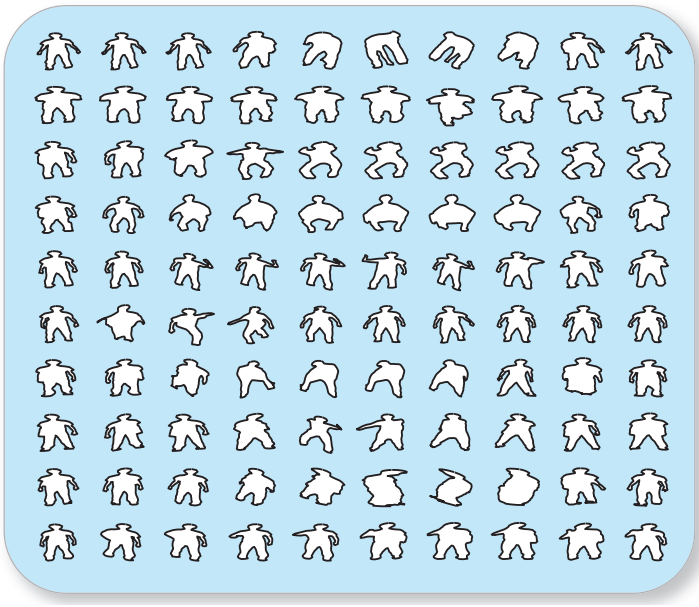
A potential weakness in the present TSRVF framework is in the choice of the reference point c, which may introduce unwanted distortions if chosen incorrectly. In manifolds such



(a) A dataset with skewed class proportions



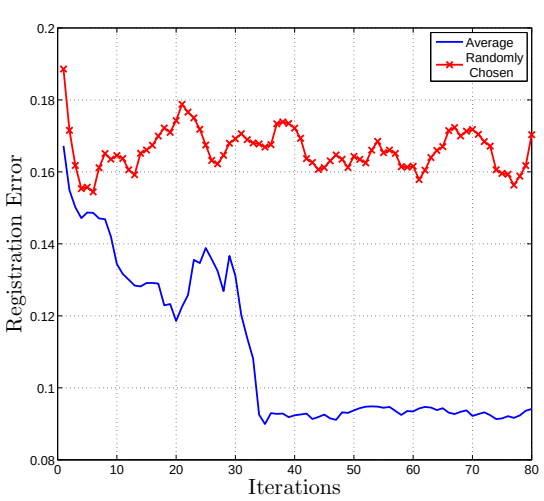
(b) Row-wise: functional K-medoids



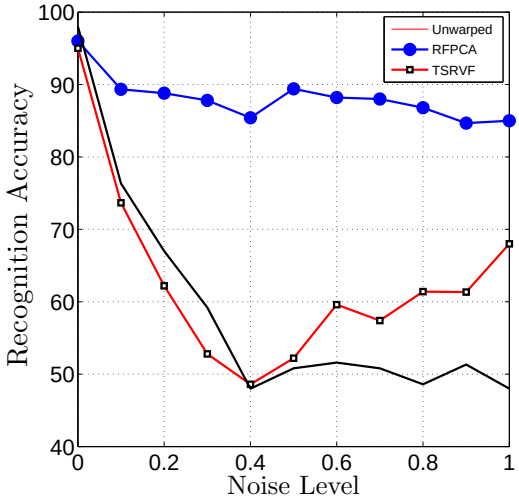
(c) Row-wise: functional-Precis

Figure 9: Diverse action sampling using Precis[99] by sampling in RF-PCA space $\in \mathbb{R}^{10}$ on a highly skewed dataset. K-medoids picks more samples (marked) from classes that have a higher representation, while Precis remains invariant to it. The K-medoids and diverse clustering operations are performed $\sim 500\times$ faster in the RF-PCA space. Figure 8b shows a 2D axis sampled in the latent space. It's clearly seen that even in only 2 dimensions, some action information ("style") is discernible.

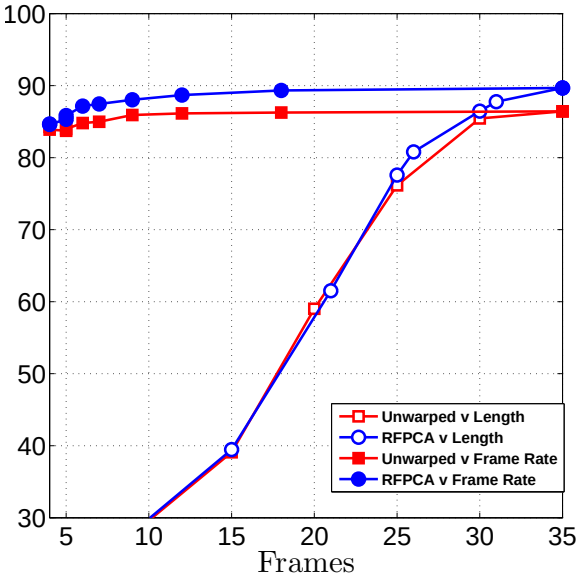
as the $SE(3)$ and SPD, a natural candidate for c is I_4, however for other manifolds such as the Grassmann, the reference must be chosen experimentally. In such cases, a common



(a) Convergence & choice of reference point $\mathbf{C}$



(b) Presence of Noise



(c) Trajectory length and sampling rate

Figure 10: Robustness experiments for different factors as measured by their effect on recognition accuracy. Experiments in table (10a) and figure (10b) are performed on the Grassmann manifold, & figure (10c) shows results on the $SE(3) \times SE(3) \ldots SE(3)$ manifold. It can be clearly seen that the RFPCA representation is robust in the presence of noise, and remains more robust to different sampling rates than unwarped trajectories.

solution is to choose the Riemannian center of mass (RCM), since it is equally distant from all the points thereby minimizing the possible distortions. In our experiments we show that choosing an arbitrarily bad reference point can lead to poor convergence when warping

multiple trajectories. We test the stability of the TSRVF representation to the choice of reference point by studying the convergence rate. We chose a set of 10 similar actions from the UMD actions dataset and measured registration error over time. The registration error is measured as $\Sigma_j d(\mu(t) - \alpha_j(t))^2$, where $\mu(t)$ is the current estimate of the mean as described in algorithm 4. When c is chosen as the mean, the convergence occurs in about 35 iterations as seen in 10a. To generate an arbitrary reference point, we chose a point at random from the dataset and travel along an arbitrary direction from that point. The resulting point is chosen as the new reference point and the unwarped trajectories are now aligned by transporting the TSRVFs to the new c. In order to account for the randomness, we repeat this experiment 10 times and take the average convergence error. The distortion is clearly visible in figure 10a, where there is no sign of convergence even after 80 iterations.

5.2 EFFECT OF NOISE

In the Euclidean setting, the robustness of PCA to noisy data is well known. We examine the consequences of performing PCA on noisy trajectories for activity recognition here. There are many different stages of adding noise to a trajectory in this context - a) sensor noise which is obtained due to poor background segmentation or sensor defect that causes the resulting shape feature to be distorted, b) warping noise that is caused by a poor warping algorithm and c) TSRVF noise, which is obtained due to a poor choice of the reference point, or obtained as a consequence to parallel transport. We have studied the effect of the reference point previously, and the effect of poor warping is unlikely in realistic scenarios. We consider the noise at the sensor level which is most likely, by inducing noise in the shape feature. We perform this by perturbing each shape point on the Grassmann manifold along a random direction, $v_r \in \mathcal{T}_{\alpha(i)}(\mathcal{G})$, for a random distance, k drawn from a uniform distribution: $k \in \mathcal{U}(0,1)$. We generate the random tangent and the random distance to be traversed uniformly. Therefore, the i^{th} point in a trajectory is transformed as : $\hat{\alpha}(i) = \mathbf{exp}(\alpha(i), k\, v_r)$. We then perform a recognition experiment on the noisy datasets using the RFPCA, TSRVF and unwarped representations. Figure 10b shows the results of

the experiment on the UMD actions dataset, with k on the X-axis. As expected, the RFPCA representation is least affected, while the TSRVF representation performs slightly better than the unwarped trajectories. The different levels of noise indicate how far along the random vector one traverses to obtain the new *noisy* shape.

5.3 ARBITRARY LENGTH & SAMPLING RATES

The choice of parameter T in algorithm 4, directly affects the resulting dimensionality of the the trajectory before performing coding. Here we investigate its effect on coding and recognition. We can generate different trajectory lengths by considering two factors a)frame-rate, where $\widehat{\alpha}(t) = \alpha(\mathbf{m}t)$ where the factor is governed by $\mathbf{m}$, and b) arbitrary end point, where $\widehat{\alpha}(t) = \alpha(1 : T')$, such that $T' < T$. The TSRVF is invariant to frame rate or sampling rate, therefore for a wide range of sampling rates, the recognition accuracy remains unchanged. To observe this, we perform a recognition experiment on the Florence3D skeleton actions dataset. The results for both factors are shown in figure 10c, and it is seen that in both cases the TSRVF warped actions are recognized better than the unwarped actions with an average of 5% better accuracy.

Canonical length: Using the coding framework proposed in this paper, it is conceivable that there is a close relationship between the *true* length of a trajectory and its intrinsic dimensionality. For example - a more complex trajectory contains more information which naturally requires a higher dimensional RFPCA space to truly capture its variability. However, determining the explicit relationship between the RF-PCA dimension and the canonical length of a trajectory is out of the scope of this work.

