Perturbation Robust Representations of Topological Persistence Diagrams

by

Kowshik Thopalli

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2017 by the
Graduate Supervisory Committee:

Pavan Turaga, Chair
Antonia Suppappola
Yezhou Yang

ARIZONA STATE UNIVERSITY

December 2017

ABSTRACT

Topological methods for data analysis present opportunities for enforcing certain invariances of broad interest in computer vision: including view-point in activity analysis, articulation in shape analysis, and measurement invariance in non-linear dynamical modeling. The increasing success of these methods is attributed to the complementary information that topology provides, as well as availability of tools for computing topological summaries such as persistence diagrams. However, persistence diagrams are multi-sets of points and hence it is not straightforward to fuse them with features used for contemporary machine learning tools like deep-nets. In this paper theoretically well-grounded approaches to develop novel perturbation robust topological representations are presented, with the long-term view of making them amenable to fusion with contemporary learning architectures. The proposed representation lives on a Grassmann manifold and hence can be efficiently used in machine learning pipelines. The proposed representation.The efficacy of the proposed descriptor was explored on three applications: view-invariant activity analysis, 3D shape analysis, and non-linear dynamical modeling. Favorable results in both high-level recognition performance and improved performance in reduction of time-complexity when compared to other baseline methods are obtained.

ACKNOWLEDGMENTS

I should first thank Professor Pavan Turaga, for giving me an wonderful opportunity work in his lab. He has been supportive all through out. I should also thank him for his patience for putting up with my ignorance in most areas.

I would like to thank my collaborators and fellow members of the lab including Suhas, Qiao Wang for sharing their valuable inputs. I will have to thank the alumni of this group, Mr.Rushil,Vinay and Kuldeep for always being there to answer my questions.I would also like to thank Karthikeyan Natesan Ramamaurthy for spending quality time with us and guiding us through this work. A special mention to Anirudh som for explaining the many intricacies involved in collaborative works like these ans also his inputs on various technical fronts.

I am also thankful to the many friendships that I made in this phase which helped me stay focused on the job. I am thankful to my graduate advisor, Sno Kleespies from the department of Electrical Computer and Energy Engineering at ASU, for helping me make my stay at ASU, a comfortable one.

This thesis would not have been possible without the love, guidance and unconditional support of my parents,sister and brother-in-law

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Over the years, tools from topological data analysis (TDA) have been used to characterize the topological structure of data, where the data is perceived as a noisy sampling from an underlying space. One of the prominent TDA tools is persistent homology. It provides a multi-scale summary of different homological features (Carlsson *et al.* (2005)). This multi-scale information is represented using a persistence diagram (PD), a 2-dimensional (2D) Cartesian plane with a collection of points. For a point (b, d) in the PD, a homological feature appears at scale b and disappears at scale d. Due to the simplicity of PDs, there has been a surge of interest to use persistent homology for summarizing high-dimensional complex data and has resulted in successful implementation in several research areas (Anirudh *et al.* (2016)). However, application of machine learning techniques on the space of PDs has always been a challenging task. The gold-standard approach for measuring the '*distance*' between PDs is the *bottleneck* or the L_p-*Wasserstein* metric (Edelsbrunner *et al.* (2002)). These distance measures make it possible to do clustering of PDs. However, a simple metric structure is not enough to use other machine learning tools such as support vector machines (SVMs), neural networks, random forest decision trees, principal component analysis and so on. Also, these metrics are only stable under small perturbations of the data which the PDs summarize, and the complexity of computing distances between PDs grows in the order of $\mathcal{O}(n^2)$, where n refers to the number of points in the PD. Several efforts have been made to overcome these problems by attempting to map PDs to spaces that are more suitable for machine learning tools (Adams *et al.* (2017)). Keeping the above issues in mind, one would prefer a suitable representation of the PDs that has the following desired properties -

1. Computationally efficient.
2. Can be easily integrated with machine learning tools.
3. Is stable with respect to topological noise.
4. Interpretable relationship to the PD from which it was computed.

Using a novel perturbation framework, in this paper we propose a topological noise robust representation of PDs that attempts to incorporate the above traits. We first generate a set of perturbed PDs by randomly shifting the points in the original PD by a certain level. A perturbed PD is analogous to extracting the persistence diagram from data that is subjected to topological noise. Next we construct a 2D probability distribution function (PDF) using kernel density estimation on each of the perturbed PDs. Finally, we simplify the end representation-space of the set of 2D PDFs to be a Grassmannian, which is a non-constantly curved manifold. We can develop very efficient machine learning pipelines over the topological descriptors by leveraging known metrics and statistical results on the Grassmann manifold space. Our experiments suggest that our proposed framework recovers the lost performance due to functional methods, while still enjoying orders of magnitude faster processing times over the classical Wasserstein and bottleneck approaches.

Outline of thesis: Section 1.1 discusses related work in more detail. Chapter 2 and Chapter 3 provides the necessary background on Topological Data Analysis, persistent homology, scalar-field topology and the Grassmannian. Chapter 4 gives a detailed description of the proposed framework and end representation of the PD for statistical learning tasks. Section Chapter 5 describes the experiments and results. Chapter 6 concludes the thesis.

1.1 Related Work

In persistent homology, PDs provide a compact multi-scale summary of the occurrence of different topological features. The traditional metrics used to compare the (dis)similarity between PDs are the *bottleneck* and the L_p-*Wasserstein* metrics. These measures are sta-

ble with respect to small continuous deformations of the topology of the inputs, which is the main reason for their popularity (Cohen-Steiner *et al.* (2005)). For a set of PDs these metrics can be used to calculate Fréchet mean, variance and perform clustering (Turner *et al.* (2014)). However, these methods perform poorly under large deformations conditions which we will demonstrate later in section 4. Also, different machine learning tools demand more than just a metric structure. To overcome this complication, researchers have resorted to transforming PDs to other suitable representation spaces (Anirudh *et al.* (2016)). Bubenik proposed the notation of persistence landscapes (PL), which is a stable functional representation of a PD lying in a Banach space (Bubenik (2015)). It can be thought of as a sequence of envelope functions defined on the points in the PD which are ordered based on their importance. Bubenik's main motivation for defining PLs was to get a unique mean representation for a set of PDs. However, the major drawback of using PLs is that they provide decreasing amount of importance to higher order homology groups, which are more informative than their lower-order counter parts.Kernel methods have also been proposed

Rouse *et al.* create a simple vector representation by overlaying a grid over a PD and count the number of points that fall into each bin (Rouse *et al.* (2015)). This method is not stable, since a small shift in the points can result in a different end feature representation. Despite the fact, this idea has appeared in many other forms (Adams *et al.* (2017)). Our feature representation is also a variant of the above grid count method. Donatini *et al.* and Ferri *et al.* transform H_0 homology PDs into persistence surfaces by fitting and summing up Gaussian distribution centered at each point in the PD (Donatini *et al.* (1998)). Reininghaus *et al.* create a more stable representation by taking the weighted sum of positive Gaussians centered on each point above the diagonal and mirror the same below the diagonal but with negative Gaussians (Reininghaus *et al.* (2014)). Adams *et al.* design persistence images (PI), by computing the integral over each bin of the grid defined over the Gaussian-surface representation of the PD (Adams *et al.* (2017)). Both PI and the multi-scale kernel defined

by Reininghaus *et al.* show stability with respect to the traditional Wasserstein metrics. However, these methods perform well under small perturbation conditions and resort to weighting of points based on their lifetime. Depending on the problem, one may like to use a different weighting function. Giving importance to points with medium lifetimes helped Bendich *et al.* to best distinguish the data (Bendich *et al.* (2016)). Cohen-Steiner *et al.* benefited by prioritizing points near the death-axis and away from the diagonal (Cohen-Steiner *et al.* (2005)). In this paper, we propose a unique perturbation framework that goes around the task of selecting a weighting function by considering all the possible topological noise one could expect to see in the PD. We summarize this information and treat it as a point on the Grassmann manifold. We later show the effectiveness of our features in sections 4 5 for different different classification problems using data collected from different sensing devices.

The geometric properties of the Grassmannian have been used for various computer vision applications, such as object recognition, shape analysis, human activity modeling and classification, face and video-based recognition, *etc.* (Gopalan *et al.* (2012)). Lin *et al.* obtain descriptive projections by optimizing over the Grassmann manifold . Begelfor and Werman do clustering of shapes by using the affine shape space of the Grassmannian (Begelfor and Werman (2006)). Turaga *et al.* develop models that encode a linear subspace structure for image and video-based recognition applications. We suggest our readers to the following papers that provide a good introduction to the geometry, statistical analysis and techniques for solving optimization problems on the space of the Grassmann manifold (Turaga *et al.* (2010)). In our framework, the set of 2D PDFs obtained after perturbing the PD are mapped to a point on the Grassmann manifold.

Chapter 2

TOPOLOGICAL DATA ANALYSIS

This chapter provides a relatively detailed yet mathematically concise introduction to the rapidly emerging field of Topological Data Analysis (TDA). A very important tool in this exploratory space is persistent homology which will be presented in detail.

2.1 TDA -Introduction

Topological Data Analysis (TDA) refers to a collection of data analysis methods that find topological structure in data (Carlsson (2009)). This field of TDA has emerged from various advances in applied and algebraic topology and computational geometry and was pioneered by the works of Edelsbrunner *et al.* (2000) and Zomorodian and Carlsson (2005) in the persistent homology and was popularized by Carlsson (2009).

The main motivation of TDA is that data has shape and shape matters to gain powerful insights about its qualitative and quantitative features. Approaches to achieve this lie within topology and geometry (Chazal (2016)).

The goal of TDA is to provide well-founded mathematical and algorithmic methods to analyze and summarize the complex topological structures contained in the high-dimensional data using low-dimensional algebraic representations. Data is typically represented as point clouds in Euclidean space and in general metric spaces. TDA is also applied on functions defined on data. We will in detail, cover algorithms and concepts involved in TDA with respect to both these types of data i.e., point clouds in metric spaces and functions defined over data.

Since TDA is a very fast evolving field, a number of approaches have been proposed to achieve this and as mentioned earlier persistent homology is one of the most common approaches of it. Persistent homology is a multi-scale approach to extract the topological features of data. A typical pipeline of implementing TDA will be as follows Chazal and Michel (2017).

