

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/220184682>

# Shape Google: Geometric Words and Expressions for Invariant Shape Retrieval

Article in ACM Transactions on Graphics · January 2011

DOI: 10.1145/211297.2 · Source: DBLP

---

CITATIONS  
446

READS  
496

4 authors, including:



Michael Bronstein  
University of Lugano  
255 PUBLICATIONS 16,770 CITATIONS

[SEE PROFILE](#)



Leonidas J. Guibas  
Stanford University  
623 PUBLICATIONS 56,509 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Inverse Procedural Modeling of Structured Shapes [View project](#)



Efficient Deformable Shape Correspondence via Kernel Matching [View project](#)

# Shape Google: geometric words and expressions for invariant shape retrieval

ALEXANDER M. BRONSTEIN

Department of Electrical Engineering, Tel-Aviv University

MICHAEL M. BRONSTEIN

Institute of Computational Science, Faculty of Informatics, Università della Svizzera Italiana

LEONIDAS J. GUIBAS

Department of Computer Science, Stanford University

and MAKS OVSJANIKOV

Institute for Computational and Mathematical Engineering, Stanford University

The computer vision and pattern recognition communities have recently witnessed a surge of feature-based methods in object recognition and image retrieval applications. These methods allow representing images as collections of “visual words” and treat them using text search approaches following the “bag of features” paradigm. In this paper, we explore analogous approaches in the 3D world applied to the problem of non-rigid shape retrieval in large databases. Using multiscale diffusion heat kernels as “geometric words”, we construct compact and informative shape descriptors by means of the “bag of features” approach. We also show that considering pairs of “geometric words” (“geometric expressions”) allows creating spatially-sensitive bags of features with better discriminative power. Finally, adopting metric learning approaches, we show that shapes can be efficiently represented as binary codes. Our approach achieves state-of-the-art results on the SHREC 2010 large-scale shape retrieval benchmark.

Categories and Subject Descriptors:

■

## 1. INTRODUCTION

The availability of large public-domain databases of 3D models such as Google 3D Warehouse has created the demand for shape search and retrieval algorithms capable of finding similar shapes in the same way a search engine responds to text queries. However, while text search methods are sufficiently developed to be ubiquitously used e.g. in a Web application, the search and retrieval of 3D shapes remains a challenging problem. Shape retrieval based on text metadata (annotations and tags added by humans) is often not capable of providing the same experience as a text search engine [Min et al. 2004].

*Content-based* shape retrieval using the shape itself as a query and based on the comparison of the geometric and topological properties of shapes is complicated by the fact that many 3D objects manifest rich variability, and shape retrieval must often be *invariant* under different classes of transformations. A particularly challenging setting, which we address

in this paper, is the case of *non-rigid* or *deformable* shapes, which includes a wide range of shape transformations such as bending and articulated motion.

An analogous problem in the image domain is *image retrieval*: the problem of finding images depicting similar scenes or objects. Similar to three-dimensional shapes, images may manifest significant variability (Figure 1), and the aim of a successful retrieval approach is to be insensitive to such changes, while maintaining high discriminative power. Significant advances have been made in designing efficient image retrieval techniques (see an overview in [Veltkamp and Hagedoorn 2001]), but the majority of 2D retrieval methods do not immediately generalize to 3D shape retrieval [Tangelder and Veltkamp 2008].

Recently, feature-based methods have gained popularity in the computer vision and pattern recognition communities with the introduction of scale invariant feature transform (SIFT) [Lowe 2004] and similar algorithms [Matas et al. 2004; Bay et al. 2006]. The ability of these methods to demonstrate sufficiently good performance in many settings, including object recognition and image retrieval and the public availability of the code made SIFT-like approaches a commodity and a *de facto* standard in a variety of image analysis tasks.

One of the strengths of feature-based approaches in image retrieval is that they allow one to think of an image as a collection of primitive elements (visual “words”), and use the well-developed methods from text search such as the “bag of features” paradigm. One of the best implementations of these ideas is *Video Google*, a web application for object search in large collections of images and videos developed in Oxford university by Zisserman and collaborators [Sivic and Zisserman 2003; Chum et al. 2007], borrowing its name through an analogy with the famous text search engine. Video Google makes use of feature detectors and descriptors to represent an image as a collection of visual words indexed in a “visual vocabulary.” Each image is compactly encoded into a vector of frequencies of occurrences of visual words, a representation referred to as a “bag of features.” Images containing similar

visual information tend to have similar bags of features, and thus comparing bags of features allows retrieving similar images.