6 Conclusion

In this chapter we introduced techniques to explore and analyze sequential data on Riemannian manifolds, applied to human activities, visual speech recognition, and stroke rehabilitation. We employ the TSRVF space [110], which provides an elastic metric between two trajectories on a manifold, to learn the latent variable space of actions, which

is a generalization of manifold learning to Riemannian trajectories. We demonstrate these ideas on the curved product space $SE(3) \times \cdots \times SE(3)$ for skeletal actions, the Grassmann manifold, and the SPD matrix manifold. We propose a framework that allows for the parameterization of Riemannian trajectories using popular coding methods – RF-PCA which generalizes functional PCA to manifolds and PGA to sequences, sparsity inducing coding RF-KSVD and discriminative RF-LCKSVD. The learned codes not only provide a compact and robust representation that outperforms many state of the art methods, but also the visualization of actions due to its ability to reconstruct original non-linear features. We also show applications for intuitive visualization of abstract properties such as quality of movement, which has a proven benefit in rehabilitation. The proposed representation also opens up several opportunities to understand various properties of Riemannian trajectories, including their canonical lengths, their intrinsic dimensionality, ideal sampling rates, and other inverse problems which are sure to benefit several problems involving the analysis of temporal data.

Chapter 6

A HETEROGENEOUS DICTIONARY MODEL FOR HUMAN ACTIONS

Previously, we studied methods that would allow us to *generate new actions* by sampling the subspace learned using mfPCA. In this section, we will look at another way to learn a generative model for actions, that is also able to discriminate effectively. Here, instead of representing each action as a point, we will assume actions to be piecewise linear models and attempt to represent them sparsely using dictionary learning. Sparse coding attempts to represent data vectors using a linear combination of a small number of vectors chosen from a 'dictionary'. The dictionary that leads to an optimal sparse representation can be either predefined or learned from the training samples themselves. It is now well known that the latter can lead to improved representation and recognition results [4, 76]. If the data is truly low-dimensional, sparse coding can effectively identify its low degrees of freedom, and hence sparse models have proved successful in several inverse problems in signal/image processing [4], and computer vision [135]. When compared to classical subspace methods which are efficient only if the data lies in a single low-dimensional subspace, sparse coding can recover data lying in a union of low-dimensional subspaces and hence provide a greater flexibility in representation.

Traditionally, most sparse coding applications deal with static data such as images, but there have been recent attempts to extend these concepts to videos [50, 84]. To this end, problems of activity analysis have gained lot of attention where typically a dictionary is learned either per class of actions or on the entire set of all actions and sparse codes are generated per frame. Most human actions evolve over time where they usually begin with a rest pose and end in an extreme pose. This transition is smooth resulting in smoothly varying features. The geometric structure of these transitions is not known in general, but attempts have been made to model this structure, e.g. actions have been considered to

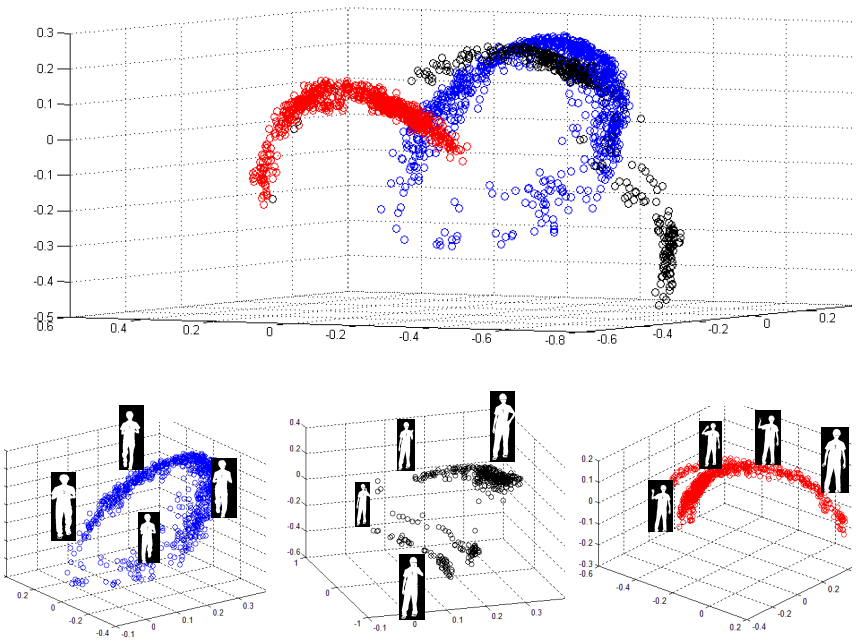


Figure 1: Here we show the feature evolution of *Running, Talk on Phone* and *Waving*. The features are projected to a lower dimensional space for visualization. The top figure shows the three actions on a common coordinate frame. It is seen that these structures can be well approximated by piece-wise linear models.

trace out non-linear manifolds in feature spaces [41]. While such models are quite rich and general, they are accompanied by difficulties in learning the model and coding data using the model. However, as shown in fig 1, a simple piecewise linear model is sufficient to represent most common activities such as *Waving, Running* and *Talking on the phone*. In addition to the representational simplicity, this also affords solving the sparse-coding problem efficiently.

In such cases, centered clustering approaches such as K-Means will not be able to effectively model the underlying patterns which will result in a loss in performance. To cluster data that lies along hyperlines, He *et al.* [56] proposed the K-hyperline clustering algorithm, which is an iterative procedure that performs a least squares fit of K one dimensional linear subspaces to the training data. The relation between K-hyperline clustering and dictionary learning has been explored in [116]. Taking into consideration that cluster centers computed by this algorithm are constrained to pass through the origin, we propose a new heteroge-

neous dictionary model. The elementary features in this dictionary correspond to the $1D$ affine subspaces that represent human activities and hence the dictionary is interpretable. The proposed dictionary is learned with features that are extracted per frame from the videos in an action dataset.

Although several dictionary learning approaches are known, only a few have been proposed that consider the geometric structure along which activities evolve. A few recent attempts have been made to generalize dictionary learning to Riemannian manifolds [52, 53, 57], but none of them deal with sequences which is of interest here. Most of the methods involve improving an initial dictionary, obtained using methods such as K-SVD [4], by maximizing information between dictionary atoms [84], learning class specific dictionaries [50] etc. The idea of features lying along lines has been used before - Taheri *et al.* [113] modeled facial expressions as deviations along geodesics, which are generalizations of high dimensional lines to non Euclidean spaces, from a "neutral expression", and Troje [42] showed that using simple PCA one can identify important directions in landmark data, that are later used for applications like gender classification.

We present a dictionary model for human activities by considering piecewise linear models of activities. Each dictionary atom consists of a tuple - a point and a direction in space. We also introduce new constraints to the traditional sparse coding problem, and adapt it to the heterogeneous dictionary. We show that this can be an effective generative model for human actions. Furthermore, we demonstrate that using such a dictionary, one can achieve state-of-the-art recognition results, and maintain very low reconstruction errors for unseen test activities.

1 Proposed Dictionary Model

In this section, we will formulate our dictionary learning problem and present a method to generate sparse codes using the proposed dictionary.

1.1 Learning The Dictionary

When a dictionary is constructed using K-hyperline clustering, each atom corresponds to a linear subspace. We generalize this dictionary to be a collection of affine subspaces, where each atom is described by a point and an associated direction in space. To learn such a dictionary, we propose a $1D$ affine subspace clustering algorithm. In this method, we incorporate an additional step of calculating the sample mean $\boldsymbol{\mu}_j$ of the j^{th} cluster along with the least-squares fit of a $1D$ subspace, $\mathbf{d}_j$, in K-hyperline clustering. The algorithm is described in Table 1. To identify the cluster membership, we project a data sample onto each dictionary atom and choose the one that results in the least representation error. The projection is performed as

$$P_{\mathbf{H}}(\mathbf{x}) = \boldsymbol{\mu} + \hat{\beta}\mathbf{d}, \quad \text{where} \quad \hat{\beta} = \min_{\beta} \|\mathbf{x} - \boldsymbol{\mu} - \beta\mathbf{d}\|_2^2. \tag{6.1}$$

Note that in this case, the least squares solution for β is $\mathbf{d}^T(\mathbf{x} - \boldsymbol{\mu})$.

1.2 Sparse Coding

Let us assume that a test sample in $\mathbb{R}^n$ can be represented as a linear combination of a small number of affine subspaces. Assuming that the set of dictionary atoms given by $\{\boldsymbol{\mu}_j, \mathbf{d}_j\}_{j=1}^K$ is known, the generative model for a test sample $\mathbf{x}$ can be written as

$$\mathbf{x} = \sum_{j \in S} \alpha_j \boldsymbol{\mu}_j + \beta_j \mathbf{d}_j. \tag{6.2}$$

where S is the set of atoms that participate in the representation of $\mathbf{x}$.

The solution to (6.2) can be obtained using convex programming. The key consideration is that for a given j, $\boldsymbol{\mu}_j$ and $\mathbf{d}_j$ must be chosen together. Furthermore, it is also useful to ensure that the new mean is in the convex hull of the means of S. This can be posed and solved as group Lasso [142],

$$\begin{aligned} \arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \|\mathbf{x} - (\mathbf{M}\boldsymbol{\alpha} + \mathbf{D}\boldsymbol{\beta})\|_2^2 + \lambda \sum_{i=1}^{K} \left\| \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \right\|_2 \\ \text{s.t. } \alpha_i \geq 0, \sum_i \alpha_i = 1, \end{aligned} \tag{6.3}$$

Input

Features $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ and size of dictionary, K.

Output

Affine subspaces $\{\mathbf{H_1}, \cdots, \mathbf{H_K}\}$ represented using the means $\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}$ and the directions $\{\mathbf{d}_1, \cdots, \mathbf{d}_K\}$.