2.1.1 TDA -pipeline

1. The input data is typically a finite set of points with some notion of distance or similarity between them. This notion of similarity or distance plays a key role in the process of extraction of interesting topological structure of data. This distance can be induced by a metric in the ambient space or as an intrinsic metric provided by a pair wise distance metric. The choice of this distance or similarity is carefully made according to the application. Inmost cases this could be the familiar Euclidean metric when the data are embedded in $(\mathbb{R}^d)$.

2. Simplicial complexes are then built on the data to highlight the latent topology . This step is needed because point clouds in themselves carry does not non-trivial topology. Filtration i.e, nested simplicial complexes forms the vital step in the analysis. Simplicial complexes can be seen as higher dimensional generalizations of neighboring graphs. We will discuss simplicial complexes and the process of filtration in more detail in a subsequent section.

3. Upon completion of the process of filtration on the point cloud data we now extract topological or geometric information. This generally results in triangulation of the shape underlying the data from which summaries can be extracted by using tools like persistent homology.

4. The final stage of the pipeline is to use these extracted features to gain insights about the shape of the data by visualization and or use these features for analysis in machine-learning tasks. The choice of using these features in a stand-alone fashion or to explore its complementary properties with other features is again guided by application.

2.2 Homology

We will first discuss the concepts related to homology for the input data which is represented as point cloud in $(\mathbb{R}_d)$. We will look at various definitions of topology associated theorems and concepts of simplicial complexes. The goal of persistent homology is the measurement of the scale or resolution of a topological features. A short introduction to homology is as follows. Homology is a very involved subject in it's own right.Here an emphasis is laid on the intuition behind the idea rather than on its involved math. Homology characterizes sets based on connected components and holes. That is to say that homology is the concept utilized to summarize the connectivity of a topological space. The homology of a given space $\mathcal{M}$ is the collection of Abelian groups of various dimensions. The kth dimensional group representing the kth dimensional holes in $\mathcal{M}$. the kth homology group is represented by $H_k(\mathcal{M})$. The rank of this group β_k is called the kth Betti number. Generally speaking, the kth Betti number β_k is the number of kth dimensional holes in $\mathcal{M}$, that is β_0 is the number of connected components of $\mathcal{M}$, β_1 is the number of loops, β_2 is the number of enclosed voids and so on.

2.3 Simplicial Complexes

We will now briefly review the concepts involved in simplicial complexes. As mentioned earlier simplicial complexes, can be seen as generalization of neighborhood graphs

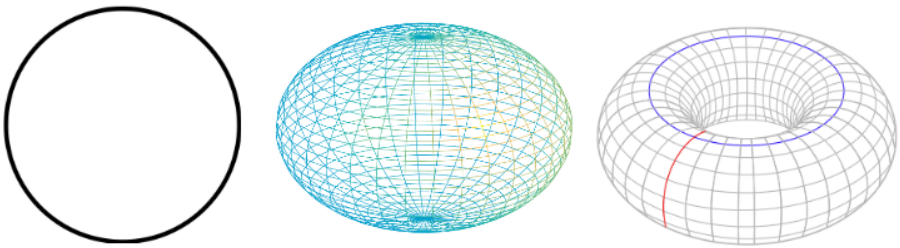


Figure 2.1: The circle has one connected component and one 1-dimensional hole: β_0 = 1, β_1 = 1. A sphere in $\mathbb{R}^3$ has one connected component and one 2-dimensional hole (void): β_0 = 1, β_1 = 0, β_2 = 1. While the torus has one connected component, two 1-dimensional holes and one void: β_0 = 1, β_1 = 2, β_2= 1

in higher dimensions. Given a set $\mathbb{X} = \{x_{\mathrm{o}}, \ldots, x_p\} \subset \mathbb{R}^d$ of p+1 affinely independent points , the *k-dimensional simplex* $\sigma = [x_{\mathrm{o}}, \ldots, x_p]$ spanned by $\mathbb{X}$ is the convex hull of $\mathbb{X}$.Affinely independent means that the k vectors x_i - x_o for i = 1 . . . ,p are linearly independent.

Vertices of σ are the points of $\mathbb{X}$ and *faces* of σ are the simplices spanned by the subsets of $\mathbb{X}$.

A *simplicial complex* K in $\mathbb{R}^d$ is a collection of simplices such that

1. Any face of a simplex of K is a simplex of K

2. the intersection of any two simplices of K is either a common face or empty or both

The underlying space of $K \subset \mathbb{R}^d$ which is a union of the simplices of K inherits from the topology of $\mathbb{R}^d$. Hence K can be viewed as a topological space through its underlying space Chazal and Michel (2017).

2.3.1 Building Simplicial Complexes

There are various methods that can be employed to build simplicial complexes on the given data set or in a general a topological or metric space. Here some of the most common approaches in practice are presented. We will first discuss about the widely-used Vietoris

Rips Complex.

Vietoris-Rips Complex

Let we be given point cloud $\mathbb{X}$ in a metric space (M, ρ) and a real number $\epsilon \geq 0$. The *Vietoris-Rips complex* of diameter ϵ is the simplicial complex or the set of simplices $[x_0, \ldots, x_k]$ such that d(x_i, x_j) $\leq \epsilon$ for all (*i,j*). This complex is denoted by Rips$_\epsilon$($\mathbb{X}$).That is consider the vertices $\{v_1, \ldots, v_p\}$ corresonding to the given data points. The simplex [v_1,v_2,...,v_k] is added to the simplicial complex iff the diameter of the set $\{x_1, \ldots, x_k\}$ is less than ϵ

Čech complex

Again lets start with a set of points $\mathbb{X}$ =$\{x_0, \ldots, x_p\}$.Now for any given $\epsilon > 0$, balls of radius ϵ around each point are then constructed.

After this, an abstract collection of vertices $\{v_1, \ldots, v_p\}$ corresponding to the given points are considered. Vertices v_i,v_j are connected by an edge if the balls B$_r$(x_i)and B$_r$(x_j) Krim *et al.* (2016). If three balls B$_r$(x_i), B$_r$(x_j) and B$_r$(x_k) we add the triangle [v_i,v_j,v_k]. This process of adding higher dimensional simplices if the balls intersect is continued in the similar fashion. What results by this procedure is what is called as Čech complex. One of the major disadvantages of this complex is that in order to build this on data you need to know the actual embedding of the data in ambient space.

Rips and Čech complexes are related by

$$\mathrm{Rips}_\epsilon(\mathbb{X}) \subseteq \mathrm{Cech}_\epsilon(\mathbb{X}) \subseteq \mathrm{Rips}_{2\epsilon}(\mathbb{X})$$

and that if $\mathbb{X}$ subset $\mathbb{R}^d$ then Čech$_\epsilon$($\mathbb{X}$) and $Rips_{2\epsilon}(\mathbb{X})$have the same set of vertices and edges. Chazal and Michel (2017)

2.4 Persistent Homology

As mentioned earlier Persistent homology is the most widely used tool to compute and study multiscale topological features efficiently. This tool has multifold uses.

1. Compute Betti Numbers of each complex
2. Encoding of the evolution of homology groups of the nested complexes across the scales.

As we noted earlier, in general, a filtration of a topological space $\mathbb{S}$ is a collection of subspaces $(S_r)_{r\in T}$, where $T \subseteq \mathbb{R}$, such that for any $r, r' \in T$, if $r \leq r'$ then $S_r \subseteq S_{r'}$ and, $S = \cup_{r\in T} S_r$. As an example, if $f : \mathbb{S} \to \mathbb{R}$ is a function, then the family $S_r = f^{-1}((-\infty, r])$, $r \in \mathbb{R}$ defines a filtration called the sublevel set filtration of f.

Filtrations in TDA are generally of two kinds.

1. Filtration on point cloud
2. Sublevel set filtrations

Let us now look at different examples for both of these with respect to persistent homology.

Example on point cloud data Consider the point cloud data as show in Figure 2.2 and is inspired from **??**. We will consider the filtration obtained by a union of balls of increasing radii on the points C.

(a) Radius $r = 0$, hence it is the data itself. Each point corresponds to 0-dimensional features also called as connected components. Hence *birth* for each of these features are at $r = 0$.

(b) As r keeps increasing some balls start overlapping resulting in *death* of some connected components. This will be encoded in the persistence diagram by a point on the y-axis

(c) As r increases again, now, all the individual connected components are merged giving raise to a single connected components. However, there is still two 1 -dimensional holes that need to be captured in the persistence diagram. Their start points *birth* are at this radius

(d) As we increase r more, one 1-dimensional hole gets collapsed.Hence at this radius one 1-dimensional feature has died. So, this will be represented in a persistence diagram as the pair(*birth*,*death*) where *birth* and *death* of this particular feature is the radius in step c and step d respectively.

(e) When r has been further increased, the remaining 1-dimensional features also gets merged.

This process has been beautifully captured by the representation of barcodes and persistence diagrams. The long the interval of the barcode the farthest is it from the diagonal in the persistence diagram. In this particular case there were two 1 dimensional holes and 1 connected components. This would map directly to our initial discussion of $Betti$-numbers. That is β_0=1 and β_1=2. So, persistence diagrams can be viewed as mutli-scale topological feature encoding the homology of union of balls for all radii and its evolution with r.

Example: Filtrations on functions :consider the Figure 2.3.

