Persistence-based Structural Recognition

Chunyuan Li, Maks Ovsjanikov, Frederic Chazal

Persistence-based Structural Recognition

Chunyuan Li[1,3], Maks Ovsjanikov[2], Frederic Chazal[1]
[1] Geometrica, INRIA Saclay, France [2] LIX, Ecole Polytechnique, Palaiseau, France
[3] National Institute of Standards and Technology, MD, USA
chunyuan.li@hotmail.com maks@lix.polytechnique.fr frederic.chazal@inria.fr

Abstract

This paper presents a framework for object recognition using topological persistence. In particular, we show that the so-called persistence diagrams built from functions defined on the objects can serve as compact and informative descriptors for images and shapes. Complementary to the bag-of-features representation, which captures the distribution of values of a given function, persistence diagrams can be used to characterize its structural properties, reflecting spatial information in an invariant way. In practice, the choice of function is simple: each dimension of the feature vector can be viewed as a function. The proposed method is general: it can work on various multimedia data, including 2D shapes, textures and triangle meshes. Extensive experiments on 3D shape retrieval, hand gesture recognition and texture classification demonstrate the performance of the proposed method in comparison with state-of-the-art methods. Additionally, our approach yields higher recognition accuracy when used in conjunction with the bag-of-features.

1. Introduction

Over the years, the *bag-of-features* (BoF) model has been extremely popular for the recognition of text, images and shapes [2, 7, 17, 27, 36]. In the image domain, this approach corresponds to treating a given image as a collection of unordered local descriptors, extracted from feature points, and quantizing them into discrete “visual/geometric words” [40]. The distribution of visual words in an image is then summarized by a fixed-sized vector using various pooling techniques [4]. Due to the compactness and informativeness of the resulting representation, BoF is widely used for learning and recognition.

A large number of extensions of the BoF have been proposed with the aim to recover the geometric (spatial) relations between features, which are discarded by BoF. The two most common approaches include explicitly encoding

[3, 6, 9, 45, 46]

or decomposing the object into spatial subregions and performing pooling in each subregion separately [25]. These approaches, however, assume explicit parametrization of the spatial information that needs to be captured (e.g. distribution of pairwise offsets as in [6, 46] or commute times in [3]) or a pre-defined spatial decomposition to guarantee the stability of the representation under deformations. Thus, devising a compact and informative representation to characterize the structural properties of features while preserving stability, remains a challenging open problem.

Unlike the majority of the previously proposed methods that try to augment the bag-of-features representation to include some notion of spatiality, we propose to use a different approach, which is inherently well-suited to capture structural properties of functions defined on different domains in a stable way. In particular, we propose to complement to the BoF approach, by computing the *persistence diagrams* (PD) of functions defined on different data modalities, including 2D shapes, textures and triangle meshes.

Intuitively, instead of considering each feature point and its associated descriptor vector independently, we analyze all the points together by considering each dimension of the feature vector as a real-valued function, defined on the entire domain. With the assumption that the descriptors are stable and informative enough in the object space, we compute the persistent homology of the resulting functions. The derived PD provides a simple yet powerful description that captures the structural properties of the object. Importantly, unlike the BoF representation which captures the distribution of the values in the function, PD exploits the connectivity between different points in the domain to characterize the *relative* prominence of different feature points.

In order to demonstrate the usefulness of this representation, we argue both theoretically and experimentally that it captures information which is complementary to the information in the bag-of-features representation. In particular, we perform extensive experiments on 3D shape retrieval, hand gesture recognition and texture classification, where we demonstrate state-of-the-art performance by using the PDs in conjunction with the bag-

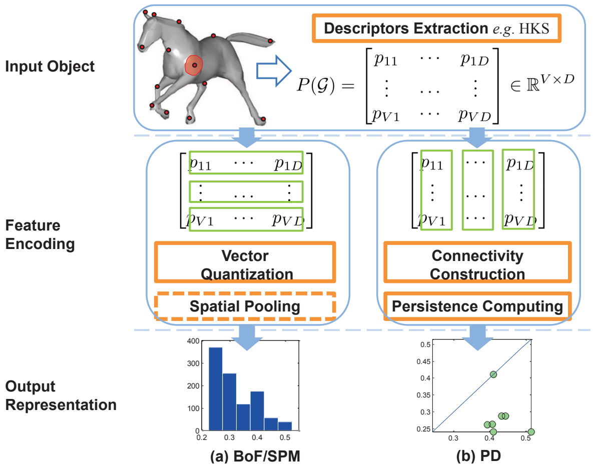

Input Object
Descriptors Extraction e.g. HKS
$P(\mathcal{G}) = \begin{bmatrix} p_{11} & \cdots & p_{1D} \\ \vdots & \dots & \vdots \\ p_{V1} & \cdots & p_{VD} \end{bmatrix} \in \mathbb{R}^{V \times D}$
Feature Encoding
Vector Quantization
Spatial Pooling
Connectivity Construction
Persistence Computing
Output Representation
400
300
200
100
0
0.2
0.3
0.4
0.5
0.5
0.45
0.4
0.35
0.3
0.25
0.3
0.4
0.5
(a) BoF/SPM
(b) PD

Figure 1. Schematic comparison of BoF with PD.

argue that this is largely due to the fact that persistence diagrams allow to distinguish feature configurations that are structurally different. Schematic comparison between the BoF with PD is shown in Fig. 1.

The rest of the paper is organized as follows. In Sect. 2, we discuss some related works. Sect. 3 presents the framework of our proposed algorithm. Experimental results in three applicable scenarios are provided in Sect. 4. Finally, we conclude in Sect. 5.