Membership classes, $C_1, \cdots, C_K$.

Algorithm

Initialize: $\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}$ and $\{\mathbf{d}_1, \cdots, \mathbf{d}_K\}$.

while convergence not reached

Compute memberships:

- For each sample $\mathbf{x}_i$ compute the projection of $\mathbf{x}_i$ onto

each $\mathbf{H}_j$, denoted by $P_{\mathbf{H_j}}(\mathbf{x}_i)$.

- $k = \arg\min_j ||\mathbf{x}_i - P_{\mathbf{H_j}}(\mathbf{x}_i)||_{j=1}^K$ and $C_k = C_k \cup \{i\}$.

Update $\mathbf{H}_j$: For each cluster j, compute $\{\boldsymbol{\mu}_j, \mathbf{d}_j\}$ as the

sample mean and the first principal component of all

samples indexed by C_j, respectively.

end

Table 1: The dictionary learning algorithm.

where $\mathbf{M} = [\boldsymbol{\mu}_j]_{j=1}^K$ and $\mathbf{D} = [\mathbf{d}_j]_{j=1}^K$.

2 Experimental Validation

In this section, we demonstrate the use of the dictionary model in representation and recognition of human actions. First, we perform an experiment to validate the proposed generative model, in comparison to a centered clustering approach. Following this, we show

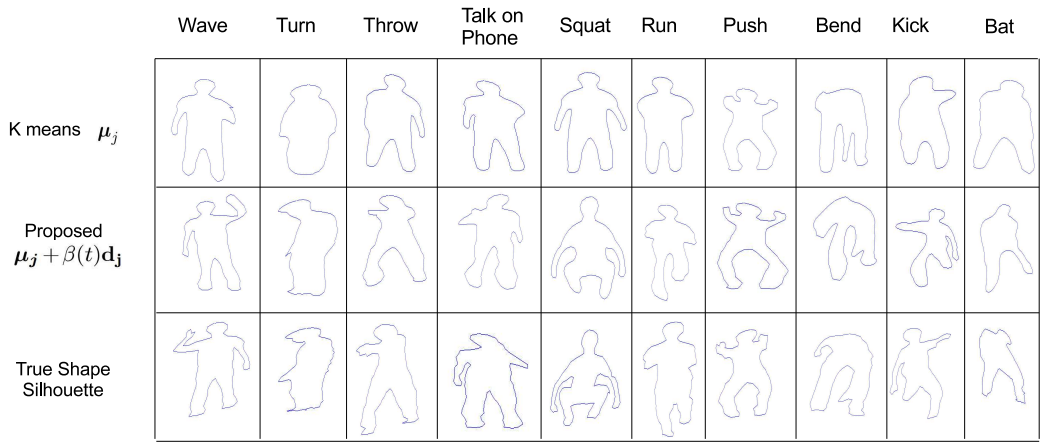


Figure 2: Actions generated by sampling along the learned lines on the UMD actions data set [123]. Some generated actions such as *wave, talk on phone, kick* appear to be laterally inverted as our representation is affine invariant.

that this dictionary can generalize well in representing unseen human actions. Finally, we demonstrate that by aggregating the sparse codes in multiple temporal scales, we can achieve the state-of-the-art performance in activity recognition.

2.1 Generative Model For Human Actions

In this experiment we show that the proposed dictionary can be used to parameterize human actions, thereby demonstrating that the model is an intuitive choice. We perform this experiment using a shape feature due to its obvious advantage in visualization. We use the UMD Actions Dataset [123], as its background is relatively static and allows us to do easy background subtraction. Having extracted the foreground, we perform morphological operations and extract the contour of the human. We sampled a fixed number of points on the contour to obtain the set of landmarks describing the shape. To represent these landmarks, we used an affine invariant representation where the set of m landmark points are given by the $m \times 2$ matrix $L = [(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)]$ for the centered shape. However, shape features do not lie in the Euclidean space [106] and one must take into account the non-linearity of the space while dealing with them. Since we are dealing with

the vector space, we will use embedding approaches as they are conceptually simpler and easier to implement. These allow us to work with these complex features while staying in a Euclidean space. With each set of landmarks, we generate an $m \times m$ projection matrix that is $P = UU^T$, where $L = USV^T$ is the rank-2 SVD. Let P_v be the vectorized form of P, we use P_v as a feature to learn our dictionary. To recover the shape from this vector we re-obtain the projection matrix P and perform a rank-2 SVD on it. Now the feature corresponding to a shape at time t is generated as $P_v(t) = \boldsymbol{\mu_j} + \beta(t)\mathbf{d_j}$, parameterized by $\beta(t)$ which determines to what extent one must travel from $\boldsymbol{\mu_j}$ along the direction $\mathbf{d_j}$. We used different values of β for each action in the range $-1 < \beta(t) < 1$. In fig 2, we show the generated silhouette in each action and compare it to the ground truth.

2.2 Reconstruction Of Unseen Actions

In this experiment, we test the efficiency of the proposed dictionary in modeling unseen actions from test data. Since every action is modeled as a combination of means and directions, an unseen action will typically have a mean that is different from any of the previously learned actions. Hence, we model the new mean as a linear combination of means and find its principal direction as a combination of the known directions. For our experiments, we obtained activities from the Weizmann activity dataset [48] which consists of 90 videos of 10 different actions, each performed by 9 different persons. The classes of actions include running, jumping, walking, side walking etc. In order to evaluate the performance of the proposed sparse coding model, we used the features of all subjects from 6 different activities in the Weizmann dataset for obtaining the dictionary and evaluated the reconstruction error for features from the other 4 activities. The set of *unseen* testing activities included *jack, pjump, skip* and *wave1*. For all our experiments on this dataset we used the histogram of oriented optical flow (HOOF) feature that was introduced in [31]. This feature bins optical flow vectors based on their directions and their primary angle with the horizontal axis, weighted by their magnitudes. Using magnitudes alone is susceptible to noise and can be very sensitive to scale. Thus all optical flow vectors, $v = [x, y]^T$ with

direction $\theta = \tan^{-1}(\frac{y}{x})$ in the range -$\frac{\pi}{2} + \pi\frac{b-1}{B} \leq \theta < -\frac{\pi}{2} + \pi\frac{b}{B}$ will contribute by $\sqrt{x^2 + y^2}$ to the sum in bin b, where $1 \leq b \leq B$, typically $B = 30$ is used. Finally, the histogram is normalized to sum up to 1.

Using the training activities, we computed K (fixed at $20, 30$ and 40) clusters to identify the principal directions and their cluster centroids. For the test activities, we performed sparse coding of the features using the computed centers and directions as the dictionary atoms. Table 2 compares the average reconstruction error obtained for features from the test activities using different coding schemes. Since more than one atom can be used for representation, the reconstruction error in our model is significantly lower than those obtained with K-means or K-hyperline clustering. The plot in Fig 3 shows the reconstruction error obtained by varying the sparsity parameter λ.

2.3 Recognition Of Human Activities

In this experiment, we propose a method for performing recognition of human activities from the Weizmann dataset using sparse codes obtained from the features of each activity. Of the 9 subjects that performed the activities, we used 6 subjects from each class for training and the rest for testing. Hence, we used a total of 60 activities for learning the dictionary and training the classifier. Using the features described in the previous experiment, the sparse codes are computed by setting $\lambda = 0.1$. We aggregate the sparse codes of the training features, in multiple temporal scales, to create one overall feature vector per activity. Given a set of sparse codes stacked in a matrix, aggregation is performed by finding the value corresponding to the absolute maximum of elements in each row. Since aggregation destroys temporal information, we divide each activity into 1, 2, 4, and 6 temporal segments, and perform aggregation independently in each, in order to partially preserve the temporal information. Hence, if each sparse code is of length K, we will obtain a overall feature vector of length $13K$. These overall feature vectors are used to train an SVM classifier. For a test activity, the overall feature vector is computed similarly and classification is performed.

Method	No. of clusters		
	K=20	**K=30**	**K=40**
K-means - $\boldsymbol{\mu}$	0.3295	0.3069	0.2985
K-Hyperline $\mathbf{d}$	0.2657	0.2485	0.2399
$(\boldsymbol{\mu}, \mathbf{d})$ Dictionary	**0.1171**	**0.1039**	**0.0956**

Table 2: Comparison of Reconstruction Error Obtained using the Proposed Sparse Coding.

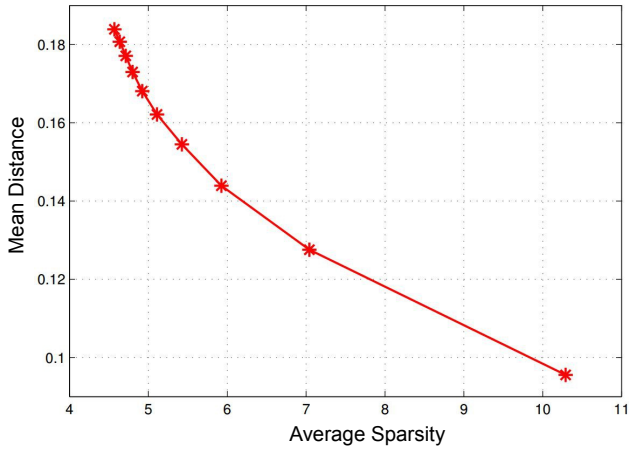


Figure 3: Effect of Sparsity on Reconstruction Error.