Zisserman *et al.* showed that employing weighting schemes for bags of features that take into consideration the average occurrence of visual words in the whole database allows for very accurate retrieval [Sivic and Zisserman 2003; Chum et al. 2007]. Since the comparison of bags of features usually boils down to finding weighted correlation between vectors, such a method is suitable for indexing and searching very large (Internet-scale) databases of images.

In a follow-up work, Grauman *et al.* [Jain et al. 2008] showed that an optimal weighting of bags of features can be learned from examples of similar and dissimilar images using metric learning approaches [Torralba et al. 2008]. Shakhnarovich [2005] proposed the *similarity-sensitive hashing*, which regards metric learning as a boosted classification problem. This method appeared very efficient in learning invariance to transformations in the context of video retrieval [Bronstein et al. 2010c]. In [Bronstein et al. 2010e], an extension of this approach to the multi-modal setting was presented. In [Strecha et al. 2010], similarity-sensitive hashing algorithms were applied to local SIFT descriptors to improve the performance of feature matching in wide-baseline stereo reconstruction problems.

Behmo *et al.* [2008] showed that one of the disadvantages of the bag of features approaches is that they lose information about the spatial location of features in the image, and proposed the commute graph representation, which partially preserves the spatial information. An extension of this work based on the construction of vocabularies of spatial relations between features was proposed in [Bronstein and Bronstein 2010a].

The success of feature-based methods in the computer vision community is the main inspiration for the current paper, where we present a similar paradigm for 3D shapes.

## 1.1 Related works in the shape analysis community

Shape retrieval is an established research area with many approaches and methods. For a detailed recent review, we refer the reader to [Tangelder and Veltkamp 2008]. In rigid shape retrieval, global descriptors based on volume and area [Zhang and Chen 2001], wavelets [Paquet et al. 2000], statistical moments [Kazhdan et al. 2003; Novotni and Klein 2003; Tal et al. 2001], self-similarity (symmetry) [Kazhdan et al. 2004], and distance distributions [Osada et al. 2002] were used. Methods reducing the 3D shape retrieval to image retrieval use 2D views [Funkhouser et al. 2003; Chen et al. 2003], slices [Jiantao et al. 2004], silhouette and contour descriptors [Napoléon et al. 2007]. Graph-based methods based on skeletons [Sundar et al. 2003] and Reeb graphs [Hilaga et al. 2001; Biasotti et al. 2003; Tung and Schmitt 2005] are capable of dealing with deformations e.g. matching articulated

shapes. Lipman and Funkhouser [2009] proposed the Möbius voting scheme for sparse shape matching.

Isometric shape deformations were first explicitly addressed by Elad and Kimmel [2001; 2003]. The authors used multi-dimensional scaling (MDS) [Borg and Groenen 1997; Bronstein et al. 2006] to construct a representation of the intrinsic geometry of shapes (captured as a matrix of inter-point geodesic distances and referred to as *canonical forms*) in a low-dimensional Euclidean space. A moment-based shape descriptor [Tal et al. 2001] was then applied to obtain a shape signature. The method of Elad and Kimmel was extended in [Mémoli and Sapiro 2005; Bronstein et al. 2006b; 2006a] to compare shapes as metric spaces using the Gromov-Hausdorff distance [Gromov 1981], which tries to find the minimum-distortion correspondence between two shapes. Numerically, the Gromov-Hausdorff distance computation can be carried out using a method similar to MDS [Bronstein et al. 2006b] or graph labeling [Torresani et al. 2008; Wang et al. 2010]. In follow-up works, extensions of the Gromov-Hausdorff distance used diffusion geometries [Bronstein et al. 2010d], different distortion criteria [Mémoli 2007; Mémoli 2009], local photometric [Thorstensen and Keriven 2009] and geometric [Dubrovina and Kimmel 2010; Wang et al. 2010] data, and third-order [Zeng et al. 2010] distortion terms. While allowing for very accurate and theoretically isometry-invariant shape comparison, the main drawback of the Gromov-Hausdorff framework is that it is based on computationally-expensive optimization. As a result, such methods are mostly suitable for one-to-one or one-to-few shape retrieval cases, even though there have been recent attempts to overcome this difficulty using Gromov-Hausdorff stable invariants [Chazal et al. 2009], which can be computed efficiently in practice.