2. Related Work

The past decade has witnessed a surge in popularity of BoF and its extensions for solving various computer vision problems. Among them, one particularly successful method is to form a global representation by spatially aggregating the local descriptors pioneered by Lazebnik *et al.* [25]. The subsequent research on spatial aggregation get compelling performance because of two seemingly independent advantages: the better design of (1) spatial regions and (2) aggregating operator. The spatial pyramids manually define the multi-scale grid-structured regions over the image space, and many excellent visual recognition methods either directly use them [25, 44], or modify the spatial decomposition to fit their data [9,26]. Recently, Jia *et al.* [23] proposed to learn more adaptive regions by the receptive fields. Popular aggregating (pooling) strategies include averaging and max pooling, which have been used by Lazebnik *et al.* [25] and Yang *et al.* [44] respectively. Coates *et al.* proposed to aggregate over multiple features in the context of deep learning [14].

Our work relies on the theory of persistent homology, which falls under the umbrella of topological data analysis (TDA).It was first formalized by Edelsbrunner *et al.* [19] by building on earlier notions of size functions used by Frosini *et al.* for shape analysis [41] and later developed in [11, 19, 47]. Persistence diagrams appear prominently in TDA, and in particular provide an efficient way to encode topological properties of real-valued functions defined over spaces. PDs have provable stability properties [11, 15], and allow to infer robust topological information on the studied data. Persistent homology has been successfully applied to clustering tasks [13], vision tasks such as shape segmentation [37], component detection [30] and recognition [10, 12, 20]. Our work is inspired by [10, 12] but is different in that we exploit a given connectivity on the object and use feature functions to build the persistence diagrams rather than using point samples and simplicial complex filtrations. We also show that multiple PDs of different functions as well as the BoF representation can be combined to improve recognition tasks in a variety of scenarios. In this paper we focus on 0-dimensional persistent homology that encodes connectivity information in an intuitive and easy-to-compute way.

3. Topological Object Representation Using Persistence Diagram

3.1. Feature Encoding: From BoF to PD

We assume that every object (e.g., image, 2D shape or 3D mesh) is represented as a connectivity graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where the set of nodes $\mathcal{V}$ of size V are samples on the object, and the set of edges $\mathcal{E}$ represent neighborhood relations between samples. For example, considering images as grids of points, we use the 4-neighborhood to define the edges in $\mathcal{E}$. Similarly, for 3D objects, we use the vertices and the edges of the faces in the mesh to define $\mathcal{G}$.

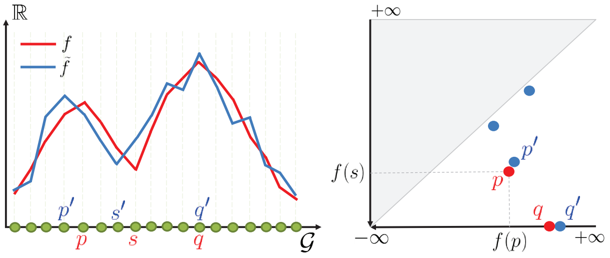
Similarly to the BoF approach we start by associating a D-dimensional feature vector to every sample in $\mathcal{V}$ resulting in a $V \times D$ matrix:

$$P(\mathcal{G}) = \begin{bmatrix} p_{11} & \cdots & p_{1D} \\ \vdots & \dots & \vdots \\ p_{V1} & \cdots & p_{VD} \end{bmatrix} \in \mathbb{R}^{V \times D}.$$

While the BoF approach can be thought of as analyzing P by considering its rows, which correspond to point descriptors, we propose to consider P column-wise by interpreting each column as a real-valued function defined on the nodes of $\mathcal{G}$. The persistence diagrams of these functions allow us to capture the structural properties of the object and in particular robustly encode the relative prominence of the different feature points.

3.2. Persistence Diagram

Given a real-valued function f defined on the nodes of $\mathcal{G}$, the (0-dimensional) persistence diagram of f encodes the evolution of the connectivity of the 1-parameter family of superlevel-sets $\mathbb{F}^\alpha = f^{-1}([\alpha, +\infty))$ as α goes from $+\infty$ to $-\infty$ in the following way. A node $v \in \mathcal{V}$ is called a *peak*



(a) (b)

Figure 2. Sketch of persistence computing: (a) a smooth function f maps the nodes of $\mathcal{G}$ to $\mathbb{R}$, $\tilde{f}$ is a noisy version of f; (b) superimposition of PDs of f (red) and $\tilde{f}$ (blue), showing the one-to-one correspondence between the prominent peaks of f and $\tilde{f}$.

if it is a local maximum of f, *i.e.*, if $f(v) \geq f(u)$ for any u such that (v, u) is an edge of $\mathcal{G}$. For any peak v we say that v is *born* at $f(v)$. For $\alpha \leq f(v)$, let $\mathcal{C}(v, \alpha)$ be the connected subgraph in $\mathbb{F}^\alpha \subseteq \mathcal{G}$ that contains v. The infimum of α such that $f(v)$ is the global maximum of f restricted to $\mathcal{C}(v, \alpha)$ is called the *death* of v. See Fig. 2 for an example where $\mathcal{G}$ represents a line-graph (in green) and the node p of $\mathcal{G}$ is a peak given the function f shown in red in Fig. 2 (a). A new connected subgraph is born in the superlevel-set $\mathbb{F}^\alpha$ when $\alpha = f(p)$, and it dies when $\alpha = f(s)$.

The lifespan of peak v is thus determined by its birth $f_a = f(v)$ and death $f_b \leq f_a$. This allows us to associate a point (f_a, f_b) on a 2D plane to each peak v. Collecting all these points, we obtain the *persistence diagram* (PD) of f (illustrated in Fig. 2(b)). The difference $\tau = f_a - f_b \geq 0$ is called the *prominence* (or *persistence*) of the peak v. Note that the prominence of v is equal to the vertical distance of its corresponding point in the PD to the diagonal. Lastly, the death time of the global maxima of f, i.e. the peaks v that satisfy $f(v) = \max f$, is set to $\min f$, and associated to the point $(\max f, \min f)$ in the PD. Overall, the PD encodes the relative prominence of the different peaks of a given function by considering the connectivity information in the domain.