For all $\epsilon < p_1$, F_e is empty hence there is no information whatsoever in the barcode diagram or in the diagram. However, as $\epsilon = p_1$, we can see that a connected component appears in F_{p_1}. We thus register p_1 as the birth time of the first connected component and an

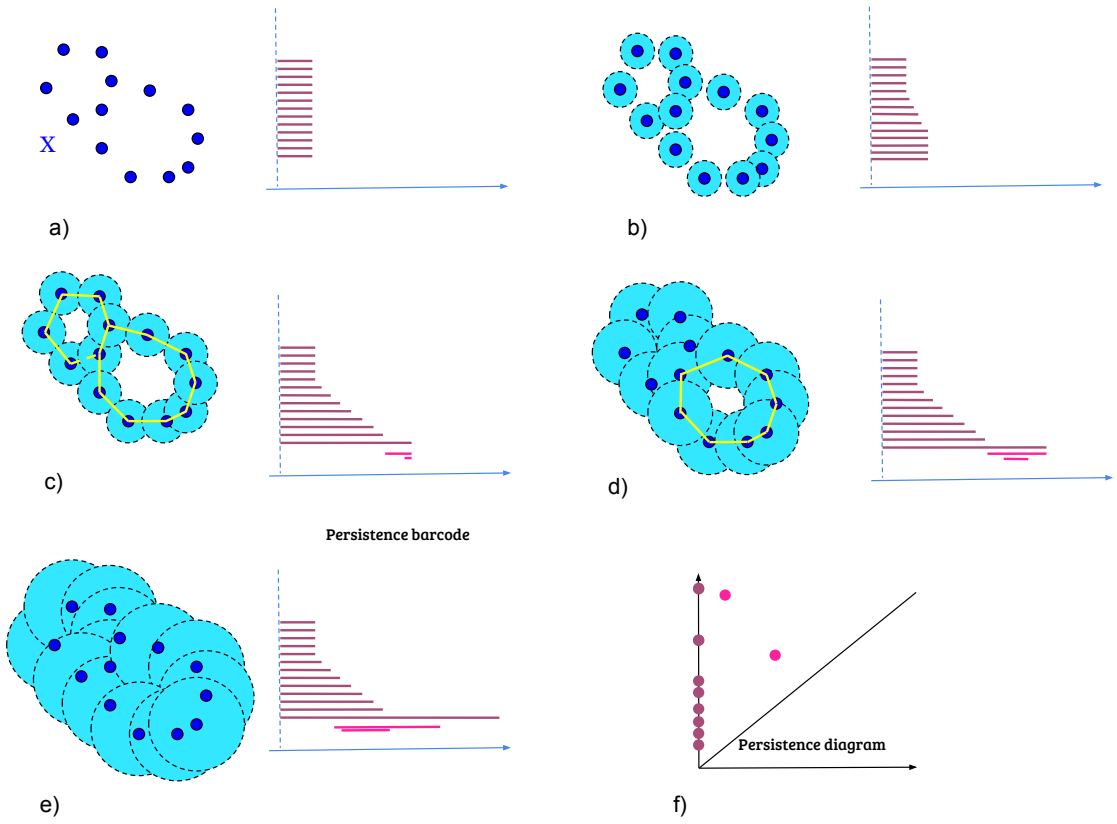


Figure 2.2: Filtration of the distance function to a point cloud and the "construction" of its persistence barcode as the radius of balls increases.

interval starting at p_1 will thus be started to track this connected component.As we proceed further to $\epsilon = p_2$, we observe that another connected component appears. We thus create another interval starting at $\epsilon = p_2$ to track this component. At $\epsilon = p_3$ one more connected components starts to appear, so in a similar fashion we create an third interval starting at $\epsilon = p_3$ to track this feature. When ϵ reaches p_4 the two connected components created at p_1 and p_3 merge together to give raise to a larger component. Hence the interval started at p_3 will now be ended at p_4 which is denoted appropriately in the barcode as well as in the persistence diagram.The lifespan which is defined as the death-birth of the feature is thus p_4-$p-3$. At $\epsilon = p_5$, we notice that the connected component that had its birth time at p_2 dies. Thus we represent this as the point (p_2, p_5) in the persistence diagram. However, the

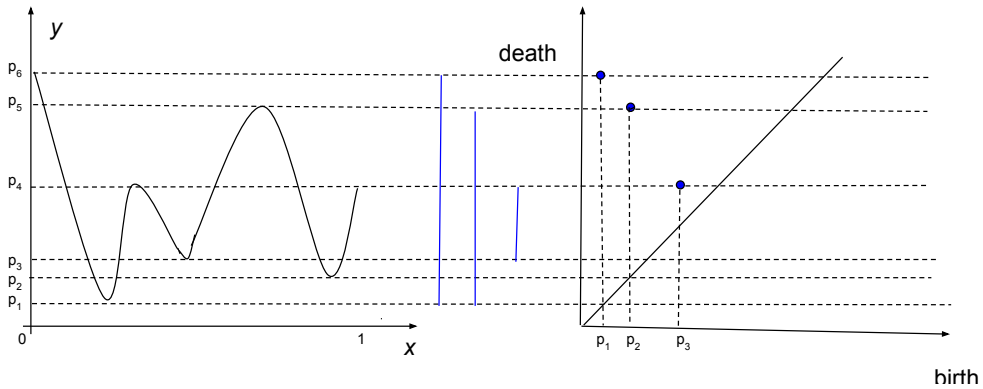


Figure 2.3: The Persistence Barcode and the Persistence Diagram of a Function $f : [0, 1] \to \mathbb{R}$.

interval created for the connected component that was born at p_1 continues to exist till p_6 if we stop the filtration at p_6 thus it can be represented as (p_1, p_6). Note that if the filtration is not stopped at p_6 but extends to $+\infty$, the filtration doesnot change after p_6, hence the interval or the (birth,death) representation of this connected component will be $(p_1, +\infty)$.

Having seen these two examples on two different kinds of data, we can now formally define what is a persistence diagram. $(V_r \mid r \in T)$

Definition 1 *A generalized persistence diagram is a countable multiset of points in* $\mathbb{R}^2$ *i.e.,* $\{(x, y) \in \mathbb{R}^2 \mid x, y \geq 0, x < y\}$ *along with the diagonal* $\Delta = \{(x, y) \in \mathbb{R}^2 \mid x = y\}$*, where each point on the diagonal has infinite multiplicity.*

It is not difficult to see why it is a multiset of points once we consider the example figure shown in 2.3. Here, it can be observed that in the persistence barcode there could be several copies of the same interval for a function. Correspondingly, in the persistence diagram, which is nothing but the representation of the birth and death of the connected components,each point has an integer -valued multiplicity. Hence persistence diagram is indeed a multiset of points in $\mathbb{R}^2$. A feature cannot die before it is born i.e., death$\geq$ birth hence the condition x$\leq$y. However, why is the other condition that each point on the

diagonal has infinite multiplicity imposed?. The answer to that question is stability. In fact, without the guarantee that these diagrams are stable with respect to small perturbations in the input data, we cannot infer information about the data from its persistence diagram.That is to say that persistence diagrams are only useful when we are certain that a slight change in the data does not have a huge impact in the diagram Chazal (2016). To formulate this notion of stability, we will need to have some notion of *d*istance or *s*imilarity between any two given persistence diagrams.

There is a innate metric associated to persistence diagrams called as *bottleneck distance*.

Definition 2 *Bottleneck distance*

Let A and B be two persistence diagrams. The bottleneck distance between these two persistence diagrams is given by

$$d_{L_\infty}(A, B) = \inf_{\phi: A \to B} \sup_{x \in A} \|x - \phi(x)\|$$

and the infimum is taken over all possible bijections between these two persistence diagrams.

That is to say that bottleneck distance is found out by minimizing the largest distances of any two *corresponding* points, found over all bijections between the diagrams.

Definition 3 *Space of persistence diagrams*

The space of the persistence diagrams $\mathcal{D}$ can be defined as

$$\mathcal{D} = \{x \mid d(x, \emptyset) < \infty\}.$$

where $\emptyset$ is a diagram with only a diagonal.

It is easy to see that bottleneck distance is indeed a metric on the space of the persistence diagrams $\mathcal{D}$. That is because,

1. $d_{L_\infty}(A, B) = 0$ iff $A = B$.

2. $d_{L_\infty}(A, B) = d_{L_\infty}(B, A)$ and

3. $d_{L_\infty}(A, B) \leq d_{L_\infty}(A, C) + d_{L_\infty}(C, B)$

The more general class of distances for measuring the distance between two persistence diagrams is the Wasserstein distance.

Definition 4 *p-Wasserstein Distance*

$$d_{L,p}(A, B)^p = (\inf_{\phi:A\to B} \sum_{x\in A} \|x - \phi(x)\|^p)$$

It should be noted that as $p \to \infty$ p-Wasserstein distance becomes bottleneck distance.

Theorem 1 (Stability of persistence diagrams) *Turner* et al. *(2014) let f, g are two tame Lipschitz functions $f, g : \mathbb{X} \to \mathbb{R}$*

$$d_{L^2}(\textit{Diag}(f), \textit{Diag}(g)) \leq 2^{\frac{k+2}{2}} \left[C\|f - g\|_\infty^{2-k}\right]^{1/2},$$

$k \in [1, 2)$ and C depends on Lipschitz properties of f, g.

Hence by having a infinite multiplicty of the points on the diagonal we can achieve stability with respect to small perturbations in the given data.

It is also important to note that the space of the persistence diagrams $\mathcal{D}$ is not a hilbert space but an Alexandrov space with non-negative curvature Turner *et al.* (2014). Hence these diagrams which are very non-linear in nature can not be used directly in the traditional machine learning paradigms. A number of approaches have been proposed in literature to over come this problem and are discussed in 1.1

The recent developments in the field of computational topology have enabled many efficient computation methods to compute topological invariants from data. There are

many freely available software now to perform TDA such as Gudhi Library(C++ and python) Maria *et al.* (2014) , TDA-R package Fasy *et al.* (2014), Javaplex(matlab) Adams *et al.* (2014), a highly distributed program DIPHA(python) Bauer *et al.* (2014) etc. These easy to use and efficient software have enabled researchers to apply TDA to very challenging tasks.

Chapter 3

DIFFERENTIAL GEOMETRY

In this section, a brief and terse introduction of differential Geometry is provided along with key concepts. We will look at a specific manifold called Grasmann Manifold for the reasons mentioned earlier.We also suggest the readers to refer to excellent textbooks such as Absil *et al.* (2009) and this chapter by Turaga *et al.* (2010).

Manifold: A topological space is called a manifold if it is locally Euclidean i.e.,

Definition 5 *A Hausdorff space $\mathcal{M}$ is defined as a manifold if for each point $p \in \mathcal{M}$, there exists a neighborhood $\mathcal{U}$ of p and a mapping f: $\mathcal{U} \to \mathbb{R}^n$ such that $f(\mathcal{U})$ is open in $\mathbb{R}^n$ and where f is a diffeomorphism.*

A topological manifold with globally defined differential structure is a differentiable manifold. This implies that there exists a map that goes from small neighborhoods to open sets of euclidean spaces thus allowing us to perform calculations in euclidean space. If many such mappings exists then they are compatible or their compositions are smooth.