Proposed dictionary	**98.88**
K-means dictionary	84.44
Guha *et al.*, Multiple Dictionaries [50]	**98.9**
Guha *et al.*, Single Dictionary [50]	**96.67**
Chaudhry *et al.* [31]	95.66

Table 3: Recognition Performance (%) using the Proposed Sparse Codes.

In order to improve the reliability of recognition results, we repeat the experiment 3 times with randomly chosen training and test sets. Table 3 compares our average performance to other methods reported in the literature. It can be seen that our method compares well with Guha *et al.*, where we are able to match their performance with just a single dictionary as compared to learning a dictionary per class.

3 Conclusion And Future Work

The proposed model opens up several interesting avenues of research, we outline a few of them and conclude our work in this section.

We introduced a sparse representational model for human actions. We first showed that in feature spaces, common actions are approximately piecewise linear. Using this idea, we proposed a dictionary model where each atom is a $1D$ affine subspace described by a mean and an associated direction in feature space. We show that the sparse codes generated using this dictionary perform well in applications of recognition and reconstruction of human actions. Such a model also allows us to represent unseen actions accurately.

Extensions to non linear spaces: Features belonging to non linear spaces such as manifolds have become increasingly popular in the image processing and computer vision communities recently. An interesting extension to the proposed work could be to learn the proposed dictionary model on manifolds. Incorporating the non-linearity of the ambient space will lead to a model robust enough to work with these new features.

Compression of actions: With rising popularity of robots and intelligent surveillance systems, low bandwidth transmission for activities or events could prove to be extremely important. Using the proposed parametric form, extremely high compression ratios could be achieved since only the parameter(s) need to be transmitted as compared to several high dimensional features per action video.

Chapter 7

DYNAMICAL PROPERTIES OF RIEMANNIAN TRAJECTORIES

Dynamic phenomena such as human activities are commonly observed through visual sensors, typically resulting in feature trajectories sampled in time. Accurate metrics on such trajectories are those that take its temporal nature into account. For example, in human action recognition it is well known that accounting for temporal re-parametrization improves the distance metric between two actions, resulting in significantly improved recognition performance [123, 110]. For problems where the elasticity of the metric does not suffice, one has to go a step further and study the properties of the dynamical system that generates the trajectory. A diverse set of applications have benefited from dynamics based metrics such as – human action recognition [6, 141], bio-mechanics [107], dynamics of crowds[5], and dynamic scene recognition [93]. It has also been shown that such properties can help in fine grained classification between similar kinds of human movement [127]. Exploiting the dynamics is relatively easy when the concerned feature space is Euclidean, but the last few years have seen an increased interest in modeling features that lie on non Euclidean spaces such as Riemannian manifolds. Some examples of such features are – shapes on Kendall's shape space [63], Histogram features [31], skeletal features [125], and Covariance features [120]. More recently, there has also been an interest in modeling trajectories on such non-linear manifolds using elastic metrics [110, 13]. However, the study of dynamical invariants has remained unexplored.

In this chapter, we address the problem of uncovering the properties of dynamical processes evolving on Riemannian manifolds, for applications in human action analysis. While the problem of modeling Riemannian trajectories is recent, to the best of the authors' knowledge there has not been a study to exploit Riemannian dynamics in computer vision. In this regard, we propose the largest-Riemannian Lyapunov exponent (L-RLE), which is a

generalization of the largest Euclidean Lyapunov exponent [133], a widely used feature that measures chaos in a time series. Traditional chaotic invariant measures determine the average rate of divergence (or convergence) between nearby states of a system over time. We show that the proposed measure can be used to quantify the amount of chaos within a Riemannian dynamical process. Further, we show that for human action analysis from silhouettes on the Grassmann manifold and curves in SO(3), the representational manifold itself is a good candidate for the *phase space*. Experiments indicate that the L-RLE correlates well with the largest Lyapunov exponent extracted on Euclidean features for human action analysis, while also retaining the robustness advantages and invariances of the manifold-valued features.

More broadly, the chaotic properties of time series data have been found useful in modeling temporal data in several applications. Measures such as the largest Euclidean Lyapunov exponent (L-ELE) allow us to quantify the intrinsic dynamical nature of these phenomenon. Existing methods to compute the L-ELE assume the time series is Euclidean, i.e., the underlying metric is the commonly used ℓ_2 - norm. Whereas many state-of-the-art representations or features involve features that are non-Euclidean. We generalize the notion of a Lyapunov exponent to two different manifolds - the special orthogonal group denoted as $SO(3)$, which is a lie group containing all 3×3 rotation matrices. These occur commonly in data collected from smartphones, and fitness trackers, which measure orientation information. Next, we represent shape silhouettes of humans performing different activities, and model the shapes as affine invariant subspaces, which naturally lie on the Grassmann manifold [119]. This shape representation provides invariance to affine transformations in addition to scale and rotation changes - which is useful in modeling small camera viewpoint changes.

1 Related Work

Recent years have seen advancements in understanding different properties of Riemannian trajectories, motivated by the increasing availability temporal data from videos. For

example, [110] proposed a rate invariant representation known as the Transport Square Root Velocity Function (TSRVF) for trajectories, such that the final metric remains unchanged to identical time warping. Next, [13] showed that the TSRVF could be used to exploit statistical properties of the trajectories to obtain a lower dimensional embedding. Another study [140] models human actions as Riemannian trajectories, by transporting all the points in the trajectory to the starting point, this representation is used to learn a subspace that preserves geodesics. The development of such tools to manipulate, and represent such non-linear trajectories provides a foundation to explore even higher order properties such as dynamics.

While the idea of studying the dynamics of Riemannian trajectories remains to be addressed, the idea of using differential geometry to understand the phase space obtained from Euclidean time series exists [23]. A closely related theoretical piece of work proposed the generalization of the finite time Lyapunov exponent (FTLE) and Lagrangian coherent structures (LCS) to Riemannian manifolds [68]. We differentiate our work by proposing an algorithm to compute the largest Lyapunov exponent, a different measure of chaos than FTLE. We also validate our work on real and synthetic data, with varying degrees of chaos.

Traditional dynamical modeling approaches for Euclidean space data include parametric methods such as Hidden Markov Models (HMMs) and Linear Dynamical Systems (LDSs), which have been used for computer vision applications like action recognition [137, 132] and gait analysis [21, 62]. Recent work by Ali *et al.* proposed the use of nonparametric modeling approach using ideas from chaos theory to model the dynamics in human actions [6]. The authors use Rosenstein's algorithm [88] to estimate largest Lyapunov exponent from trajectories of action data as part of their feature representation. We propose an extension of Rosenstein's algorithm that computes it for Riemannian manifolds.

2 Dynamical Systems On Geometric Spaces

Dynamical systems are mathematical models which simulate a physical phenomenon of states evolving over time. Chaos theory studies the behavior of nonlinear dynamical systems, that are highly sensitive to initial conditions. Any perturbation to the initial conditions of such systems yield widely diverging dynamics. This behavior is known as deterministic chaos. Convincing evidence for existence of deterministic chaos has been provided from a variety of research experiments [90, 112]. Exponential divergence of closely spaced trajectories is a signature of chaotic systems. Hence, quantifying divergence of closely spaced trajectories has been a well-studied problem in the field of chaos theory.

Many natural systems showing chaotic behavior have been studied in the past [54, 94], the most famous one being the weather. A detailed description of such systems was first described mathematically by Lorenz [74]. He presented a system of 3 coupled differential equations to demonstrate the chaotic behavior in such systems. This led him to his now famous speculation that a butterfly flapping wings in Brazil (which is a small change in the initial conditions in the atmosphere) might cause a tornado in Texas. Such dependence of the evolution of a system on its initial conditions makes chaotic motion a complex phenomenon.

Correlation dimension [2], largest Lyapunov exponent [133], and correlation sum [2] are a few examples of invariant measures proposed in the literature to quantify complexity of nonlinear dynamical systems. In comparison, largest Lyapunov exponent is a widely used measure of chaos in various engineering applications, including computer vision and biomechanics to model human movements and quantify chaos in the reconstructed phase space [39, 83, 108, 114, 97].

Due to the inherent variability in human movement, tools from chaos theory have found wide applications in the bio-mechanics community for analysis of human actions [107].

In the most general sense, a dynamical system is the tuple $\langle \mathcal{M}, f, \mathcal{T} \rangle$, where $\mathcal{M}$ is a manifold, $\mathcal{T}$ is non-negative time and f is a diffeomorphism that governs the evolution of

trajectories, defined as $f : \mathcal{M} \times \mathcal{T} \rightarrow \mathcal{M}$. In the Euclidean space, one can learn the parametric representation of the function f. When it is difficult to estimate the function directly, one can estimate properties of the function. One such property is the largest Euclidean Lyapunov exponent (L-ELE), which has seen a lot of success for dynamical analysis of Euclidean signals. There is currently no existing method to estimate f directly for trajectories Riemannian manifolds, therefore we propose an algorithm to first generalize the L-ELE to manifolds.

The largest Lyapunov exponent, denoted as λ, is a measure of average rate of divergence (or convergence) of initially closely-spaced trajectories over time [2, 131]. A positive value indicates orbital divergence and hence chaos in the system. A negative value indicates orbital convergence and hence a dissipative system. A practical method for estimating the largest Lyapunov exponent from a time series proposed by Rosenstein [88] quantifies chaos by monitoring the rate of divergence of closely spaced trajectories over time. The algorithm is fast, easy to implement and robust to changes in embedding dimension, size of dataset, embedding delay and noise level. We refer to the Euclidean space largest Lyapunov exponent as the largest Euclidean Lyapunov exponent (L-ELE) to differentiate it with our measure, the largest Riemannian Lyapunov exponent (L-RLE). More formally, the Lyapunov exponent is defined as follows:

$$d_j(i) = d_j(0)\mathrm{e}^{\lambda_1(i\Delta t)}, \tag{7.1}$$

where $d_j(0)$ is the initial separation in the phase space and $d_j(i)$ is the separation after i time steps of Δt.