Computation. In practice the persistence diagram can be simply computed using the Union-Find algorithm [16] by sorting the nodes of $\mathcal{G}$ according to their function value and keeping track of the corresponding connected components.

Stability. One important property of the persistence diagram is that it is stable under some perturbations of the feature function. Intuitively, the stability theorem in [11, 15] states that if the values of a function f are perturbed by no more than some $\varepsilon > 0$, then the points on the persistence diagrams will also be perturbed by no more than ε. In Fig. 2(a), the blue function $\tilde{f}$ is a noisy version of the red function f. The PDs of f and $\tilde{f}$ are plotted in Fig. 2(b). These two PDs are similar, in the sense that the distance between points that are far from the diagonal is small, even if some extra points with low prominence can appear.

Invariance. Similarly to the BoF representation, PD is invariant to rotations, translations and scaling of the object provided that the feature function remains the same. However, unlike the BoF which is invariant to any permutation of the nodes, the PD is only invariant under continuous deformations. In particular, given two objects represented as graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ and a map T between them, for any function f on $\mathcal{G}_2$ the PDs of f and $f \circ T$ are the same if T is an isomorphism. Conversely, if T is not an isomorphism then there exists a function f whose PD will be different from the one of $f \circ T$. Intuitively this implies that PDs capture local spatial information through the connectivity structure.

Relation to spatial pyramid approaches. Note that PDs can be seen as alternatives to spatial decomposition methods, and espacially Spatial Pyramid Matching SPM [25] and ScSPM [44], that apply BoF approach to each region of a multi-scale decomposition of the object and often use max pooling to characterize the features in each region. Unlike these approaches, PDs do not require an explicit decomposition of the object, and allow to capture structural properties of functions in a robust and compact way.

3.3. PDs for object comparison

In this paper we propose to use persistence diagrams of different feature functions as descriptors for object recognition and classification. To illustrate this idea, we compute the PD for several 3D meshes in Fig. 3 where we use the Heat Kernel Signature (HKS) [21, 39] for fixed time as the feature function. The first row displays the HKS value for the four models: two horses, a dinosaur and pliers. The second row shows the corresponding PDs.

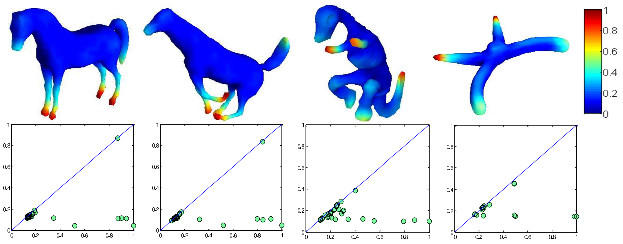


Figure 3. HKS function and its persistence diagrams.

Note that in this case the PD is invariant to isometric deformations because the underlying HKS feature is an isometry invariant. The PDs allow us to distinguish these 3 classes of objects since the diagrams associated to the horses are more similar to each other to those of the other objects.

Quantitatively, we use the following construction to compare PDs. For two functions f and g with respective PDs $\mathcal{D}_1$ and $\mathcal{D}_2$, we construct a weighted bipartite graph B between $\mathcal{D}_1 \cup \Pi(\mathcal{D}_2)$ and $\mathcal{D}_2 \cup \Pi(\mathcal{D}_1)$ where Π is the projection onto the diagonal, i.e. for a point $(a, b) \in \mathbb{R}^2$, $\Pi(a, b) = (\frac{a+b}{2}, \frac{a+b}{2})$. This can guide the points with smaller persistence corresponding to the points on diagonal, and thus remove their interference in prominent point matching. The weight of an edge $(u, v) \in B$ is defined as $\|v - u\|_\infty$ if u or v is in $\mathcal{D}_1 \cup \mathcal{D}_2$ and is set to 0 if both u and v are in $\Pi(\mathcal{D}_1) \cup \Pi(\mathcal{D}_2)$. The *bottleneck distance* $\mathrm{d}_{\text{btnk.}}(\mathcal{D}_1, \mathcal{D}_2)$ is defined as the minimum value d such there exists a perfect matching in B with all edges having weights less than d. Similarly, the *degree-p Wasserstein distance* $\mathrm{d}_{\text{wsst.}}$ is defined as the minimum value d such there exists a perfect matching in B whose the sum of the weights raised to the power p of all edges is less than d^p. As a consequence, computing both distances boils down to computing maximum matchings in bipartite graphs. This can be done efficiently using e.g. the Hungarian algorithm for PDs with few points. However, this might become prohibitive for PDs with many points.

For this reason we also use an alternative representation of PDs, the persistent landscape [8], which allows us to represent a PD as a point in a high dimensional Euclidean space. Given a PD with points $\{(a_i, b_i)\}_{i=1}^n$, $a_i \geq b_i$. For each point (a_i, b_i), let $f_{(a_i,b_i)} : \mathbb{R} \to \mathbb{R}$, $f_{(a_i,b_i)}(t) = \min(a_i - t, t - b_i)_+$, where z_+ denotes $\max(z, 0)$. The *persistence landscape* is defined as the set of functions $\beta_k : \mathbb{R} \to \mathbb{R}, k \in \mathbb{N}$ given by

$$\beta_k(t) = k\text{th largest value of } \left\{f_{(a_i,b_i)}(t)\right\}_{i=1}^n \qquad (1)$$

with $\beta_k(t) = 0$ if $k > n$. Given k we consider the first k landscape functions, and discretize them with m sampling points along t. The derivatives of the sampling points are calculated and form a mk-size vector. The *landscape distance* $\mathrm{d}_{\text{ldsp.}}^{k,m}$ is defined as the L_p norm between these vectors.