Tangent and tangent space: The set of all possible tangent vectors for a given point $p \in \mathcal{M}$ is called a tangent space at p. It is denoted by $T_p\mathcal{M}$. It is important to note that the tangent space is a vector space. Hence this allows us to do familiar operations such as PCA, regression etc. The dimension of the tangent space is same as that of the manifold. Tangent vectors can be viewed as directional directives. For example consider a point $p \in \mathcal{M}$. Now consider a curve $\gamma(t)$ on the manifold passing through p such that $\gamma(0) = p$. The derivative of this curve at p, $\gamma'(0)$, is the tangent or the velocity vector. If one considers all possible curves through this point $\{\gamma_i(t)\}, i = 1, 2, \ldots$, then the set of all velocity vectors $\{\gamma_i'(0)\}$ is the tangent space $T_p\mathcal{M}$, at this point. The pole of the tangent space is the point at which

the tangents .The collection of all the tangent spaces of $\mathcal{M}$ is called the tangent bundle of $\mathcal{M}$.

Riemannian metric: Now, we wish to obtain a metric on $\mathcal{M}$. In order to achieve this one should impose a Riemannian structure on the manifold. A Riemannian metric is function that smoothly associates, to each point $p \in \mathcal{M}$, an inner product on the tangent space $T_p\mathcal{M}$. A smooth manifold equipped with a Riemannian metric is called a Riemannian manifold. Fletcher (2010)

Geodesic: In the euclidean space, the shortest path between any two points is a straight line. However, for non-linear spaces this is not true. In order to discuss the similar notion notion of the shortest distance between two points on a given manifold, consider a curve on the manifold $\gamma : [p1, p2] \rightarrow \mathcal{M}$ such that $\gamma(p1) = x$ and $\gamma(p2) = y$. Let the energy function be $E = \int_a^b ||\gamma'(t)||^2 \, dt$. The curve that achieves the minimum of this functional E is called the geodesic.This is the curve that minimizes the path length between two points on the manifold. The norm $||.||$ is induced by the Riemannian metric at $\gamma(t)$.

It has earlier been noted that since tangent space is a vector space, many computations that are not possible to be performed in non-linear spaces can be performed. Hence it is convenient for us to do our calculations on the tangent space rather than on the manifold directly because of its complex structure.In order to do this we need two maps defined. One to go to tangent space from the manifold and two from the tangent space back to the manifold. The first map is called exponential map and the second is called inverse-exponential or logarithm map. We provide definitions of both of the maps below

Exponential Map:Given that a unique geodesic $\gamma(t)$ exists locally at $p \in \mathcal{M}$ and $\gamma(0) = p$ and $\gamma'(0) = v \in T_p\mathcal{M}$, the exponential map at p is the function $\exp_p : T_p\mathcal{M} \rightarrow \mathcal{M}$ given by $\exp_p(v) = \gamma(1)$. For a neighborhood, $U \subset T_p\mathcal{M}$ containing 0, it can be shown that $\exp_p$ is a diffeomorphism, i.e., it has an inverse which is also continuous.

Logarithm Map: There is an inverse map of the exponential map that exists at least in the

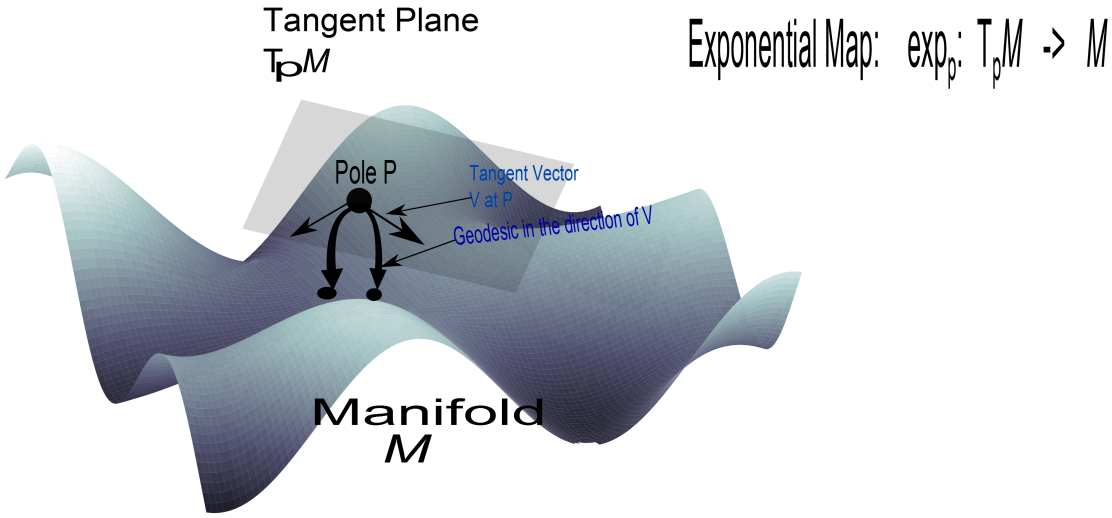


Figure 3.1:

neighborhood $U \subset T_p\mathcal{M}$ containing 0, called the logarithm map given by $\exp_p^{-1} : \mathcal{M} \to T_p\mathcal{M}$.

The algorithm for computing the Exponential and logarithm maps depends both on the manifold of interest and the pole of the tangent space.

3.1 Grassmann Manifold

Definition: Let n, p be two positive integers such that $n > p > 0$. The set of p-dimensional linear subspaces in $\mathbb{R}^n$ is called a Grassmann manifold, denoted by $\mathbb{G}_{p,n}$ or $\mathbb{G}(p, n)$. Each point $\mathcal{Y}$ on $\mathbb{G}_{p,n}$ is represented as a basis, *i.e.* a linear combination of the set of p orthonormal vectors $Y_1, Y_2, \ldots, Y_p$ as shown below

$$\mathcal{Y} = \Big[Y_1, Y_2, \ldots, Y_p\Big] = \begin{bmatrix} y_{1,1} & \cdots & y_{p,1} \\ \vdots & \cdots & \vdots \\ y_{1,n} & \cdots & y_{p,n} \end{bmatrix} \tag{3.1}$$

Distance Metrics: The *geodesic* distance $(d_{\mathbb{G}})$ between two points $\mathcal{Y}_1$ and $\mathcal{Y}_2$ on the Grassmann manifold is the length of the shortest constant speed curve that connects these points.

To do this first the velocity matrix $A_{\mathcal{Y}_1,\mathcal{Y}_2}$ or the inverse exponential map needs to be calculated, such that the geodesic path starts at $\mathcal{Y}_1$ and ends at $\mathcal{Y}_2$ in unit time. $A_{\mathcal{Y}_1,\mathcal{Y}_2}$ can be computed using a numerical approximation method described in Liu *et al.* (2003). Once we have $A_{\mathcal{Y}_1,\mathcal{Y}_2}$, the distance between $\mathcal{Y}_1$ and $\mathcal{Y}_2$ can be represented by the following equation, where $(.)^{\mathrm{T}}$ is the transpose operator

$$d_{\mathbb{G}}(\mathcal{Y}_1, \mathcal{Y}_2) = trace(A_{\mathcal{Y}_1,\mathcal{Y}_2}.{A_{\mathcal{Y}_1,\mathcal{Y}_2}}^{\mathrm{T}}) \tag{3.2}$$

The *arc-length0* metric is another commonly used distance measure that is derived from the intrinsic geometry of the Grassmann manifold Edelman *et al.* (1998). It takes the L-2 norm of the subspace angles computed between two subspaces as the distance measure.

$$d^2_{\text{arc}}(\mathcal{Y}_1, \mathcal{Y}_2) = \sum_{i=1}^{N} \theta_i^2 \tag{3.3}$$

The *symmetric directional* distance is one of many methods to compute distances between subspaces of different dimension p Sun *et al.* (2007); Wang *et al.* (2006). It is a widely used measure in several fields, such as computer vision da Silva and Costeira (2009); Basri *et al.* (2011); Bagherinia and Manduchi (2011); Luo and Huang (2014); Yan and Pollefeys (2006), communications Sharafuddin *et al.* (2010), applied mathematics Draper *et al.* (2014) and so on. It is equivalent to the *chordal metric* defined in Ye and Lim (2016).

$$d_{\text{chordal}}(\mathcal{Y}_1, \mathcal{Y}_2) = \left(\max(k, l) - \sum_{i,j=1}^{k,l} ({y_{1i}}^{\mathrm{T}}.y_{2j})^2\right)^{\frac{1}{2}} \tag{3.4}$$

In the above equation k and l are the subspace dimensions for the orthonormal matrices $\mathcal{Y}_1 = [y_{11}, y_{12}, \ldots, y_{1k}]$ and $\mathcal{Y}_2 = [y_{21}, y_{22}, \ldots, y_{2l}]$ respectively. We restrict ourselves to distance computations between same-dimensional subspaces, where $k = l$. In this paper we perform recognition with the d^2_{arc} and d_{chordal} distance metrics using a nearest neighbor classifier.

Chapter 4

PERTURBED TOPOLOGICAL SIGNATURES

In this chapter, we provide a detailed description of our proposed approach, with a goal to create robust representations of persistence diagrams with respect to topological noise. We call our representations as PTS short for Perturbed Topological Signatures. We explain our proposed approach in this section after the extraction of persistence diagrams(PDs). As explained in chapter 2, we extract persistence diagram by using two methods depending on the type of the input data. If the input data considered is point cloud data in $\mathbb{R}^n$, we use the *Vietoris-Rips complex* filtration and if the input data is actually functional data where functions are defined over the input data then sub-level set filtration is employed to extract persistence diagrams. The choice of representation of input data, that is, is it point cloud or function valued data is dataset and task specific and we explain the choices made and the methodology used to extract persistence diagrams for each of the experiments on three different datasets in chapter 5.

As discussed earlier, the goal is to create robust representation of persistence diagrams with respect to topological noise. The approach is straight forward and is as follows. A given persistence diagram is perturbed many times, thus resulting in a set of perturbed persistence diagrams for the same given data. An intelligent summarization of these set of perturbed persistence diagrams is achieved. This summarization, which in our case is a point on a Grassmann Manifold is then used for different classification tasks. This idea is similar to data augmentation practices employed in the computer vision community to obtain representations invariant to a property of interest. Examples include Gopalan *et al.* (2012), where robust descriptors of face images with respect to blur are obtained by convolving an image with a complete set of ortho-normal basis vectors which is equivalent to

blurring the image at various levels. Data augmentation has also been gaining steady popularity in the deep learning researchers as well, be it to increase the amount of training data available because the original available data is less Dosovitskiy *et al.* (2014) or achieving rotation invariance Marcos *et al.* (2016).

As noted, the space of persistence diagrams is not the familiar Hilbert space on which a number of standard machine learning algorithms exist but rather Alexandrov space with non-negative curvature, making it not conducive to standard machine learning.