2.1 Largest Riemannian Lyapunov Exponent (L-RLE)

The L-ELE is computed as follows [88]: the embedding parameters lag and dimension are estimated using the Fast Fourier Transform (FFT), which are used to construct the phase space. Next, in the phase space the nearest neighbors are calculated constrained on

temporal separation. This is used to estimate how far two points have diverged in the phase space as the attractor evolves over time. In generalizing this to Riemannian manifolds, we first describe how the manifold itself can be treated as the phase space next.

The manifold as a phase space: The phase space is defined as an approximation to the high dimensional state space of the dynamic system that governs the observed time series. Obtaining the phase space directly is challenging because we often do not have access to all the information required to reconstruct it, instead many algorithms resort to reconstructing the phase space. However, reconstruction of the phase space requires estimating the period using the FFT, which do not generalize well to manifolds. On the other hand, action features such as shape silhouettes and stick figures are already high dimensional, and contain a lot of information. For example, the states in a action sequences may be closely related to the poses of the human, which are naturally points on an appropriate high-dimensional shape manifold. Therefore, we propose that the underlying manifold can be treated as the phase space of the system, where each time sample behaves as a "state". We show in our experiments that the manifold behaves similar to the phase space, and therefore is a good approximation.

Computing the L-RLE: In the phase space, the next step involves measuring how far two nearby points have diverged over time. With the geodesic distance, we first perform a k-nearest neighbor and then compute the quantities $d_j(0), d_j(i)$ from (7.1) for a given point and its nearest neighbor. To compute λ from (7.1), it is useful to rearrange as follows.

$$\ln(d_j(i)) \approx \ln(d_j(0)) + \lambda(i\ \Delta t) \tag{7.2}$$

Equation (7.1) represents a set of approximately parallel lines for different points in the phase space. The largest Lyapunov exponent is calculated as the slope of the "average" line. The procedure to estimate the L-RLE is outlined in algorithm 6.

Algorithm 6 Largest Lyapunov exponent on Manifolds

Input: $\alpha(t) \in \mathcal{M}, t = 1 \dots T$
Assume the manifold is the phase space.
for $j = 1 \rightarrow T$ **do**
  Find K Nearest neighbors constrained in time [88].
  for $i = 1 \rightarrow K$ **do**
    $d_j(i) = \min_k d_{\mathcal{M}}(\alpha(j), \alpha(i))$,
    $\ln(d_j(i)) = \ln(d_j(0)) + \lambda_1(i\ \Delta t)$
  end for
  Fit a line, L_j, for each set of d_js, compute its slope m_j.
end for
Average slope gives a robust estimate of L-RLE $\lambda = \frac{1}{T}\sum_j m_j$

3 Experimental Validation

To evaluate the proposed dynamical measure, we apply it to human actions to study their dynamic properties. We use the UMD actions dataset [123] which contains 10 actions such as *walk, run, squat, throw a ball, talk on the cell phone, push an object, and batting.* These are performed 10 times, giving a total of 100 actions in the dataset. The relatively static background allows us to extract the shape silhouettes easily, which we represent as a subspace which is a point on the Grassmann manifold. This results in actions being represented as trajectories on the Grassmann manifold. In our first experiment, we motivate the manifold as the phase space, before computing the RLE. We show quantitative and qualitative results indicating the advantage of the proposed measure. Apart from directly working with the shape trajectories, we consider alternate representations to understand and validate our measure.

Alternate Representations: We first motivate the idea of using the manifold itself as the phase space before computing the Largest Lyapunov exponent. Since the alternative representations are extracted from the same actions in the same dataset, and there is a severe lack of "ground truth", we make the assumption that the dynamics remains unchanged across the features. That is, the dynamics of human actions remains unchanged when represented using features that are Euclidean or non-Euclidean. To the best of our knowledge, we are not aware of any work that can claim otherwise.

Multivariate Embedding (MVE) – We compare with an established algorithm to embed multivariate time series data in the Euclidean space, for time-delay reconstruction of phase space known as the multivariate embedding [25]. This simple yet powerful extension of univariate embedding as proposed by Cao et al. [25] has proven to be useful in computer vision applications such as action synthesis and dynamic texture synthesis [17]. Recent theoretical and empirical findings have demonstrated that multivariate embedding of time series data by simple concatenation of individual univariate embedding vectors achieves good state space reconstruction as evaluated by the shape and dynamics distortion measures [129]. The embedding method only works with Euclidean time series data, and hence we consider the 2D landmarks on the silhouette per frame as our feature for each action. This results in each action being represented as an $(N \times 2) \times T$, where N is the number of landmarks on each silhouette, and T is the total number of frames. Using this data, we perform uniform multivariate embedding. Given multivariate time series data $\{x_{i,t}\}_{t=1}^{T}$, $i = 1, \ldots, p$, where p is the dimension of time series data, the reconstructed phase space vector is of the form

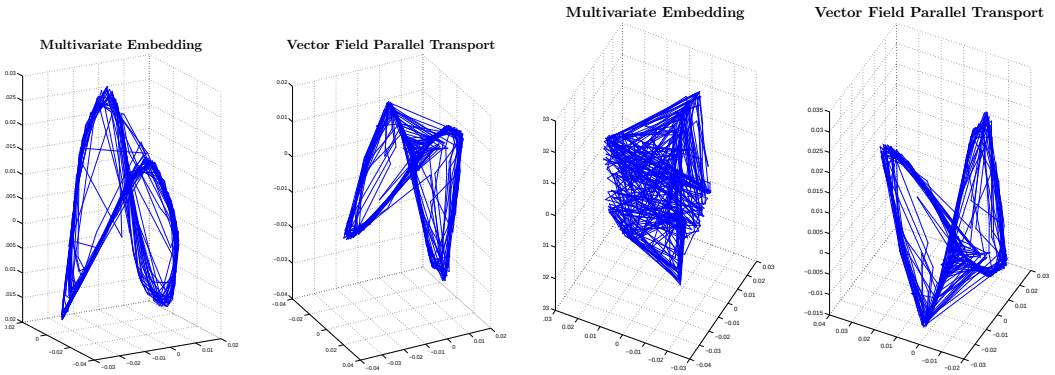
$$\begin{aligned} \mathbf{z}_t = [&x_{1,t}, x_{1,t+\tau_1}, \ldots, x_{1,t+(m_1-1)\tau_1}, \\ &x_{2,t}, x_{2,t+\tau_2}, \ldots, x_{2,t+(m_2-1)\tau_2}, \\ &\ldots, \\ &x_{p,t}, x_{p,t+\tau_p}, \ldots, x_{p,t+(m_p-1)\tau_p}]. \end{aligned} \tag{7.3}$$

where m_i and τ_i are respectively the embedding dimension and time delay for each of the p-dimension in the multivariate time series data.

Vector Field Parallel Transport (VFPT) – We also use an intermediate representation, where we represent each trajectory as a collection of tangents. For an action j, $\mathcal{F}_j = \{\dot{\alpha}_{t\to t+1}(t) | \forall i = 1, \ldots, T\}$, where $\dot{\alpha}_{t\to t+1}(t)$ represents the tangent that goes from $\alpha(t)$ to $\alpha(t+1)$ in unit time. We perform a parallel transport on all the tangent vectors and bring them to a common point at the Riemannian center of mass (RCM) [49]. We treat the transported tangent bundle as the phase space in this case. This feature takes the geometry into account while also giving us a Euclidean representation, which we can exploit for visualizing the phase space. We use the largest Euclidean lyapunov exponent

(L-ELE) algorithm [88] on this feature.

Viewpoint invariance: Figure 1a shows the phase space for the walking action, in $3D$, after performing dimensionality reduction using Laplacian Eigenmaps [18]. It is seen that in both cases, the cyclic pattern of the action is captured even after dimensionality reduction. Since the Grassmann manifold can afford us affine invariance, we artificially shear the shapes to simulate minor camera viewpoint changes. Since the multivariate embedding uses the coordinate locations in each frame, the phase space estimated from the sheared data is significantly distorted. The phase space obtained from the Grassmann representation remains unchanged, as shown in figures 1b.



(a) Phase space estimates on shape data.

(b) Phase space estimates after change in viewpoint.

Figure 1: The Grassmann manifold for shapes, as the phase space for human activities provides invariance to commonly observed problems such as viewpoint, scale and shift. The resulting phase space is significantly distorted in the case of multivariate embedding, but remains unchanged in our case. We perform dimensionality reduction to facilitate easy visualization using the Laplacian-eigen maps [18].

We estimate the L-RLE on 100 actions in the UMD Actions dataset and report the correlation between our measure and the Lyapunov from Multivariate Embedding on Euclidean features in table 1. It is seen that our measure compares well with the vector space version – indicating that for clean data the L-RLE is a good generalization of the L-ELE. Further, when we artificially shear the data to simulate minor viewpoint changes, the multivariate embedding algorithm and L-ELE algorithm fail severely due to the distortion in the data

Phase space type	Correlation
Landmarks + Multivariate Embedding + L-ELE [88] (reference)	1.00
Vector Field Parallel Transport + L-ELE	0.52
Shape feature + L-RLE	**0.76**
Landmarks + Multivariate Embedding + L-ELE [88] (sheared)	0.27
Vector Field Parallel Transport + L-ELE (sheared)	0.40
Shape feature + L-RLE (sheared)	**0.763**

Table 1: The proposed Riemannian Lyapunov exponent (RLE) on the Grassmann manifold closely relates to the estimate obtained from the multivariate embedding on the landmarks of the silhouette. It is also much more robust to affine transforms, compared to the Euclidean measure. Here we assume the standard largest Euclidean Lyapunov exponent (L-ELE) without any shearing to be the reference standard.

whereas the L-RLE remains robust to such changes.