3.4. Metric Learning for Multiple Functions

In a bid to achieve higher performance, multiple features are usually used. When these features are designed in similar ways, the final distance can be simply chosen as the sum of their persistence based distances. However, when these features are designed in very different ways, it is more reliable to give different weights to different features. We use labeled data from the dataset to learn these parameters. Suppose we have some set of objects. Each object x_i is characterized by D functions as $F(x_i) = \{f_d(x_i)\}_{d=1}^D$, and $\mathrm{d}_d(x_i, x_j)$ is the persistence based distance of the dth function between two objects x_i and x_j from the set. Additionally, we are given information that certain pairs of them are "similar":

$$\mathcal{S} : (x_i, x_j) \in \mathcal{S} \text{ if } x_i \text{ and } x_j \text{ are similar} \qquad (2)$$

and certain pairs of them are "dissimilar"

$$\mathcal{D} : (x_i, x_j) \in \mathcal{D} \text{ if } x_i \text{ and } x_j \text{ are dissimilar} \qquad (3)$$

Our goal is to learn a distance metric to respect that "similar" objects end up with smaller distances to each other, and "dissimilar" objects with larger distances to each other. We consider learning a distance metric of the form

$$Dis(x_i, x_j) = \|x_i - x_j\|_A = \sqrt{\mathbf{d}(x_i, x_j)^T A \mathbf{d}(x_i, x_j)} \qquad (4)$$

where $\mathbf{d}(x_i, x_j) = [\mathrm{d}_1(x_i, x_j), \ldots, \mathrm{d}_d(x_i, x_j), \ldots, \mathrm{d}_D(x_i, x_j)]$. The problem can be formulated as an optimization programming

$$\begin{aligned} \min\nolimits_A \textstyle\sum_{x_i, x_j \in S} \|x_i - x_j\|_A^2 \\ s.t. \textstyle\sum_{x_i, x_j \in \mathcal{D}} \|x_i - x_j\|_A \geq 1, A \geq 0 \end{aligned} \qquad (5)$$

This is a standard formulation of distance metric learning, and can be solved by the algorithms in [18, 43]. The learned metric A interprets the contribution of different features. It is further embedded as part of the pipeline to provide a more reliable measure for the rest of testing data in the dataset.

3.5. Limitations

While PDs allow to capture the relative prominence of the local extrema of different functions, we note that they do not provide any information about the distribution of other (non-critical) values. Thus, the information contained in PDs is largely *complementary* to the information captured in the BoF representations. Thus, as we show in the following section, we can often get superior performance by combining the two approaches.

4. Experimental Results

We present the results of our PD-based object recognition scheme on various datasets of three different types: 3D meshes (Section 4.1), 2D gesture contours (Section 4.2), and texture images (Section 4.3). The method was implemented in MATLAB, and the experiments were performed on a computer with an Intel Core processor running at 1.4 GHz with 4 GB RAM. In each experiment we use 3 randomly selected samples from each class to learn the weights in the metric.

4.1. 3D Shape Retrieval

Our first set of experiments is on triangulated 3D mesh retrieval. We consider two datasets: SHREC 2010 [28] and a robustness dataset [5]. The former focuses on near-isometric shapes, and consists of 200 watertight shapes equally divided into 10 categories (Figure 4 illustrates a sampling of the shapes from this dataset). The latter has the same data used in the Shape Google [33] with an extra transformation, which is the same as SHREC Robustness benchmark [5] except for a few transformations. It contains 596 shapes classified into 13 classes, and 456 shapes which do not belong to any class. For both datasets, we use the spectral descriptors as feature functions, including the heat kernel signature (HKS) [21, 39], wave kernel signature (WKS) [1] and scale invariant heat kernel signature (SIHKS) [24]. All of them are isometry-invariant. HKS and WKS characterize the macroscopic and microscopic properties of shapes, respectively. SIHKS is a scale-invariant version of HKS. We compare our method to BoF-based approach, spatially-sensitive BoF (SSBoF) [7] and ISPM [26], which is a spatial pyramid approach on surfaces.

Figure 4. Sample shapes from SHREC 2010.

Results on SHREC 2010

On this dataset, we simplify each mesh into 2000 faces, and the retrieval results are evaluated using the measures provided by the organizers, including NN, FT, ST, E-measure, and DCG (See [28] for details). We first illustrate the performance of different methods by considering the HKS and WKS functions for fixed time and energy parameters respectively. DCG and the average running time to compare two PDs are reported in Table 1. The average number of points on PDs ranges around 15 for HKS and 100 for WKS. It can be observed on this example, that bottleneck and Wasserstein distances yield a significantly higher accuracy than the landscape distance, at the price of a higher computation time. Fig. 5 displays an example where PD provides more discriminative information than the histogram of values of the function. The extremities of the objects all correspond to local maxima of the HKS function. This structural information is well preserved in PDs as the prominent points, but lost in the histogram representation that merges function values from different components into the same bins.

Table 1. Comparison between different distances.

Metrics	HKS		WKS	
	DCG	time (s)	DCG	time (s)
Bottleneck	0.8746	0.0020	0.5988	0.0315
Wasserstein ($p = 2$)	0.8774	0.0060	0.6190	0.5354
Wasserstein ($p = 3$)	0.8781	0.0061	0.6117	0.6069
Landscape ($K = 10, N = 100$)	0.8553	0.0046	0.4935	0.0071
Landscape ($K = 10, N = 50$)	0.8471	0.0026	0.4578	0.0037

Based on these experiments, for comparison of 3D shapes, we use the bottleneck distance to compare PDs, as it provides a good trade-off between time and efficiency whenever the diagrams do not contain too many points.