So, in order to able to use the rich topological information encoded in the persistence diagrams for classification tasks there should be a mapping function that can represent these diagrams in a familiar space. However, the resultant representation of PDs should be ideally stable with respect to input noise, be efficient to compute and should not loose too much of information. Once such a representation is achieved for all the PDs, along with the augmented PDs, we then wish to create an intelligent summary of these representations(which are in a familiar vector space). This summary need not be a vector space.In fact, the choice we made is that, this summary of representation of PDs, is that of a point on a Grasssmann Manifold of a fixed dimension.

The problem of achieving robustness to topological noise is a challenging one. There is a good body of work and established theorems in literature such as Cohen-Steiner *et al.* (2005) and Chazal *et al.* (2008) that show that persistence diagrams are stable with respect to small perturbations in input data with respect to the chosen metric of Wasserstein distance. However, when the input data is noisy by a good extent, this stability no longer holds. As a consequence, there is a serious fall in the performance when these are used in the machine learning paradigms such as classification tasks. This reduction in performance with respect to the increase in noise level of the input data will be demonstrated in our experiment on the synthetic dataset on 3-D shapes. We will also show that, in the experiments section that by the proposed approach we could obtain a considerable boost in the

performance on different datasets on various domains such as 3-D shape retrieval, Action recognition from motion-capture and Multi-view action recognition on IXMAS dataset.

An important thing to note is that our approach is blind to the method in which persistence diagrams are extracted,be it through *Vietoris-rips complex* filtration or *sub-level set* filtration. We will now start the description of our approach after the extraction of persistence diagram as discussed before.

1. **Rotating Persistence Diagrams:** Recollect that persistence diagrams are defined as a multi set of points in $\mathbb{R}^2$ i.e., $\{(x, y) \in \mathbb{R}^2 \mid x, y \geq 0, x < y\}$ along with the diagonal $\Delta = \{(x, y) \in \mathbb{R}^2 \mid x = y\}$, where each point on the diagonal has infinite multiplicity given in **definition** 1. Now, we define a transformation $G : \mathbb{R}^2 \to \mathbb{R}^2$ such that $G(b, d) = (\frac{b+d}{2}, d - b)$ where b and d are the birth-time and death-time of a topological feature in the persistence diagram. This is equivalent to rotating the persistence diagram in the clock-wise direction. The y-axis that is (death-birth) now represents the life-time or the *persistence* of the given topological feature in a persistence diagram. This construction helps in optimally utilizing the 2D space. The reasons for such a claim will be provided in the subsequent paragraphs.

2. **Creating a set of Perturbed PDs:** For each PD extracted and rotated, we create a set of m PDs where each PD in this set has it's points randomly displaced by a certain amount within a fixed radius. The obtained set of randomly perturbed PDs retain the same topological information of the input data as the original PD. However, together they capture all probable variations of the input data when subjected to changes in scale. This step allows us to incorporate noise-robustness, when the input data is subjected to a certain level of noise.

3. **Converting PDs to 2D Probability Distribution Functions:** We convert the initial PD and each of the m randomly perturbed rotated PDs to 2D probability distribution

functions (PDFs) by treating the points in the PD as samples from an underlying PDF. We fine tune the bandwidth parameter to get best results. One could avoid the hassle of fine tuning this parameter by generating multiple PDFs using a range of different bandwidth parameters for each of the $m + 1$ PDs and still obtain optimal results as we shall show in our experiments. For now let us consider that we generate only one 2D PDF per PD.

4. **Projecting 2D PDFs to a point on the mannian:** Once we have all the 2D PDFs and with each PDF having n bins, we vectorize each PDF and stack them together to get a 2D matrix of size $n \times m$ ($n >> m$), here m being the total number of PDFs. On this matrix we apply singular value decomposition (SVD) and select the p largest left singular vectors. This will result in a $n \times p$ orthonormal matrix which is a point on the Grassmann manifold. We will later show how our end Grassmann representation is robust to noise in the input data and perturbation of the points in the PD.

The pipeline is illustrated in 4.1 with an example from the IXMAS dataset. Specific details of IXMAS dataset are described in the experiments section. The first column represents key frames of 3 selected action classes, Check watch, sit down and walk. The second column represents the respective persistence diagram of each action sequence.Then the result of the step 2 i.e., a set of perturbed PD's is shown in figure 3 while the PDF representation of each of these perturbed PD is shown in column four. Finally the end representation which is a low -dimensional subspace of these 2-D PDF's, that lie on the Grassmann manifold is shown. Once we achieve the final Grassmann Representation, with its well-studied structure, these representation can be used in machine learning pipelines such as classification tasks employing algorithms like K-Nearest Neighbor or Support Vector Machines.

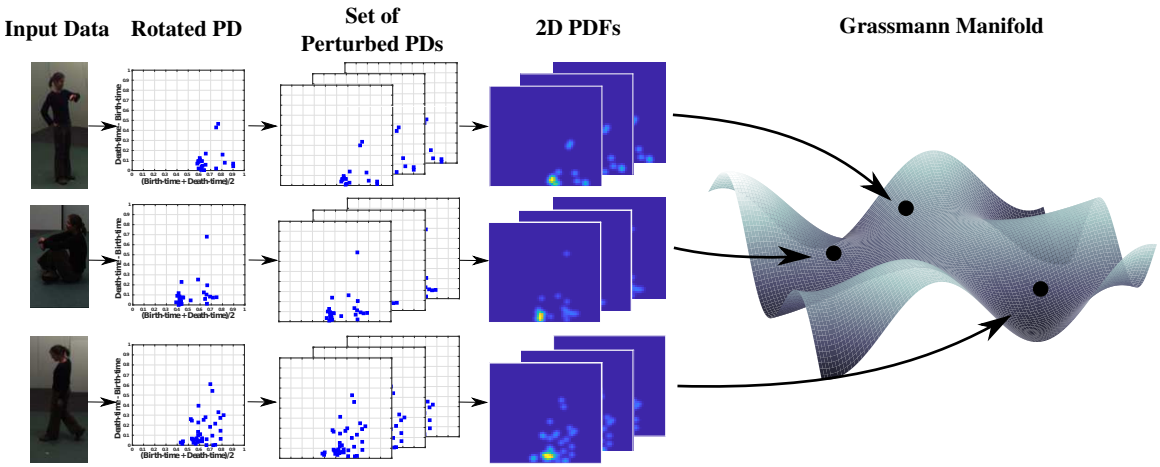


Figure 4.1: Illustration of the Proposed Framework's Pipeline.

Chapter 5

EXPERIMENTAL RESULTS

In this chapter, we show our results on the different experiments carried out. There are two kinds of experiments conducted. These are Experiments on Synthetic dataset and real datasets. The synthetic experiment performed asserts our claim that the proposed approach indeed achieves robustness to perturbations in input data when compared against the gold standard traditional metrics such as Wasserstein Metrics. We report classification accuracy achieved through our proposed approach and compare them against standard base-lines. While the datasets considered for experiments on real dataset are of three different domains namely, 3-D shapes, motion capture data and videos of human actors performing a specific task

5.1 Synthetic Experiment

This experiment as mentioned above is to establish the superiority of our method and its robustness to perturbations in the input data. We considered the SHREC 2010 dataset Lian *et al.* (2010). It consists of 200 near-isometric watertight 3D shapes with articulating parts, equally divided into 10 different classes namely - *ants, crabs, spectacles, hands, humans, octopuses, pliers, snakes, spiders* and *teddy-bears*.

The sample shapes of each class of this dataset is shown in figure 5.1

In this experiment we select one sample per each class at random. Let this set of ten shapes each representing one sample be denoted by $\mathcal{S}$. The minimum bounding sphere for each of these shapes has a mean radius of 54.4 with standard deviation of 3.7 centered at (64.4, 63.4, 66.0) with coordinate-wise standard deviations of (3.9, 4.1, 4.9) respectively. For each $s \in \mathcal{S}$, we add a zero mean Gaussian noise of ten incremental standard deviations

Figure 5.1: Sample Shapes of SHREC 2010 Dataset

ranging from 0.1 to 1 in steps of 0.1. Thus now the earlier set of original shapes $\mathcal{S}$ is augmented by 100 shapes. 10 perturbed shapes per each shapes. Let this augmented set of shapes i.e., original shapes along with 100 perturbed shapes be called $\mathcal{PS}$. For each $ps \in \mathcal{PS}$, we extract persistence diagrams using sub-level set filtration method. The functions that we define on these shapes is that of spectral descriptor functions. Scale Invariant- Heat kernel Signature, a spectral descriptor Sun *et al.* (2009); Aubry *et al.* (2011); Kokkinos *et al.* (2012) which captures scale invariant macroscopic properties of the shape is used in this experiment. The number of dimensions of this SIHKS descriptor considered are 17. Hence 17 persistence diagrams are computed per shape for each $ps \in \mathcal{PS}$. Grassmann representation of these persistence diagrams are obtained through the method explained in chapter 4.

We now perform a nearest neighbor classification on this set $\mathcal{PS}$ using $L-1$,$L-2$ Wasserstein metrics, $bottle-neck$ distance on the persistence diagrams. These results are compared against using d_{Δ} and $d_{\mathbb{G}}$ i.e., *symmetric directional distance* and *geodesic distance* respectively, defined in section 3.1 on our Grassmann representations. This process is repeated 100 times. The average accuracy after performing this experiments over 100 folds

is reported in table 5.1. It can be seen that even though the shapes are highly noisy , we still get considerably good accuracy against other baselines, see table 5.1 and at a much reduced computational load. Thus showing that our proposed method indeed achieved robustness against topological noise of the input data.