Reuter *et al.* [2005; 2009] proposed using Laplace-Beltrami spectra (eigenvalues) as isometry-invariant shape descriptors. The authors noted that such descriptors are invariant to isospectral shape transformations, a family theoretically larger than isometries. Rustamov [2007] used an isometry-invariant shape embedding similar to the eigenmaps proposed in [Bérard et al. 1994; Belkin and Niyogi 2003; Lafon 2004] to create shape representations in the Euclidean space similar in spirit to canonical forms. He used histograms of Euclidean distances (using the approach of Osada *et al.* [Osada et al. 2002]) in the embedding space to compare the shapes. Shape similarity based on the comparison of histograms of diffusion distances were used as by Mahmoudi and Sapiro [2009]. In [Bronstein et al. 2010d; Bronstein and Bronstein 2009; 2010b], an intimate relation between this method and the methods of Rustamov [Rustamov 2007] was shown, and [Bronstein and Bronstein 2010b] showed a more general framework of which [Mahmoudi and Sapiro 2009; Rustamov 2007] are particular cases.

Mémoli [2009] formulated a proper metric between shapes based on a spectral variant of the Gromov-Wasserstein distance, and showed that shape similarity measures of Reuter *et al.* [Reuter et al. 2005], Rustamov [2007], and Sun *et al.*

[2009] can be viewed as a hierarchy of lower bounds of this metric.

Local feature-based methods are less common in the shape analysis community than in computer vision, as there is nothing equivalent to a robust feature descriptor like SIFT to be universally adopted. We see a few possible reasons. First, one of the important properties of SIFT is its discriminativity combined with robustness to different image transformations. Compared to images, shapes are usually known to be poorer in features, and thus descriptors are less informative. Secondly, unlike images where invariance is usually limited to affine transformations, the degree of invariance required for shapes is usually much larger and includes non-rigid and topology-changing deformations.

Feature-based methods have been explored by Pauly *et al.* [2003] for non-photorealistic rendering. The authors detect multiscale robust contour features (somewhat analogous to *edges* in images). A similar approach was proposed in [Kolomenkin et al. 2009]. In [Mitra et al. 2010], local features were used to detect shape self-similarity and grid-like structures. Local moments [Clarenz et al. 2004] and volume descriptors [Gelfand et al. 2005] have been also proposed for rigid shape retrieval and correspondence. Shilane and Funkhauser [2006] use a learnable feature detector and descriptor maximizing the likelihood of correct retrieval, based on spherical harmonics and surface area relative to the center of mass. Mitra *et al.* [2006] proposed a patch-based shape descriptor and showed its application to shape retrieval and comparison. They also showed an efficient hashing scheme of such descriptors. An approach similar to the one presented in this paper was presented in [Li et al. 2006] for rigid shapes.

Relatively few feature-based methods are invariant to isometric deformations by construction. Raviv *et al.* [Raviv et al. 2007] and Bronstein *et al.* [Bronstein et al. 2009] used histograms of local geodesic distances to construct an isometry-invariant local descriptor. In [Bronstein et al. 2009], statistical weighting scheme similar to [Sivic and Zisserman 2003; Chum et al. 2007] was also used to define point-wise significance measure in the problem of partial shape matching. Ben-Chen *et al.* [Ben-Chen et al. 2008] use conformal factors as isometry-invariant local descriptors. Zaharescu *et al.* [Zaharescu et al. 2009] proposed a SIFT-like descriptor applied to functions defined on manifolds. Sun *et al.* [Sun et al. 2009] introduced deformation-invariant descriptors based on diffusion heat kernels (referred to as *heat kernel signatures* or HKS). More recently, a scale-invariant modification of these descriptors (SI-HKS) were constructed in [Bronstein and Kokkinos 2010] using local scale normalization, and a volumetric heat kernel signature (vHKS) was proposed in [Raviv et al. 2010]. These former two approaches (HKS and SI-HKS) are adopted here for their discriminativity and efficient computability.

## 1.2 Main contribution

In this paper, we bring the spirit of feature-based computer vision approaches to the problem of non-rigid shape search

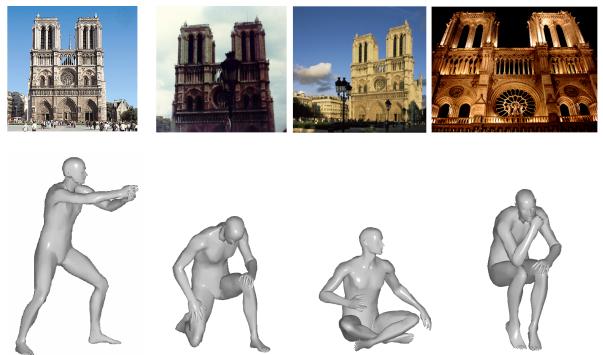


Fig. 1. Examples of invariant image (top) and shape (bottom) retrieval. Shown on the left is a query, and on the right a few examples of desired correct matches retrieved from a large database. Transformations shown in image retrieval are viewpoint variation, different illumination, background variation, occlusion and partially missing data; in shape retrieval, different non-rigid shape deformations are shown.