We first compare our PD-based technique with BoF and ISPM methods using multi-dimensional descriptors. In particular, we use 17-dimensional SIHKS, 10-dimensional HKS and 10-dimensional WKS. In this case we do not use the metric learning step as the different dimensions are commensurable. Throughout this experiment with use 32 words to construct the vocabulary for the BoF representation and use soft vector quantization. We summarize the results of SIHKS and HKS in Table 2. As can be seen in this table, the PD-based method consistently outperforms the BoF representation across different feature functions.

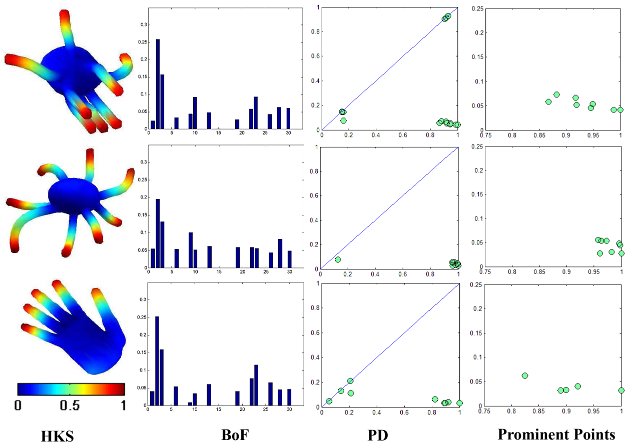


Figure 5. An example that PD discriminates different shapes, while BoF fails.

We then combine SIHKS, HKS and WKS to form a single high-dimension descriptor. The combined performance of our method again outperforms the BoF approach, in part because PDs allow to capture structural properties of objects which are discarded by the BoF representation.

We also show that the advantages of the BoF and PD-based approaches can be combined by considering the joint representation where $d_{\mathtt{BoF+PD}} = \lambda * d_{\mathtt{BoF}} + d_{\mathtt{PD}}$, where λ is the relative weight to normalize the distances. On this dataset, we manually picked λ of SIHKS, HKS, WKS and the high-dimension spectral descriptor to be 0.033, 0.2, 200, 0.1 for BoF+PD, and 0.2, 1, 2000, 0.2 for ISPM+PD, respectively. The combined metric PD+BoF/ISPM always yields the the highest results.

We also compare with the state-of-the-art method, including ShapeDNA [35] and DM-EVD [38], which show the best performance in a larger contest [29]. We test the ShapeDNA method on the the same simplified meshes, and take the result of DM-EVD and other methods from [28], in which DM-EVD is performed on the fine meshes. As can be seen in Table 2, our combined result is comparable to these methods.

Results on TOSCA-based dataset

On the TOSCA-based dataset [33], we use multi-dimensional descriptors: HKS and SIHKS with 6 and 5 dimensions respectively. Table 3 summarizes the results obtained using our method compared with the performance of BoF, SSBoF, ISPM and PD in terms of equal error rate (EER) [33]. Overall, the PD-based approach alone shows slightly worse performance than the BoF methods on this dataset. However, we obtain superior performance by combining the two methods. On this dataset, we use the weights: $\lambda = 10, 5, 10$ for HKS, SIHKS and HKS+SIHKS respectively in BoF+PD and $50, 10000, 10$ for HKS, SIHKS

Table 2. Comparison of PD, BoF, SSBoF and ISPM on SHREC 2010 [28].

Methods		NN	1-Tier	2-Tier	e-Measure	DCG
HKS	BoF	0.9100	0.4811	0.6374	0.4492	0.8061
	SSBoF	0.9150	0.4737	0.6321	0.4412	0.8040
	ISPM	0.9300	0.5745	0.7018	0.5018	0.8597
	PD	0.9600	0.5811	0.7034	0.4971	0.8677
	BoF + PD	0.9500	0.6095	0.7382	0.5237	0.8781
	ISPM + PD	0.9700	0.6321	0.7500	0.5335	0.8869
SIHKS	BoF	0.9700	0.7145	0.8308	0.6020	0.9191
	SSBoF	0.9700	0.7097	0.8297	0.5967	0.9164
	ISPM	0.9700	0.7750	0.8734	0.6359	0.9315
	PD	0.9850	0.8532	0.9697	0.7045	0.9740
	BoF + PD	0.9850	0.8534	0.9697	0.7051	0.9737
	ISPM + PD	0.9900	0.8784	0.9724	0.7114	**0.9778**
WKS+ HKS+ SIHKS	BoF	0.9700	0.7426	0.8758	0.6275	0.9367
	SSBoF	0.9750	0.7318	0.8511	0.6102	0.9292
	ISPM	0.9750	0.8021	0.8979	0.6533	0.9408
	PD	0.9850	0.8532	0.9697	0.7045	0.9740
	BoF + PD	0.9850	0.8561	0.9705	0.7061	0.9743
	ISPM + PD	0.9900	0.8779	0.9742	0.7118	0.9778
ShapeDNA [35]		0.9850	0.7974	0.9203	0.6653	0.9536
BOF-dSIFT-ERC-Tree		0.9850	0.9092	0.9632	0.7055	0.9763
DM-EVD [38]		1.0000	0.8611	0.9571	0.7012	0.9773
Canonical Forms		0.9200	0.6347	0.7800	0.5527	0.8781

Table 3. Comparison of PD, BoF, SSBoF and ISPM on TOSCA-based dataset.