Method	$\mathcal{N}_1$	$\mathcal{N}_2$	$\mathcal{N}_3$	$\mathcal{N}_4$	$\mathcal{N}_5$	$\mathcal{N}_6$	$\mathcal{N}_7$	$\mathcal{N}_8$	$\mathcal{N}_9$	$\mathcal{N}_{10}$	**Average Accuracy (%)**	**Time (10^{-4} sec)**
L_1-Wasserstein	96.50	100.00	95.50	97.30	100.00	100.00	100.00	100.00	100.00	100.00	98.93	256.00
L_2-Wasserstein	85.70	99.80	91.40	86.50	99.70	99.60	94.60	94.30	98.40	100.00	95.00	450.00
Bottleneck	90.40	100.00	94.60	93.50	99.90	99.80	99.70	100.00	100.00	100.00	97.79	36.00
Proposed - $\mathbf{d_\Delta}$	96.60	100.00	99.50	97.00	99.50	100.00	100.00	100.00	100.00	97.70	**99.03**	**1.60**
Proposed - $\mathbf{d_\mathbb{G}}$	97.60	100.00	98.90	95.80	99.60	100.00	100.00	100.00	100.00	97.60	**98.89**	**2.30**

Table 5.1: Correct Classification of 100 Noisy Shapes, with 10 Noisy Shapes with Different Noise Levels Extracted for 10 Different Shapes from the SHREC 2010 Dataset.

b

Figure provides a good illustration of the dataset. 5 shapes from the set $\mathcal{S}$ are chosen. Their noisy or perturbed versions obtained by adding zero-mean white Gaussian noise are also shown in column 6. It can be seen that the figures in column 6 are very noisy. Columns 2 and 5 represent the persistence diagram obtained through the SIHKS function defined on these shapes. Columns 3 and 4 are the visualizations of our Grassmann representations at $1^s t$ subspace.

5.2 Real Experiments

We apply the framework described in the previous chapter on three different datasets with different types of input data - 1) SHREC 2010 3D shape dataset Lian *et al.* (2010), 2) Motion capture dataset Ali *et al.* (2007) and, 3) IXMAS video dataset Weinland *et al.* (2007)

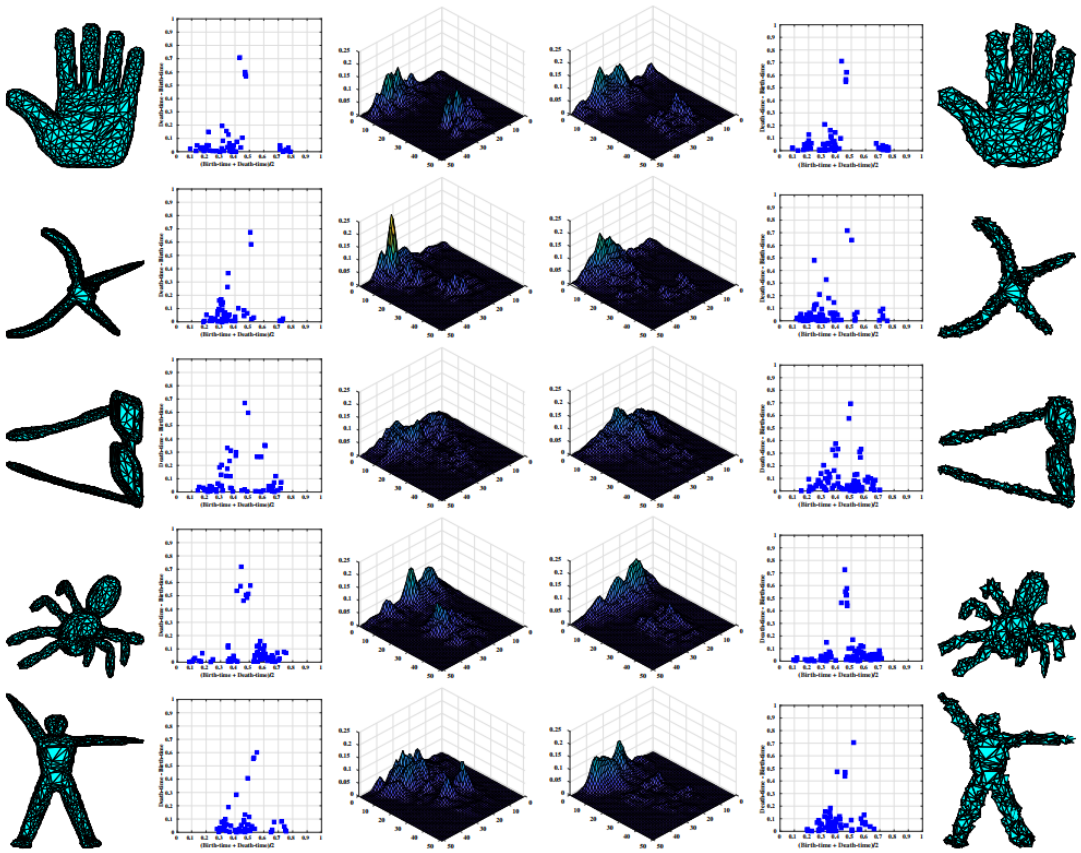


Figure 5.2: illustration of original shapes,respective persistence diagrams, visualization of Grassmann representations from columns 1 to 3. Columns 4 to 6 represent Grassmann representations,persistence diagrams and perturbed shapes with zero mean Gaussian noise with standard deviation of 1.

We will first briefly talk about each dataset and later describe the experimental objectives and procedures that were followed.

5.2.1 SHREC 2010

SHREC 2010 shape retrieval dataset consists of 200 near-isometric watertight 3D shapes with articulating parts, equally divided into 10 different classes namely - *ants, crabs, spectacles, hands, humans, octopuses, pliers, snakes, spiders* and *teddy-bears*. Each 3-D mesh has tens of thousands of faces and vertices. We simplify each of these 3-D meshes to 2000 faces and 1002 vertices as suggested in Li *et al.* (2014).

PDs are extracted using the spectral descriptor functions that are defined on each shape and are isometry-invariant. The following spectral descriptors were used to calculate the PDs - heat kernel signature (HKS), wave kernel signature (WKS) and scale-invariant heat kernel signature (SIHKS) Sun *et al.* (2009); Aubry *et al.* (2011); Kokkinos *et al.* (2012). These descriptors are standard spectral descriptors in shape analysis. HKS is based on the concept of heat diffusion over a surface. Intuitively, this can be thought of as follows. Let there be some surface and its initial heat distribution is known. Heat kernel, a function, gives us an idea of how much heat is transfer reed from one point to another point as the time increases. Using this as a feature vector will be very complex because the feature vector will be of very large dimension.Instead Heat kernel signature doesn't relate to how much of heat is transferred from one point to another with increase in temporal parameter, rather it just considers the amount of heat that has changed at a given point with respect to time resulting in less-dimensional data. SIHKS is the scale invariant version of HKS. Wave kernel signature on the other hand replace the heat equation with Schrödinger's wave equation. WKS can be intuitively thought of set of bandpass filters Bronstein (2011). Thus WKS captures local features very well but will not fare well when it comes to capturing global features. However, one inherent challenge in using these spectral descriptor is the choice of time parameters. We followed the parameters used by Li *et al.* (2014).

So, in a nut shell, HKS and WKS are used to characterize the microscopic and the macroscopic shape properties of the 3D meshes and the SIHKS descriptor being a scale-invariant version of HKS. persistence diagrams are then extracted from these descriptors and , we follow our pipeline as illustrated in chapter 4. That is we proceed to perturb the PD's and extract the 2-D PDF's. We perturb the persistence diagrams 40 times. As mentioned earlier, we use a single bandwidth parameter in the kernel density estimation to obtain 2-D PDF's. Grassmann representations of various sub-space dimensions p are then obtained from this set of 2-D PDF's by performing a principal component analysis.

To evaluate our performance, a Nearest Neighbor classifier is employed. Achieved performance is compared against the baseline methods Li *et al.* (2014). Li *et al.* report their best results after carefully tuning the weights for the weighted sum of the distance matrices obtained through Bag-of-features and Wasserstein distance.In a quest to improve the results further, they consider fusing with other descriptors such as ISPM etc along with all the three spectral descriptors namely,HKS,SIHKS and WKS. It should be noted that by using our approach after combining our end representations of HKS,SIHKS and WKS , we significantly outperform them without resorting to any further fusion practices. Our method using the symmetric directional distance metric d_{Δ} achieved an accuracy of 99.50%, outperforming all other baseline methods.Please see table 5.2 . The average classification result by varying the subspace-dimension p from 1 to 25 of the Grassmann feature representation $\mathbb{G}_{p,n}$ is 98.42$\pm$0.4 and 98.72$\pm$0.25 using d_{Δ} and $d_{\mathbb{G}}$ metrics respectively, thereby showing the stability of our feature.

Spectral Descriptor	**Methods**	**Nearest Neighbor (%)**
WKS+ HKS+ SIHKS	BoF	97.00
	SSBoF	97.50
	ISPM	97.50
	PD	98.50
	BoF+PD	98.50
	ISPM+PD	99.00
	Proposed ($d_{\mathbb{G}}$)	**99.00**
	Proposed (d_{Δ})	**99.50**

Table 5.2: Comparison of the Nearest Neighbor Classification Results of the Proposed Method with Other Baseline Methods Li *et al.* (2014).

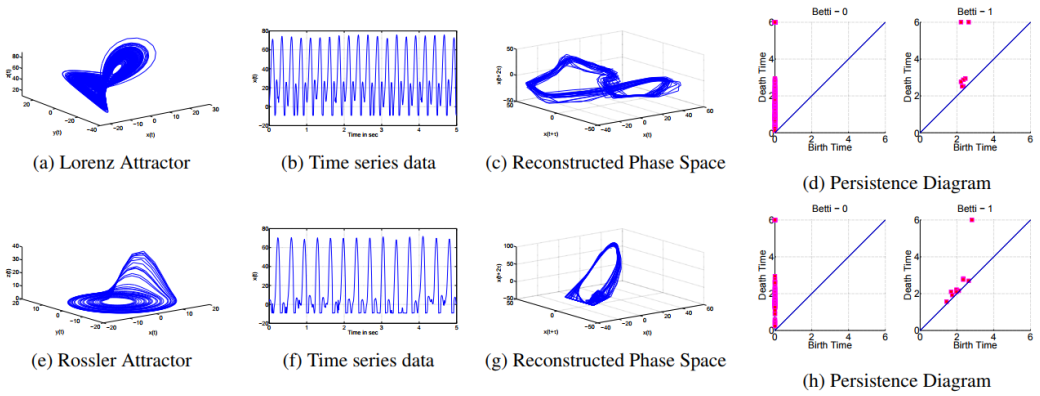
5.2.2 Motion Capture

This dataset consists of 3D human-body joint motion capture sequences, where each sequence contains 57 trajectories (19 joint trajectories along 3 axes). There are 5 different action classes - *dance, jump, run, sit and walk*, with each class containing 31, 14, 30, 35 and 48 action sample sequences respectively. Since this is a point cloud data in $\mathbb{R}^n$ we use Vietoris-Rips Complex to extract persistence diagrams.