3.1 Validation On Standard Attractors

A challenging aspect to generalizing the largest Lyapunov exponent to Riemannian manifolds is validation. In the Euclidean space, a common way to evaluate a chaotic measure is to test it on different attractors arising from closed-form dynamical equations such as the Lorenz [74] and the Rossler [89] systems. Any chaotic measure must be as close to zero as possible for perfectly periodic time series. Unfortunately, the Rossler and Lorenz systems do not generalize easily to manifolds. We approximate these systems by generating them on a tangent plane and wrapping them onto the manifold. Once again, we assume that the wrapping action from the tangent space to the manifold does not affect the chaotic nature of the time series. This maybe a restricting assumption in general, but is valid for small deviations from the pole of the tangent space. We choose the special orthogonal group SO(3), which is a Lie group, since it is 3-dimensional and allows us to naturally embed a 3-D time series generated on its Lie algebra. The properties of the Largest Euclidean Lyapunov Exponent (L-ELE) which we expect to observe here are the following: 1) The value for periodic signals must be zero, 2) The L-ELE is direct measure of the chaotic nature

of the signal, i.e. the higher the chaos within a signal, the higher its measured value. This pattern is clearly observed in table 2, where a periodic signal gives us a value thats close to zero. We also compare the L-ELE values as a reference, it is observed that even though the values are not exactly the same, the trend is clear. A trajectory thats more chaotic has a higher L-RLE value, similar to the Euclidean case.

Attractor	L-Euclidean LE	L-Riemannian LE
Lorenz [74] (higher chaos)	1.50	21.02
Rossler [89](lower chaos)	0.09	7.26
Periodic (zero chaos)	0	0.008

Table 2: **Validating the L-RLE:** We embed the standard attractors into the SO(3) lie group and evaluate the L-RLE using the proposed algorithm. It is seen that the nature of the L-RLE is consistent with the L-ELE, higher chaos implies a higher value, and periodicity implies a very low value.

4 Discussion And Conclusion

We presented a formulation to study the invariant properties of dynamical systems evolving on Riemannian manifolds. Such systems occur frequently in problems such as human movement analysis, action recognition, and crowd analysis in computer vision. The invariant properties of Euclidean dynamical systems have been useful in characterizing temporal events for such applications in computer vision. However, there is a lack of such methods for dynamical systems on non-Euclidean spaces. To address this, we proposed a generalization of the largest Lyapunov exponent, a classic chaotic measure, to Riemannian manifolds. Towards this end, we use the ambient manifold as the phase space and compute the largest Riemannian Lyapunov exponent (L-RLE). We show that it correlates well with the analogous measure for Euclidean dynamics. By estimating the L-RLE on standard attractors such as the Lorenz and Rossler, we show that our L-RLE measures the chaotic properties accurately. We have validated the L-RLE under the assumption that the dynamical latent properties of temporal events remains unchanged when observed in different feature spaces. A direction of future study could be to further investigate how the dynamical properties

are preserved when the same event is observed in different modalities. While the presented work is primarily empirical, a theoretical analysis of Riemannian dynamical invariants and associated estimation algorithms, such as the proposed one, may be fruitful areas of future work.

Chapter 8

DIRECTIONS FOR FUTURE WORK

This dissertation presented tools and techniques to model Riemannian trajectories for applications in human movement analysis.

1 Potential Future Research Directions

In this section, a plan for the extensions of the works discussed in this dissertation and possible future directions of work is laid out.

1.1 Generalized Symbolic Approximation

The framework for symbolic approximation is general enough to deal with more abstract forms of information such as graphs [59] or bag-of-words [45]. In fact, any system that is sequential can be used within this framework, the key is to have a good understanding of metrics on these abstract models. A useful extension to further improve the compression efficiency would be to utilize symbols that are sequences themselves. In this work, discretization of a manifold sequence is preceded by a Piece-wise Aggregate Approximation (PAA) step, which collapses a series of points into a single point using the Riemannian Center of Mass (RCM) [49], which is then assigned to the nearest pre-learned quantization level. Instead, one can imagine eliminating the PAA step entirely, without sacrificing the compression ratios, by learning a symbol set where each symbol is a short sequence. With tools such as the TSRVF which allow us to compare Riemannian sequences in a speed-invariant manner, such a symbol set can be learned using the competitive learning strategy proposed in this work or any other clustering scheme. However, there will be a computational trade-off since each short windowed sequence will now have to be warped before finding its true nearest neighbor.

1.2 Sampling Techniques On Manifolds

The online sampling algorithm proposed in chapter 4, can be easily generlalized to work with manifold valued data. Sampling is interesting in itself for a variety of problems like computer graphics and machine learning. It is worth studying connections between a data-driven sampling apprach such as the one proposed here, compared to directly sampling in the 'feature space', which is the norm in computer graphics. Sampling can also add to the rapidly growing set of tools that generalize machine learning to Riemannian manifolds. Active learning uses sampling as a key step in picking the *best* training set.

1.3 Topology Meets Riemannian Geometry

In the recent few years topological data analysis (TDA) has become a useful tool to visualize and understand properties of high dimensional data. A natural progression for these tools are to work with non Euclidean data – i.e., exploit topological properties of datasets that lie on Riemannian manifolds. The chapter on dynamical analysis 7 for Riemannian trajectories introduces notions of TDA implicitly by computing the Lyapunov exponent, which is a topological feature.

Further, recent advances in quantifying topological properties of high dimensional data may benefit from the tools developed in Riemannian geometry. The number of d-dimensional holes are known as the Betti-d number. It corresponds to the rank of the d-dimensional homology group. Persistent diagrams are a powerful new feature to represent the persistence of Betti numbers across multiple scales of the data. However, metrics on persistence diagrams tend to become a computational bottleneck because they need to solve for correspondence between points. A Riemannian geometric interpretation for persistent diagrams could help in addressing these issues.

1.4 Dynamic Invariants

An interesting assumption in chapter 7 is the invariance of dynamical properties across feature spaces. In other words – if a temporal phenomenon is measured in different feature spaces with similar degrees of freedom, can we learn the same dynamical system or estimate similar properties of the dynamical system across both of them? For example consider gesture recognition, measured using the Microsoft Kinect depth sensor and a fitness device. Assuming we are interested in single handed gestures that do not involve hand signs, both the sensors are essentially observing and measuring the same information. Therefore a reasonable question is if we can estimate the same Lyapunov feature from both of the feature spaces independently, even though they live in different feature spaces. This can be of great use in multi-model feature analysis, and inference problems involving multiple sensors.

REFERENCES

[1] VSUMM(video summarization). https://sites.google.com/site/vsummsite/download, 2011.

[2] Henry DI Abarbanel. *Analysis of observed chaotic data*. New York: Springer-Verlag, 1996.

[3] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80(2):199–220, 2004.

[4] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, November 2006.

[5] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. *CVPR*, pages 1–6, 2007.

[6] Saad Ali, Arslan Basharat, and Mubarak Shah. Chaotic invariants for human action recognition. In *ICCV*, pages 1–8, 2007.

[7] Cyril Allauzen and Mathieu Raffinot. Simple optimal string matching algorithm. In *Combinatorial Pattern Matching*, volume 1848 of *Lecture Notes in Computer Science*, pages 364–374. Springer Berlin Heidelberg, 2000.

[8] Jurandy Almeida, Neucimar J Leite, and Ricardo da S Torres. Vison: Video summarization for online applications. *Pattern Recognition Letters*, 33(4):397–409, 2012.

[9] R. Anirudh and P. Turaga. Geometry-based symbolic approximation for fast sequence matching on manifolds. *International Journal of Computer Vision*, pages 1–13, 2015.

[10] Rushil Anirudh, Ahnaf Masroor, and Pavan Turaga. Diversity promoting online sampling for streaming video summarization. *Manuscript*, 2016.

[11] Rushil Anirudh, Karthikeyan Natesan Ramamurthy, Jayaraman J. Thiagarajan, Pavan K. Turaga, and Andreas Spanias. A heterogeneous dictionary model for representation and recognition of human actions. In *ICASSP*, 2013.

[12] Rushil Anirudh and Pavan Turaga. Interactively test driving an object detector: Estimating performance on unlabeled data. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 175–182. IEEE, 2014.

[13] Rushil Anirudh, Pavan Turaga, Jingyong Su, and Anuj Srivastava. Elasting functional coding of human actions: from vector fields to latent variables. In *CVPR*, pages 3147–3155, 2015.

[14] Rushil Anirudh, Pavan Turaga, Jingyong Su, and Anuj Srivastava. Elastic functional coding of Riemannaian trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*, Accepted April 2016.

[15] Rushil Anirudh, Vinay Venkataraman, and Pavan Turaga. A generalized lyapunov feature for dynamical systems on riemannian manifolds. In *BMVC Workshops, 1st International Workshop on Differential Geometry in Computer Vision (Diff-CV)*, 2015.

[16] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[17] Arslan Basharat and Mubarak Shah. Time series prediction by chaotic modeling of nonlinear dynamical systems. In *IEEE International Conference on Computer Vision*, pages 1941–1948, 2009.