Descriptors	Transformation	BoF	SSBoF	ISPM	PD	BoF+PD	ISPM+PD
HKS+ SIHKS	isometry	0.0352	0.0307	0.0284	0.1464	0.0369	0.0274
	isometry+topology	0.0394	0.0320	0.0341	0.1654	0.0397	0.0307
	noise	0.1540	0.1464	0.1036	0.2087	0.1462	0.1019
	null	0.0412	0.0394	0.0299	0.1561	0.0428	0.0291
	partiality	0.0507	0.0466	0.0456	0.2630	0.0521	0.0468
	topology	0.0561	0.0572	0.0350	0.2173	0.0563	0.0362
	triangulation	0.0520	0.0495	0.0374	0.2049	0.0562	0.0396
	All	0.0535	0.0519	0.0417	0.1727	0.0527	**0.0380**
HKS	All	0.0650	0.0712	0.0595	0.2477	0.0638	0.0581
SIHKS	All	0.0670	0.0531	0.0507	0.1728	0.0658	0.0507

and HKS+SIHKS respectively in ISPM+PD. The EER of ShapeDNA for all transformations is 0.0967, which is worse than state-of-the-art work Shape Google (BoF+spectral descriptors) [33] on this dataset. The proposed method can help Shape Google and its spatial version (ISPM) achieve even better results.

4.2. Hand Gesture Recognition

Our second experiment focuses on the recent Kinect sensor-based hand gesture recognition [34] and is inspired by the work of [41]. This dataset contains 1000 gestures equally categorized into 10 classes. Each gesture is compared against a set of *20 templates* (2 samples for each class) and nearest neighbor classification is performed.

Each gesture instance is represented as a sequence of contour vertices, starting from the left side of the wrist (yellow point in Fig. 6(a)). We construct our graph by considering the sequence points as nodes, and their adjacent connectivity as edges. Ren *et al.* [34] also provide an informative function to characterize each gesture by first computing the center point, f_{center}, defined as the point with the maximal Distance Transform value (red point in Fig. 6(a)), and computing the normalized Euclidean distance from the contour vertices to the center. We show the corresponding PD in Fig. 6(d). Importantly, due to the stability property, the PD of f_{center} can represent the perturbations of the contour as points close to diagonal, and thus help to alleviate noise brought by the Kinect sensor in data acquisition.

In all of the experiments below we use the Wasserstein distance to compare persistence diagrams of functions in this dataset.

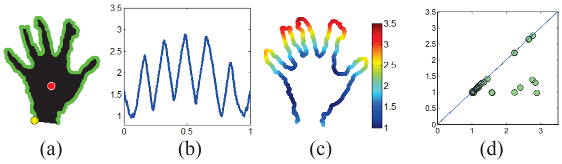
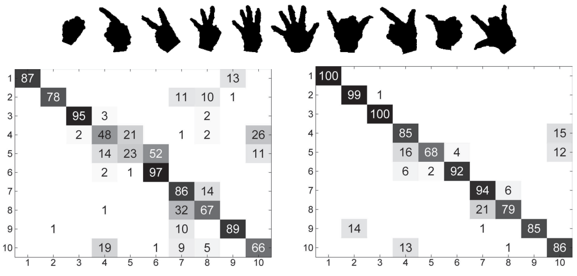


Figure 6. Persistence computing on gestures. (a) a gesture with its center point in red and starting point in yellow; (b) *distance-to-center* function f_{center}; (c) gesture colored by f_{center}; (d) PD of f_{center}.



	1	2	3	4	5	6	7	8	9	10
1	87								13	
2		78					11	10	1	
3			95	3				2		
4			2	48	21		1	2		26
5				14	23	52				11
6				2	1	97				
7							86	14		
8				1			32	67		
9		1					10		89	
10				19		1	9	5		66

(a) BoF

	1	2	3	4	5	6	7	8	9	10
1	100									
2		99	1							
3			100							
4				85						15
5				16	68	4				12
6				6	2	92				
7							94	6		
8							21	79		
9		14					1		85	
10				13				1		86

(b) PD

Figure 7. The confusion matrix of (a) BoF, and (b) PD. The gesture classes are displayed on top.

With f_{center} alone, BoF achieves its best performance of 72.60% when discretizing the function into 5 bins, while PD gets much higher result of 87.60% (see Fig. 7). This suggests that the information lost by discarding the connectivity in BoF is critical for the recognition of objects with different structural properties.

To improve the performance, we include two more functions by computing the eingenfunctions of the PCA performed on vertex coordinates. We denote the two eigenfunctions as f_1 and f_2 and their absolute values as $|f_1|, |f_2|$. Using $f_{\text{center}}, |f_1|$, and $|f_2|$, together, the BoF reaches its best result 83.20% with 15 visual words learned using k-means. Our method with metric learning solved by [18, 43] gets accuracy of 92.90% and 91.70%, respectively. Furthermore, BoF+PD combined using $\lambda = 12$ yields recognition accuracy of 93.50%.

To compare our method to the state-of-the-art, we use f_1 and f_2 directly and compute the distance by testing two sign possibilities *i.e.*,

$$\hat{\mathrm{d}}_i = \min \left\{ \mathrm{d}_{\text{wsst.}} \left(\mathcal{D}(f_i), \mathcal{D}(g_i) \right), \mathrm{d}_{\text{wsst.}} \left(\mathcal{D}(f_i), \mathcal{D}(-g_i) \right) \right\}, i = 1, 2.$$

where f_i and g_i are the i^{th} PCA eigenfunctions of the two gestures. For comparison, we also consider the classification with only one labeled sample per class (*10 templates*). Recent methods proposed by Ren *et al.* [34] and Wang *et al.* [42] are based on shape segmentation and finger earth mover distance (FEMD) introduced by Ren *et al.* [34]. As claimed by the authors, this metric is robust to the finger-touching, which also means that adding more labeled gestures with finger-touching into the training set will not sig-

Table 4. Comparison on hand gesture database [34].