The procedure to extract the PDs is well explained in the paper Venkataraman *et al.* (2016).We suggest the readers to consult that paper for further clarity. Here just a brief summary of procedure to extract PDs from Motion Capture data is provided and is illustrated in figure 5.3

The data obtained from sensors is generally a projection of the original dynamic system onto a low-dimensional space,hence, it might not have all of the useful information regarding all the latent variables of the original system, thus rendering this data inadequate to model the system. Hence, we would like to reconstruct the attractors to obtain the phase-space with the constraint that the topological information is preserved in both these settings. That is we wish to obtain m-dimensional data from the available 1-dimensional data assuming that all variables of the system have a mutual influence on one other.Thanks to a theorem due to Takens *et al.* (1981) called Takens embedding theorem, this can be achieved.

Once this high-dimensional data which is viewed as a point cloud data is obtained , we extract persistence diagrams of H0 and H1 homology groups i.e., connected components and 1-D holes using *V-R* complex as explained in chapter 2. PDs are extracted over each trajectory after phase-space reconstruction thus resulting in 57 PDs per action. We only consider H1 homology group PD's because they are the ones that have the most useful information in this context. Each of the 57 PDs when passed through our framework result



(a) Lorenz Attractor (b) Time series data (c) Reconstructed Phase Space (d) Persistence Diagram

(e) Rossler Attractor (f) Time series data (g) Reconstructed Phase Space (h) Persistence Diagram

Figure 5.3: Reproduced with approval from Venkataraman *et al.* (2016). Phase space reconstruction of dynamical attractors by delay embedding. (a), (e) shows the 3D view of trajectories of Lorenz and Rossler attractors. The one-dimensional time series (observed) of the Lorenz and Rossler systems are shown in (b), (f). (c), (g) shows the reconstructed phase-space from observed time series using delay embedding.

in 57 different Grassmann representations or points, where each point lies on a Grassmann manifold of a specific sub-space dimension. Let the 57 Grassmann points for a given action be represented by $\mathcal{Y}_1, \mathcal{Y}_2 \ldots \mathcal{Y}_{57}$ and their respective Grassmann manifold spaces by $\mathbb{G}_1(p,n), \mathbb{G}_2(p,n) \ldots \mathbb{G}_{57}(p,n)$.

The evaluation protocol is that of reporting the average classification performance over 100 random splits, with each split having 25 random test samples (5 random test samples selected from each action class) and remaining 133 training samples. For SVM classification, we train non-linear SVMs using the *projection kernel* defined below Hamm and Lee (2008).

$$k_p(\mathcal{X}_i, \mathcal{Y}_i) = \|\mathcal{X}_i^{\mathrm{T}} \mathcal{Y}_i\|_F^2 \tag{5.1}$$

Here $\mathcal{X}_i$ and $\mathcal{Y}_i$ are two points on the grasmann manifold $\mathbb{G}_i(p,n)$, and $\|.\|_F$ refers to the Frobenius norm. The results are tabulated in Table 5.3.

It is to be noted that we achieve compare performance with respect to the Wasserstein

Method	Accuracy (%)	Time (sec)
Chaos Ali *et al.* (2007)	52.44	-
VR Complex Zomorodian (2010)	93.68	-
T-VR Complex Venkataraman *et al.* (2016)	96.48	$(1.2\pm1.23)\times 10^3$
Hilbert Sphere 1NN Anirudh *et al.* (2016)	89.87	(0.059 ± 0.044)
Hilbert Sphere PGA+SVM Anirudh *et al.* (2016)	91.68	-
Proposed - NN	85.96	$\mathbf{0.300}\times\mathbf{10^{-4}}$
Proposed - SVM	91.92	-

Table 5.3: Comparison of Accuracies Obtained Using the Proposed Approach to That of Other Baselines on Mocap Dataset

metric, the gold standard metric for comparing persistence diagrams, we do this at a much lesser computational time.We also compare our result with that of Anirudh *et al.* (2016). We outperform Anirudh *et al.* (2016) which uses squareroot framework to represent the 2-D PDF's onto a hilbert sphere, followed by Principal Geodesic analysis and SVM classification with linear kernel.

This is to say that even though there is a slight reduction in performance when compared to Wasserstein distance, we achieve comparable performance with our SVM approach and at much faster speed.

5.2.3 IXMAS Dataset

The IXMAS multi-view action dataset consists of synchronized action videos captured from 5 different views. Out of these five views, four of them are side views and one is a top view. This is a popular dataset in multi-view action recognition settings. It contains videos of 11 daily-life actions performed 3 times by 10 actors. The 11 action classes are as follows - *check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave,*

punch, kick, pick up. We follow the protocol as mentioned in Junejo *et al.* (2011). That is we employ a leave one out strategy which implies that the same actor's action videos would not be present in training and testing sets simultaneously.

We follow the following approach to obtain persistence diagrams on this dataset. Having inspired from works on shape analysis using descriptors such as DM1,DM2,DM3,AM3 etc Som *et al.* (2016, 2017); Som (2016) and Venkataraman and Turaga (2016),we extract histograms of these descriptors across 5 different scales and 50 bins per each frame. We extract one PD per bin across all the frames. So in total for a given action sequence we extract 250 PDs.After processing through our pipeline we extract 250 Grassmann representations of each action sequence.This is a lot of data, comparing two sets of Grassmann points for 2 action sequences is itself not easy and to do it over the entire dataset is impractical. Instead we can summarize all this information and map it to just one point on a another Grasmann manifold. We do this by first selecting the 1st k subspaces $(k < p)$ from each of these points and stack them together to get a $n \times (250.k)$ matrix. We then perform a Principal component analysis,and hence the resultant is a point on the Grassmannian.Now each sequence is mapped to a point on the Grassmannian and we use the well defined kernels to map this to RKHS Hamm and Lee (2008).

It has been established in Junejo *et al.* (2011) and other papers that self-similarity matrix(SSM) i.e., a pairwise distance matrix of low-level features such as Histogram of Oriented Gradients, optical flow serve as good descriptors for view-independence. They define a log-polar descriptors on these matrices. Using Bag of features approach and chi-squared kernels, classification using SVM's are carried out. We exploit the complementary capabilities of our method in IXMAS by combining our extracted kernels by using kernels defined on Grassmann to those of the kernels extracted from SSM's and show significant boost in the same-camera and cross-camera settings.please see table 5.4

Method	**Same Camera Accuracy (%)**	**Any-To-Any Accuracy (%)**
SSM-hog	67.30	52.60
SSM-of	66.60	53.80
SSM-hog-of	76.28	61.25
SSM-hog + PG-AM3	73.15	58.36
SSM-hog + PG-DM1	74.25	59.26
SSM-hog + PG-DM2	74.92	59.77
SSM-hog + PG-DM3	76.18	60.33
SSM-of + PG-AM3	72.01	58.85
SSM-of + PG-DM1	73.67	59.56
SSM-of + PG-DM2	73.45	60.60
SSM-of + PG-DM3	74.41	61.50
SSM-hog-of + PG-AM3	79.30	64.92
SSM-hog-of + PG-DM1	79.60	65.39
SSM-hog-of + PG-DM2	79.85	65.70
SSM-hog-of + PG-DM3	81.12	66.16

Table 5.4: Comparison of Accuracies Obtained Using the Proposed Approach to That of Other BaselinesJunejo *et al.* (2011) on IXMAS Dataset

5.3 Time-complexity of Comparing Topological Representations

Dataset	Average Number of Points in PD	Average Time Taken (10^{-4} sec)					Subspace Dimension (p) of PTS Feature
		L_1-Wasserstein	L_2-Wasserstein	Bottleneck	$d_{\mathbb{G}}$	d_{Δ}	
SHREC 2010 Lian *et al.* (2010)	71	256.00	450.00	36.00	**2.30**	**1.60**	10
IXMAS Weinland *et al.* (2007)	23	16.00	16.00	3.43	**2.21**	**0.68**	20
Motion Capture Ali *et al.* (2007)	27	22.00	22.00	2.72	**0.24**	**0.19**	1

Table 5.5: Comparison of the Average Time Taken to Measure Distance Between Two PDs Using the L_1, L_2-Wasserstein and Bottleneck Metrics, and Between Two PTS Features Using $d_{\mathbb{G}}$ and d_{Δ} metrics. The Time Reported is Averaged Over 3000 Distance Calculations Between the Respective Topological Representations for All Three Datasets Used in Section 5

The $d_{\mathbb{G}}$ and d_{Δ} metrics used to compare different PTS representations are fast and computationally less complex compared to the bottleneck and Wasserstein distance measures. The average time taken to compare two topological signatures (PD or PTS) for each of the datasets is tabulated in table 5.5. The table also shows the average number of points seen per PD and the subspace dimension p used for the PTS representation to get best results. All experiments are carried out on a standard Intel i7 CPU using Matlab 2016b with a working memory of 32 GB.

Chapter 6

CONCLUSION

In this thesis, we have presented a method for achieving robustness to input noise data by creating an intelligent summary of the topological signatures.The synthetic experiment shows the superiority of our approach when the input is subjected to a significant amount of topological noise. Experiments on SHREC 2010 datasets show that the proposed approach also greatly helps in correctly classifying different 3-D shapes.On the MoCap system, we achieve comparable performance with respect to Wasserstein metric and we outperform the results reported Anirudh *et al.* (2016) while On the IXMAS dataset, the proposed approach's, complementary strengths are explored and exploited. It has to be noted that the computation time required to compare the end representations using the proposed methods is orders faster than the traditional methods.

6.1 Future Directions

There are several future avenues for research. They include fusion with contemporary deep-learning architectures for exploiting the complementarity of both paradigms. We expect that topological methods will push the state-of-the-art in invariant representations, with key contributions being to recast the required invariance as a topological property of an appropriately redefined metric space. Additionally, the proposed methods may help open new feature-pooling options in deep-nets. At the heart of this thesis is the question how do we generate appropriate summaries of persistence diagrams with a specific objective of making them robust with respect to input topological noise.This research choose that summary to be as a point on a Grassmann Manifold. However, this meant that we need to

first represent the PDs in familiar space on which statistical analysis such as PCA or SVD could be performed. This was because of the space of the persistence diagrams i.e., an Alexandrov space is a mathematically complex space. One possible direction of research is to develop kernel functions that could embed points in this space in RKHS or RKKS. One another direction of further scope could be to model topological noise and derive a generalized model to extract a more stable representation of persistence diagrams.

REFERENCES

Geometric Approaches for Modeling Movement Quality: Applications in Motor Control and Therapy, Ph.D. thesis, URL `http://login.ezproxy1.lib.asu.edu/login?url=https://search.proquest.com/docview/1796260937?accountid=4485` (2016).