[18] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

[19] Donald Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[20] Armin Biess, Dario G Liebermann, and Tamar Flash. A computational model for redundant human three-dimensional pointing movements: integration of independent spatial and temporal motor plans simplifies movement dynamics. *The Journal of Neuroscience*, 27(48):13045–13064, 2007.

[21] Alessandro Bissacco, Alessandro Chiuso, Yi Ma, and Stefano Soatto. Recognition of human gaits. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 52–57, 2001.

[22] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry. Revised 2nd Ed.* Academic, New York, 2003.

[23] Thomas J Bridges and Sebastian Reich. Computing lyapunov exponents on a stiefel manifold. *Physica D: Nonlinear Phenomena*, 156(3):219–238, 2001.

[24] Benno Büeler, Andreas Enge, and Komei Fukuda. Exact volume computation for polytopes: a practical study. In *Polytopes combinatorics and computation*, pages 131–154. Springer, 2000.

[25] Liangyue Cao, Alistair Mees, and Kevin Judd. Dynamics from multivariate time series. *Physica D: Nonlinear Phenomena*, 121(1):75–88, 1998.

[26] Hasan Ertan Çetingül and René Vidal. Intrinsic mean shift for clustering on stiefel and grassmann manifolds. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Miami, Florida, USA*, pages 1896–1902, 2009.

[27] Kaushik Chakrabarti, Eamonn J. Keogh, Sharad Mehrotra, and Michael J. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Trans. Database Syst.*, 27(2):188–228, 2002.

[28] S. Chakraborty, O. Tickoo, and R. Iyer. Adaptive keyframe selection for video summarization. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 702–709, Jan 2015.

[29] A.B. Chan and N. Vasconcelos. Classification and retrieval of traffic video using auto-regressive stochastic processes. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 771–776, June 2005.

[30] R. Chaudhry and Y. Ivanov. Fast approximate nearest neighbor methods for non-Euclidean manifolds with applications to human activity analysis in videos. In *European Conference on Computer Vision*, Crete, Greece, September 2010.

[31] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR, 2009.*, pages 1932 –1939, June 2009.

[32] Yinpeng Chen, Margaret Duff, Nicole Lehrer, Hari Sundaram, Jiping He, Steven L Wolf, Thanassis Rikakis, Tuan D Pham, Xiaobo Zhou, Hiroshi Tanaka, et al. A computational framework for quantitative evaluation of movement during rehabilitation. In *AIP Conference Proceedings-American Institute of Physics*, volume 1371, page 317, 2011.

[33] Ondrej Chum, Michal Perdoch, and Jiri Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In *CVPR*, pages 17–24, 2009.

[34] Zachary Danziger and Ferdinando A Mussa-Ivaldi. The influence of visual motion on motor learning. *The Journal of Neuroscience*, 32(29):9859–9869, 2012.

[35] Sandra Eliza Fontes de Avila, Ana Paula Brando Lopes, Antonio da Luz Jr., and Arnaldo de Albuquerque Arajo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56 – 68, 2011.

[36] D. Desieno. Adding a conscience to competitive learning. *IEEE International Conference on Neural Networks*, 1:117–124, 1988.

[37] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo. 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. *Cybernetics, IEEE Transactions on*, PP(99):1–1, September 2014.

[38] L. Devroye, W. Szpankowski, and B. Rais. A note on the height of suffix trees. *SIAM Journal on Computing*, 21(1):48–53, 1992.

[39] Jonathan B Dingwell and Joseph P Cusumano. Nonlinear time series analysis of normal and pathological human walking. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 10(4):848–863, 2000.

[40] Xiaomin Duan, Huafei Sun, and Linyu Peng. Riemannian means on special euclidean group and unipotent matrices group. *The Scientific World Journal*, 2013, 2013.

[41] Ahmed Elgammal and Chan-Su Lee. Nonlinear manifold learning for dynamic shape and dynamic appearance. *Comput. Vis. Image Underst.*, 106(1):31–46, April 2007.

[42] Troje N. F. Decomposing biological motion : A framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2:371–387, 2002.

[43] M. Faraki, M.T. Harandi, and F. Porikli. More about vlad: A leap from euclidean to riemannian manifolds. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 4951–4960, June 2015.

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

[45] Utkarsh Gaur, Yingying Zhu, Bi Song, and Amit K. Roy Chowdhury. A "string of feature graphs" model for recognition of complex activities in natural videos. In *ICCV*, pages 2595–2602, 2011.

[46] Joydeep Ghosh, Yong Jae Lee, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1353. IEEE, 2012.

[47] C. R. Goodall and K. V. Mardia. Projective shape analysis. *Journal of Computational and Graphical Statistics*, 8(2), 1999.

[48] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2247 –2253, dec. 2007.

[49] K. Grove and H. Karcher. How to conjugate C^1-close group actions. *Math.Z*, 132:11–20, 1973.

[50] T. Guha and R.K. Ward. Learning sparse representations for human action recognition. *IEEE Transactions on PAMI*, 34(8):1576 –1588, aug. 2012.

[51] Mehrtash Tafazzoli Harandi, Mathieu Salzmann, and Richard Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices. In *ECCV*, pages 17–32, 2014.

[52] Mehrtash Tafazzoli Harandi, Conrad Sanderson, Richard I. Hartley, and Brian C. Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II*, pages 216–229, 2012.

[53] Mehrtash Tafazzoli Harandi, Conrad Sanderson, Chunhua Shen, and Brian C. Lovell. Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 3120–3127, 2013.

[54] Alan Hastings and Thomas Powell. Chaos in a three-species food chain. *Ecology*, pages 896–903, 1991.

[55] Zhaoshui He, Andrzej Cichocki, Yuanqing Li, Shengli Xie, and Saeid Sanei. K-hyperline clustering learning for sparse component analysis. *Signal Processing*, 89(6):1011–1022, 2009.

[56] Zhaoshui He, Andrzej Cichocki, Yuanqing Li, Shengli Xie, and Saeid Sanei. K-hyperline clustering learning for sparse component analysis. *Signal Process.*, 89(6):1011–1022, June 2009.

[57] Jeffrey Ho, Yuchen Xie, and Baba C. Vemuri. On A nonlinear generalization of sparse coding and dictionary learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1480–1488, 2013.

[58] Zhuolin Jiang, Zhe Lin, and L.S. Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2651–2664, Nov 2013.

[59] Michael I. Jordan. *Learning in Graphical Models*. Cambridge, MA: MIT Press, 1998.

[60] Shantanu H. Joshi, Eric Klassen, Anuj Srivastava, and Ian Jermyn. A novel representation for Riemannian analysis of elastic curves in R^{n}. In *CVPR*, 2007.

[61] Jürgen Jost. *Riemannian Geometry and Geometric Analysis (6. ed.)*. Springer, 2011.

[62] Amit Kale, Aravind Sundaresan, AN Rajagopalan, Naresh P Cuntoor, Amit K Roy-Chowdhury, Volker Kruger, and Rama Chellappa. Identification of humans using gait. *IEEE Transactions on Image Processing*, 13(9):1163–1173, 2004.

[63] D.G. Kendall. Shape manifolds, Procrustean metrics and complex projective spaces. *Bulletin of London Mathematical society*, 16:81–121, 1984.

[64] Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. Large-scale video summarization using web-image priors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2698–2705. IEEE, 2013.

[65] T. Kohonen. *Self-Organizing Maps*. Berlin: Springer - Verlag., 1995.

[66] John D. Lafferty and Guy Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6:129–163, 2005.

[67] Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in neural information processing systems*, 16:329–336, 2004.

[68] Francois Lekien and Shane D Ross. The computation of finite-time lyapunov exponents on unstructured meshes and for non-euclidean manifolds. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(1):017505, 2010.

[69] M. Lewandowski, D. Makris, S.A. Velastin, and J.-C. Nebel. Structural laplacian eigenmaps for modeling sets of multivariate sequences. *Cybernetics, IEEE Transactions on*, 44(6):936–949, June 2014.

[70] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–14, Jun. 2010.

[71] Jessica Lin, Eamonn J. Keogh, Stefano Lonardi, and Bill Yuan chi Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *DMKD*, pages 2–11, 2003.

[72] Jessica Lin and Yuan Li. Finding approximate frequent patterns in streaming medical data. In *CBMS*, pages 13–18, 2010.

[73] T. Lin and H. Zha. Riemannian manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:796–809, 2008.

[74] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[75] Yui Man Lui, J. Ross Beveridge, and Michael Kirby. Action classification on product manifolds. In *CVPR*, pages 833–839, 2010.

[76] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *NIPS*, 2008.

[77] Abdullah Mueen, Eamonn J. Keogh, Qiang Zhu, Sydney Cash, and M. Brandon Westover. Exact discovery of time series motifs. In *SDM*, pages 473–484, 2009.

[78] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[79] Omar Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *(CVPR), 2013*, pages 716–723. IEEE, 2013.

[80] P. Patel, E. Keogh, J. Lin, and S. Lonardi. Mining motifs in massive time series databases. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 370–377, 2002.

[81] X. Pennec. Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154, 2006.

[82] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

[83] Matjaž Perc. The dynamics of human gait. *European journal of physics*, 26(3):525–534, 2005.

[84] Qiang Qiu, Zhuolin Jiang, and R. Chellappa. Sparse dictionary-based representation and recognition of action attributes. In *ICCV*, pages 707 –714, nov. 2011.