Methods	Accuracy (%)	Time (s)
Shape Context without bending cost	83.2	12.346
Shape Context with bending cost	79.1	26.777
Skeleton Matching	78.6	2.4449
Near-convex Decomposition+FEMD [34]	93.9	4.0012
Thresholding Decomposition+FEMD [34]	93.2	0.075
PSD+FEMD [42]	94.1	1.967
f_{center} (*10 templates*)	86.4	0.075
Multiple functions (*10 templates*)	90.1	0.3750
f_{center} (*20 templates*)	87.6	0.1057
Multiple functions (*20 templates*)	95.4	0.5285

nificantly improve accuracy. We validate it using PSD [42] for the settings that 1, 2 or 3 labeled samples are respectively chosen per class. For each setting, 3 trials with different labeled samples (with finger touching) are tested, the mean accuracy of PSD is 94.1%, 93.2%, and 93.1%. We show the comparison with the state-of-the-art in Table 4, the results of other methods are from [34]. PD can reach higher accuracy with more training samples, without sacrificing efficiency, because it avoids the time-consuming shape decomposition stage. With the same training set *20 templates*, persistence based method is 95.40%, while PSD is 93.80%.

4.3. Texture Image Classification

Our last experiment is on texture classification. For this we use the standard Outex database [31]. This database consists of 4320 texture images equally categorized into 24 classes. To evaluate our method, we first randomly select 20 images from each class, with one of them acting as a label, and rest are testing samples. We use the CLBP descriptor function [22], which is based on the well-known LBP descriptor [32]. CLBP describes each local region on an image as three discretized components named center (C), sign (S) and magnitude (M). We use CLBP-S, CLBP-MC and CLBP-SMC proposed by Guo *et al.* [22] as the feature functions since these three functions convey different information. CLBP functions are (1) discrete and severely bounded, (2) have strong local self-repetition. Therefore, the associated PDs contain many points. To efficiently discriminate these PDs, we use persistence landscape method. The dissimilarity is defined as a combination of multiple normalized landscape distances, $\mathrm{d} = \sum_{i=1}^{5} (k_i/k_1) * \mathrm{d}_{\text{ldsp.}}^{k_i,n}$, where $k_i = [100, 200, 500, 800, 1000]$ and $n = 21$.

The results are summarized in Table 5. Given these multiple functions, BoF reaches its best result 84.6491% using 100 words. PD with metric learning gives 75.8772%, while the manually chosen relative weights for PD with different functions result in the accuracy 77.4123%. Although higher overall accuracy is always provided by BoF, PD shows its advantage by yielding much higher accuracy in some classes. We show one such example with CLBP-SMC function in Fig. 8. The top two images have larger histogram distance, but their PDs are more similar: fewer points appear on the top-right area, and points in the middle are with higher density. When combining BoF and PD, λ for CLBP-S, CLBP-MC, CLBP-SMC and multiple functions is set as 800,100,165,100, BoF+PD always gives the best performance. Furthermore, we test PD with CLBP-SMC on the whole database, and get accuracy 84.1406%. Its accuracy by BoF is 96.5625%, which is also the reported best in the same settings [22]. By setting $\lambda = 225$, we find that BoF+PD gives an even better result 97.0313%.

Table 5. Comparison between PD and BoF on texture database.

Methods		Accuracy (%)
CLBP-S	BoF	76.0965
	PD	54.8246
	BoF + PD	79.1667
CLBP-MC	BoF	74.1228
	PD	55.2632
	BoF + PD	77.1930
CLBP-SMC	BoF	83.9912
	PD	57.8947
	BoF + PD	85.9649
CLBP-S+ CLBP-MC+ CLBP-SMC	BoF	84.6491
	PD	75.8772
	BoF + PD	87.5000

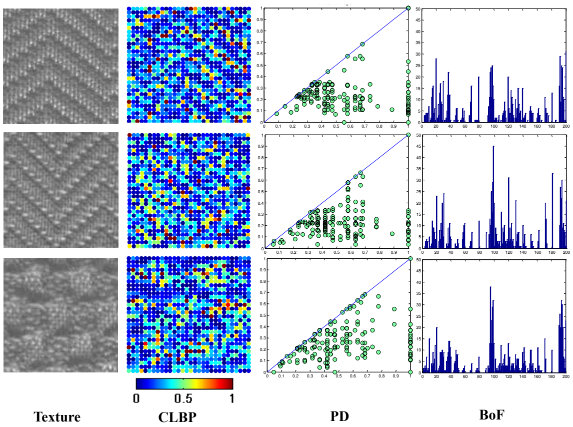


Figure 8. The example of PD and BoF for texture classification.

5. Conclusion and Future Work

In this paper, we present a method for object recognition using topological persistence. Our experiments on a variety of applicable scenarios demonstrate the effectiveness of this approach. PD and BoF capture different information of a function: PD describes it structural properties, while BoF describes the quantity statistic of the function values.

One interesting future direction is to create a principled theoretical framework to combine the PD and BoF for better recognition, and to extend the persistence-based approach to other applications, such as range scans or RGB-D data.

Acknowledgements The authors acknowledge the Marie Curie CIG-334283-HRGP, the CNRS chaire d'excellence, the Google Faculty Research Award, the ANR project TopData ANR-13-BS01-0008 and the ERC project Gudhi.