Absil, P.-A., R. Mahony and R. Sepulchre, *Optimization algorithms on matrix manifolds* (Princeton University Press, 2009).

Adams, H., T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta and L. Ziegelmeier, "Persistence images: A stable vector representation of persistent homology", Journal of Machine Learning Research **18**, 1–35 (2017).

Adams, H., A. Tausz and M. Vejdemo-Johansson, "Javaplex: A research software package for persistent (co) homology", in "International Congress on Mathematical Software", pp. 129–136 (Springer, 2014).

Ali, S., A. Basharat and M. Shah, "Chaotic invariants for human action recognition", in "IEEE 11th International Conference on Computer Vision (ICCV)", pp. 1–8 (2007).

Anirudh, R., V. Venkataraman, K. Natesan Ramamurthy and P. Turaga, "A riemannian framework for statistical analysis of topological persistence diagrams", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops", pp. 68–76 (2016).

Aubry, M., U. Schlickewei and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis", in "Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on", pp. 1626–1633 (IEEE, 2011).

Bagherinia, H. and R. Manduchi, "A theory of color barcodes", in "Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on", pp. 806–813 (IEEE, 2011).

Basri, R., T. Hassner and L. Zelnik-Manor, "Approximate nearest subspace search", IEEE Transactions on Pattern Analysis and Machine Intelligence **33**, 2, 266–278 (2011).

Bauer, U., M. Kerber and J. Reininghaus, "Distributed computation of persistent homology", in "2014 Proceedings of the Sixteenth Workshop on Algorithm Engineering and Experiments (ALENEX)", pp. 31–38 (SIAM, 2014).

Begelfor, E. and M. Werman, "Affine invariance revisited", vol. 2, pp. 2087–2094 (IEEE, 2006).

Bendich, P., S. P. Chin, J. Clark, J. Desena, J. Harer, E. Munch, A. Newman, D. Porter, D. Rouse, N. Strawn and A. Watkins, "Topological and statistical behavior classifiers for tracking applications", Aerospace and Electronic Systems, IEEE Transactions on **52**, 6, 2644–2661 (2016).

Bronstein, A. M., "Spectral descriptors for deformable shapes", arXiv preprint arXiv:1110.5015 (2011).

Bubenik, P., "Statistical topological data analysis using persistence landscapes", The Journal of Machine Learning Research **16**, 1, 77–102 (2015).

Carlsson, G., "Topology and data", Bulletin of the American Mathematical Society **46**, 2, 255–308 (2009).

Carlsson, G., A. Zomorodian, A. Collins and L. J. Guibas, "Persistence barcodes for shapes", International Journal of Shape Modeling **11**, 02, 149–187 (2005).

Chazal, F., *High-Dimensional Topological Data Analysis* (CRC Press, 2016).

Chazal, F., D. Cohen-Steiner, M. Glisse, L. J. Guibas and S. Oudot, *Proximity of persistence modules and their diagrams*, Ph.D. thesis, INRIA (2008).

Chazal, F. and B. Michel, "An introduction to topological data analysis: fundamental and practical aspects for data scientists", (2017).

Cohen-Steiner, D., H. Edelsbrunner and J. Harer, "Stability of persistence diagrams", in "Proceedings of the twenty-first annual symposium on Computational geometry", pp. 263–271 (ACM, 2005).

da Silva, N. P. and J. P. Costeira, "The normalized subspace inclusion: Robust clustering of motion subspaces", in "Computer Vision, 2009 IEEE 12th International Conference on", pp. 1444–1450 (IEEE, 2009).

Donatini, P., P. Frosini and A. Lovato, "Size functions for signature recognition", in "SPIE's International Symposium on Optical Science, Engineering, and Instrumentation", pp. 178–183 (International Society for Optics and Photonics, 1998).

Dosovitskiy, A., J. T. Springenberg, M. Riedmiller and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks", in "Advances in Neural Information Processing Systems", pp. 766–774 (2014).

Draper, B., M. Kirby, J. Marks, T. Marrinan and C. Peterson, "A flag representation for finite collections of subspaces of mixed dimensions", Linear Algebra and its Applications **451**, 15–32 (2014).

Edelman, A., T. A. Arias and S. T. Smith, "The geometry of algorithms with orthogonality constraints", SIAM journal on Matrix Analysis and Applications **20**, 2, 303–353 (1998).

Edelsbrunner, H., D. Letscher and A. Zomorodian, "Topological persistence and simplification", in "Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on", pp. 454–463 (IEEE, 2000).

Edelsbrunner, H., D. Letscher and A. Zomorodian, "Topological persistence and simplification", Discrete & Computational Geometry **28**, 4, 511–533 (2002).

Fasy, B. T., J. Kim, F. Lecci and C. Maria, "Introduction to the r package tda", arXiv preprint arXiv:1411.1830 (2014).

Fletcher, T., "Terse notes on riemannian geometry", (2010).

Gopalan, R., S. Taheri, P. Turaga and R. Chellappa, "A blur-robust descriptor with applications to face recognition", IEEE transactions on Pattern Analysis and Machine intelligence **34**, 6, 1220–1226 (2012).

Hamm, J. and D. D. Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning", in "Proceedings of the 25th international conference on Machine learning", pp. 376–383 (ACM, 2008).

Junejo, I. N., E. Dexter, I. Laptev and P. Perez, "View-independent action recognition from temporal self-similarities", IEEE transactions on Pattern analysis and machine intelligence **33**, 1, 172–185 (2011).

Kokkinos, I., M. Bronstein and A. Yuille, *Dense scale invariant descriptors for images and surfaces*, Ph.D. thesis, INRIA (2012).

Krim, H., T. Gentimis and H. Chintakunta, "Discovering the whole by the coarse: A topological paradigm for data analysis", IEEE Signal Processing Magazine **33**, 2, 95–104 (2016).

Li, C., M. Ovsjanikov and F. Chazal, "Persistence-based structural recognition", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 1995–2002 (2014).

Lian, Z., A. Godil, T. Fabry, T. Furuya, J. Hermans, R. Ohbuchi, C. Shu, D. Smeets, P. Suetens, D. Vandermeulen *et al.*, "Shrec'10 track: Non-rigid 3d shape retrieval.", 3DOR **10**, 101–108 (2010).

Liu, X., A. Srivastava and K. Gallivan, "Optimal linear representations of images for object recognition", in "Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on", vol. 1, pp. I–I (IEEE, 2003).

Luo, D. and H. Huang, "Video motion segmentation using new adaptive manifold denoising model", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 65–72 (2014).

Marcos, D., M. Volpi and D. Tuia, "Learning rotation invariant convolutional filters for texture classification", in "Pattern Recognition (ICPR), 2016 23rd International Conference on", pp. 2012–2017 (IEEE, 2016).

Maria, C., J.-D. Boissonnat, M. Glisse and M. Yvinec, "The gudhi library: Simplicial complexes and persistent homology", in "International Congress on Mathematical Software", pp. 167–174 (Springer, 2014).

Reininghaus, J., S. Huber, U. Bauer and R. Kwitt, "A stable multi-scale kernel for topological machine learning", (2014).

Rouse, D., A. Watkins, D. Porter, J. Harer, P. Bendich, N. Strawn, E. Munch, J. DeSena, J. Clarke, J. Gilbert *et al.*, "Feature-aided multiple hypothesis tracking using topological and statistical behavior classifiers", in "SPIE Defense+ Security", pp. 94740L–94740L (International Society for Optics and Photonics, 2015).

Sharafuddin, E., N. Jiang, Y. Jin and Z.-L. Zhang, "Know your enemy, know yourself: Block-level network behavior profiling and tracking", in "Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE", pp. 1–6 (IEEE, 2010).

Som, A., N. Krishnamurthi, V. Venkataraman, K. N. Ramamurthy and P. Turaga, "Multiscale evolution of attractor-shape descriptors for assessing parkinson's disease severity", in "IEEE Global Conference on Signal and Information Processing (GlobalSIP)", (2017).

Som, A., N. Krishnamurthi, V. Venkataraman and P. Turaga, "Attractor-shape descriptors for balance impairment assessment in parkinson's disease", vol. 2016-, pp. 3096–3100 (Institute of Electrical and Electronics Engineers Inc., 2016).

Sun, J., M. Ovsjanikov and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion", in "Computer graphics forum", vol. 28, pp. 1383–1392 (Wiley Online Library, 2009).

Sun, X., L. Wang and J. Feng, "Further results on the subspace distance", Pattern recognition **40**, 1, 328–329 (2007).

Takens, F. *et al.*, "Detecting strange attractors in turbulence", Lecture notes in mathematics **898**, 1, 366–381 (1981).

Turaga, P. K., A. Veeraraghavan, A. Srivastava and R. Chellappa, "Statistical analysis on manifolds and its applications to video analysis.", Video search and mining **287**, 115–144 (2010).

Turner, K., Y. Mileyko, S. Mukherjee and J. Harer, "Fréchet means for distributions of persistence diagrams", Discrete & Computational Geometry **52**, 1, 44–70 (2014).

Venkataraman, V., K. N. Ramamurthy and P. Turaga, "Persistent homology of attractors for action recognition", in "Image Processing (ICIP), 2016 IEEE International Conference on", pp. 4150–4154 (IEEE, 2016).

Venkataraman, V. and P. Turaga, "Shape distributions of nonlinear dynamical systems for video-based inference", IEEE transactions on Pattern analysis and machine intelligence **38**, 12, 2531–2543 (2016).

Wang, L., X. Wang and J. Feng, "Subspace distance analysis with application to adaptive bayesian algorithm for face recognition", Pattern recognition **39**, 3, 456–464 (2006).

Weinland, D., E. Boyer and R. Ronfard, "Action recognition from arbitrary views using 3d exemplars", in "Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on", pp. 1–7 (IEEE, 2007).

Yan, J. and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate", in "European conference on computer vision", pp. 94–106 (Springer, 2006).

Ye, K. and L.-H. Lim, "Schubert varieties and distances between subspaces of different dimensions", SIAM Journal on Matrix Analysis and Applications **37**, 3, 1176–1197 (2016).

Zomorodian, A., "Fast construction of the vietoris-rips complex", Computers & Graphics **34**, 3, 263–271 (2010).

Zomorodian, A. and G. Carlsson, "Computing persistent homology", Discrete & Computational Geometry **33**, 2, 249–274 (2005).