[85] Inam Ur Rahman, Iddo Drori, Victoria C. Stodden, David L. Donoho, and Peter Schrder. Multiscale representations for manifold-valued data. *SIAM J. MULTISCALE MODEL. SIMUL*, 4(4):1201–1232, 2005.

[86] Jérome Revaud, Matthijs Douze, Cordelia Schmid, and Herve Jegou. Event retrieval in large video collections with circulant temporal encoding. In *CVPR*, pages 2459–2466, 2013.

[87] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press., 1996.

[88] M.T. Rosenstein, J.J. Collins, and C.J. De Luca. A practical method for calculating largest lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, 65(1):117–134, 1993.

[89] Otto E Rössler. An equation for continuous chaos. *Physics Letters A*, 57(5):397–398, 1976.

[90] J-C Roux, Reuben H Simoyi, and Harry L Swinney. Observation of a strange attractor. *Physica D: Nonlinear Phenomena*, 8(1):257–266, 1983.

[91] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000.

[92] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[93] Aswin C. Sankaranarayanan, Pavan K. Turaga, Richard G. Baraniuk, and Rama Chellappa. Compressive acquisition of dynamic scenes. In *ECCV (1)*, pages 129–142, 2010.

[94] William M Schaffer. Order and chaos in ecological systems. *Ecology*, pages 93–106, 1985.

[95] Lorenzo Seidenari, Vincenzo Varano, Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2013*, pages 479–485, 2013.

[96] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[97] N. Shroff, P. Turaga, and R. Chellappa. Moving vistas: Exploiting motion for describing scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1911–1918, June 2010.

[98] Nitesh Shroff, Pavan K. Turaga, and Rama Chellappa. Video précis: Highlighting diverse aspects of videos. *IEEE Transactions on Multimedia*, 12(8):853–868, 2010.

[99] Nitesh Shroff, Pavan K. Turaga, and Rama Chellappa. Manifold precis: An annealing technique for diverse sampling of manifolds. In *NIPS*, 2011.

[100] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[101] S. Soatto, G. Doretto, and Y. N. Wu. Dynamic textures. *ICCV*, 2:439–446, 2001.

[102] Michael Spivak. *A Comprehensive Introduction to Differential Geometry*, volume One. Publish or Perish, Inc., Houston, Texas, third edition, 1999.

[103] A. Srivasatava and E. Klassen. Bayesian geometric subspace tracking. *Advances in Applied Probability*, 36(1):43–56, March 2004.

[104] A. Srivastava, I. Jermyn, and S. Joshi. Riemannian analysis of probability density functions with applications in vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[105] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning, and testing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(4), 2005.

[106] Anuj Srivastava, Eric Klassen, Shantanu H. Joshi, and Ian H. Jermyn. Shape analysis of elastic curves in Euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1415–1428, 2011.

[107] Nicholas Stergiou. *Innovative Analyses of Human Movement*. Human Kinetics, first edition, 2003.

[108] Nicholas Stergiou and Leslie M Decker. Human movement variability, nonlinear dynamics, and pathology: is there a connection? *Human Movement Science*, 30(5):869–888, 2011.

[109] Jingyong Su, Ian L. Dryden, Eric Klassen, Huiling Le, and Anuj Srivastava. Fitting smoothing splines to time-indexed, noisy points on nonlinear manifolds. *Image Vision Comput.*, 30(6-7):428–442, 2012.

[110] Jingyong Su, Sebastian Kurtek, Eric Klassen, and Anuj Srivastava. Statistical analysis of trajectories on Riemannian manifolds: Bird migration, hurricane tracking, and video surveillance. *Annals of Applied Statistics*, 8(1), 2014.

[111] Jingyong Su, Anuj Srivastava, Fillipe D. M. de Souza, and Sudeep Sarkar. Rate-invariant analysis of trajectories on riemannian manifolds with application in visual speech recognition. In *CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 620–627, 2014.

[112] Harry L Swinney. Observations of order and chaos in nonlinear systems. *Physica D: Nonlinear Phenomena*, 7(1):3–15, 1983.

[113] Sima Taheri, Pavan K. Turaga, and Rama Chellappa. Towards view-invariant expression analysis using analytic shape manifolds. In *FG*, pages 306–313, 2011.

[114] TM TenBroek, REA Van Emmerik, CJ Hasson, and J. Hamill. Lyapunov exponent estimation for human gait acceleration signals. *Journal of Biomechanics*, 40(2):210, 2007.

[115] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[116] J.J. Thiagarajan, K.N. Ramamurthy, and A. Spanias. Optimality and stability of the K-hyperline clustering algorithm. *Pattern Recognition Letters*, 32(9):1299–1304, 2011.

[117] Pavan Turaga, Ashok Veeraraghavan, Anuj Srivastava, and Rama Chellappa. Statistical analysis on manifolds and its applications to video analysis. In Dan Schonfeld, Caifeng Shan, Dacheng Tao, and Liang Wang, editors, *Video Search and Mining*, volume 287 of *Studies in Computational Intelligence*, pages 115–144. Springer Berlin Heidelberg, 2010.

[118] Pavan K. Turaga and Rama Chellappa. Locally time-invariant models of human activities using trajectories on the Grassmannian. In *CVPR*, pages 2435–2441, 2009.

[119] Pavan K. Turaga, Ashok Veeraraghavan, Anuj Srivastava, and Rama Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011.

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

[121] Alireza Vahdatpour, Navid Amini, and Majid Sarrafzadeh. Toward unsupervised activity discovery using multi-dimensional motif detection in time series. In *IJCAI*, pages 1261–1266, 2009.

[122] A. Vedaldi and K. Lenc. MatConvNet – convolutional neural networks for MATLAB. In *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[123] A. Veeraraghavan, R. Chellappa, and A. K. Roy-Chowdhury. The function space of an activity. *IEEE CVPR*, pages 959–968, 2006.

[124] Ashok Veeraraghavan, Amit K. Roy Chowdhury, and Rama Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1896–1909, 2005.

[125] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *(CVPR), 2014*, pages 588–595, June 2014.

[126] Vinay Venkataraman, Pavan Turaga, Nicole Lehrer, Michael Baran, Thanassis Rikakis, and Steven L Wolf. Attractor-shape for dynamical analysis of human movement: Applications in stroke rehabilitation and action recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 514–520. IEEE, 2013.

[127] Vinay Venkataraman, Pavan K. Turaga, Nicole Lehrer, Michael Baran, Thanassis Rikakis, and Steven L. Wolf. Attractor-shape for dynamical analysis of human movement: Applications in stroke rehabilitation and action recognition. In *CVPR Workshops*, 2013.

[128] S. V. N. Vishwanathan, Karsten M. Borgwardt, Imre Risi Kondor, and Nicol N. Schraudolph. Graph kernels. *CoRR*, abs/0807.0093, 2008.

[129] I Vlachos and D Kugiumtzis. State space reconstruction from multiple time series. In *Topics on Chaotic Systems: Selected Papers from Chaos 2008 International Conference*, page 378. World Scientific, 2009.

[130] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *(CVPR), 2012*, pages 1290–1297, June 2012.

[131] Garnett P Williams. *Chaos theory tamed*. Joseph Henry Press, 1997.

[132] Andrew David Wilson and Aaron F Bobick. Learning visual behavior for gesture analysis. In *IEEE International Symposium on Computer Vision*, pages 229–234, Nov. 1995.

[133] Alan Wolf, Jack B Swift, Harry L Swinney, and John A Vastano. Determining lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, 16(3):285–317, 1985.

[134] Steven L Wolf, Pamela A Catlin, Michael Ellis, Audrey Link Archer, Bryn Morgan, and Aimee Piacentino. Assessing wolf motor function test as outcome measure for research in patients after stroke. *Stroke*, 32(7):1635–1639, 2001.

[135] Wright, J. and Yang, A.Y. and Ganesh, A. and Sastry, S.S. and Yi Ma. Robust face recognition via sparse representation. *IEEE Trans. on PAMI*, 31(2):210–227, 2001.

[136] L. Xia, C.C. Chen, and JK Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW)2012*, pages 20–27. IEEE, 2012.

[137] Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 379–385, June 1992.

[138] Dong-Ming Yan, Jian-Wei Guo, Bin Wang, Xiao-Peng Zhang, and Peter Wonka. A survey of blue-noise sampling and its applications. *Journal of Computer Science and Technology*, 30(3):439–452, 2015.

[139] A. Yao. The complexity of pattern matching for a random string. *SIAM Journal on Computing*, 8(3):368–387, 1979.

[140] S. Yi and H. Krim. A subspace learning of dynamics on a shape manifold: A generative modeling approach. In *Geometric Science of Information - First International Conference, GSI*, 2013.

[141] Sheng Yi, Hamid Krim, and Larry K. Norris. Human activity as a manifold-valued random process. *IEEE Transactions on Image Processing*, 21(8):3416–3428, 2012.

[142] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[143] P. Zador. Asymptotic quantization error of continuous signals and the quantization dimension. *Information Theory, IEEE Transactions on*, 28(2):139 – 149, mar 1982.

[144] Zhengwu Zhang, Jingyong Su, Eric Klassen, Huiling Le, and Anuj Srivastava. Video-based action recognition using rate-invariant analysis of covariance trajectories. *CoRR*, abs/1503.06699, 2015.

[145] Guoying Zhao, Mark Barnard, and Matti Pietikäinen. Lipreading with local spatiotemporal descriptors. *IEEE Transactions on Multimedia*, 11(7):1254–1265, 2009.

[146] Feng Zhou and Fernando De la Torre. Generalized time warping for multi-modal alignment of human motion. In *(CVPR), 2012*, pages 1282–1289. IEEE, 2012.