References

[1] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCVW*, pp. 1626–1633, 2011. 5

[2] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999. 1

[3] R. Behmo, N. Paragios, and V. Prinet. Graph commute times for image representation. In *Proc. CVPR*, pp. 1–8, 2008. 1

[4] Y. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in vision algorithms. In *Proc. ICML*, 2010. 1

[5] A. Bronstein, M. Bronstein, et al. Shrec 2010: robust large-scale shape retrieval benchmark. *Proc. 3DOR*, 5, 2010. 4

[6] A. M. Bronstein and M. M. Bronstein. Spatially-sensitive affine-invariant image descriptors. In *Computer Vision–ECCV 2010*, pp. 197–208. Springer, 2010. 1

[7] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics (TOG)*, 30(1):1, 2011. 1, 5

[8] P. Bubenik. Statistical topology using persistence landscapes. *arXiv preprint arXiv:1207.6437*, 2012. 4

[9] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang. Spatial-bag-of-features. In *Proc. CVPR*, pp. 3352–3359, 2010. 1, 2

[10] G. Carlsson, A. Zomorodian, A. Collins, and L. J. Guibas. Persistence barcodes for shapes. *Int. J. of Shape Modeling*, 11(02):149–187, 2005. 2

[11] F. Chazal, D. Cohen-Steiner, M. Glisse, L. J. Guibas, and S. Oudot. Proximity of persistence modules and their diagrams. In *ACM Symposium of Computational Geometry*, pp. 237–246, 2009. 2, 3

[12] F. Chazal, D. Cohen-Steiner, L. J. Guibas, F. Mémoli, and S. Y. Oudot. Gromov-hausdorff stable signatures for shapes using persistence. In *Proc. SGP*, pp. 1393–1403, 2009. 2

[13] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. Persistence-based clustering in riemannian manifolds. In *Proc. SoCG*, pp. 97–106, 2011. 2

[14] A. Coates and A. Ng. Selecting receptive fields in deep networks. In *NIPS*, pp. 2528–2536, 2011. 2

[15] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Disc. & Comp. Geom.*, 37(1):103–120, 2007. 2, 3

[16] T. Cormen, C. Stein, R. Rivest, and C. Leiserson. *Introduction to Algorithms*. McGraw-Hill, 2001. 3

[17] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshops*, volume 1, pp. 1–2, 2004. 1

[18] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *Proc. ICML*, pp. 209–216, 2007. 4, 6

[19] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In *41st Annual Symposium on Foundations of Computer Science*, pp. 454–463, 2000. 2

[20] P. Frosini and C. Landi. Size theory as a topological tool for computer vision. *Pattern Recogn. and Image Analysis*, 9(4):596–603, 1999. 2

[21] K. Gebal, J. A. Bærentzen, H. Aanæs, and R. Larsen. Shape analysis using the auto diffusion function. In *Computer Graphics Forum*, volume 28, pp. 1405–1413. Wiley Online Library, 2009. 3, 5

[22] Z. Guo, L. Zhang, and D. Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Trans. on Image Processing*, 19(6):1657–1663, 2010. 7

[23] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *Proc. CVPR*, pp. 3370–3377, 2012. 2

[24] I. Kokkinos, M. Bronstein, and A. Yuille. Dense scale invariant descriptors for images and surfaces. *INRIA Report 7914*, 2012. 5

[25] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, pp. 2169–2178, 2006. 1, 2, 3

[26] C. Li and A. B. Hamza. Intrinsic spatial pyramid matching for deformable 3d shape retrieval. *IJMIR*, 2013. 2, 5

[27] F.-F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. CVPR*, pp. 524–531, 2005. 1

[28] Z. Lian, A. Godil, et al. Shrec 2010 track: non-rigid 3d shape retrieval. 2010. 4, 5, 6

[29] Z. Lian, A. Godil, et al. Shrec'11 track: Shape retrieval on non-rigid 3d watertight meshes. *3DOR*, 11:79–88, 2011. 5

[30] R. Litman, A. M. Bronstein, and M. M. Bronstein. Diffusion-geometric maximally stable component detection in deformable shapes. *Computers & Graphics*, 35(3):549–560, 2011. 2

[31] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen. Outex-new framework for empirical evaluation of texture analysis algorithms. In *Proc. ICPR*, pp. 701–706, 2002. 7

[32] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on PAMI*, 24(7):971–987, 2002. 7

[33] M. Ovsjanikov, A. M. Bronstein, M. M. Bronstein, and L. J. Guibas. Shape google: a computer vision approach to isometry invariant shape retrieval. In *Proc. ICCVW*, pp. 320–327, 2009. 4, 5, 6

[34] Z. Ren, J. Yuan, J. Meng, and Z. Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE Trans. on Multimedia*, 2013. 6, 7

[35] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace–beltrami spectra as 'shape-dna' of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006. 5, 6

[36] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, pp. 1470–1477, 2003. 1

[37] P. Skraba, M. Ovsjanikov, F. Chazal, and L. Guibas. Persistence-based segmentation of deformable shapes. In *Proc. CVPRW*, pp. 45–52, 2010. 2

[38] D. Smeets, T. Fabry, J. Hermans, D. Vandermeulen, and P. Suetens. Isometric deformation modelling for object recognition. In *Computer Analysis of Images and Patterns*, pp. 757–765. Springer, 2009. 5, 6

[39] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum*, volume 28, pp. 1383–1392, 2009. 3, 5

[40] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek. Visual word ambiguity. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(7):1271–1283, 2010. 1

[41] A. Verri, C. Uras, P. Frosini, and M. Ferri. On the use of size functions for shape analysis. *Bio. Cybern.*, 70(2):99–107, 1993. 2, 6

[42] C. Wang, W. Liu, Z. Lai, and H. Wang. Perceptually friendly shape decomposition by resolving segmentation points with minimum cost. *JVCI*, 24(3):270–282, 2013. 6, 7

[43] E. P. Xing, M. I. Jordan, S. Russell, and A. Ng. Distance metric learning with application to clustering with side-information. In *Proc. CVPR*, pp. 505–512, 2002. 4, 6

[44] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proc. CVPR*, pp. 1794–1801, 2009. 2, 3

[45] S. Zhang, Q. Tian, G. Hua, Q. Huang, and S. Li. Descriptive visual words and visual phrases for image applications. In *Proc. ACM MM*, pp. 75–84, 2009. 1

[46] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *Proc. CVPR*, pp. 809–816, 2011. 1

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