and retrieval. By analogy to Zisserman’s group works, we call our method *Shape Google*. The present paper is an extended version of [Ovsjanikov et al. 2009], where the approach was first introduced. While working on this paper, we discovered that the use of bags of features for shape representation has been independently developed by Toldo *et al.* [2009].

In this paper, we first show a feature detector and descriptor based on heat kernels of the Laplace-Beltrami operator, inspired by Sun *et al.* [2009]. Descriptors are used to construct a vocabulary of geometric words, distributions over which serve as a representation of a shape. This representation is invariant to isometric deformations, robust under a wide class of perturbations, and allows one to compare shapes undergoing different deformations. Secondly, we show that taking into consideration the spatial relations between features in an approach similar to commute graphs [Behmo et al. 2008] allows improving the retrieval performance. Finally, adopting metric learning techniques widely used in the computer vision community [Jain et al. 2008], we show how to represent shapes as compact binary codes that can be efficiently indexed and compared using the Hamming distance.

Figure 2 depicts a flow diagram of the presented approach. The shape is represented as a collection of local feature descriptors (either dense or computed at a set of stable points following an optional stage of feature detection). The descriptors are then represented by “geometric words” from a “geometric vocabulary” using vector quantization, which produces a shape representation as a bag of geometric words or pairs of words (expressions). Finally, similarity sensitive hashing is applied to the bags of features. We emphasize that the presented approach is generic, and different descriptor and detectors can be used depending on the application demands.

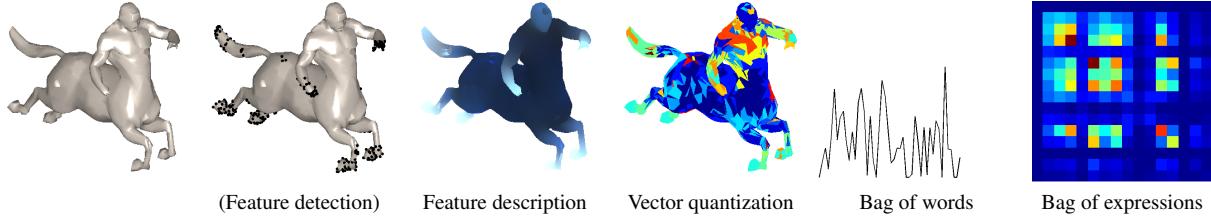


Fig. 2. Flow of the ShapeGoogle algorithm.

The rest of this paper is organized as follows. In Section 2, we start with a brief overview of feature-based approaches in computer vision, focusing on methods employed in Video Google. In Section 3, we formulate a similar approach for shapes. We show how to detect and describe local geometric features. In Section 4, we describe the construction of bags of geometric words. In Section 5, we explore metric learning techniques for representing shapes as short binary codes using Hamming embedding. Section 6 shows experimental results, and Section 7 concludes the paper.

## 2. BACKGROUND: FEATURE-BASED METHODS IN COMPUTER VISION

The construction of a feature-based representation of an image typically consists of two stages, *feature detection* and *feature description*, often combined into a single algorithm. The main goal of a feature detector is to find stable points or regions in an image that carry significant information on the one hand and can be repeatedly found in transformed versions of the image on the other. Since there is no clear definition of what is a feature, different approaches can be employed. For example, in the SIFT method, feature points are located by looking for local maxima of the discrete image Laplacian (approximated as a difference of Gaussians) at different scales. SIFT uses linear scale-space in order to search for feature points that appear at multiple resolutions of the image, which also makes the method scale-invariant [Lowe 2004]. Maximum stable extremal region (MSER) algorithm finds level sets in the image which exhibit the smallest variation of area when traversing the level-set graph [Matas et al. 2004; Kimmel et al. 2010]. Finally, it is possible to select all the points in the image or a regular subsampling thereof as the set of features (in the latter case, the detector is usually referred to as *dense* [Tola et al. 2008]).

The next stage is feature description. A feature descriptor uses a representation of local image information in the neighborhood of each feature point. For example, SIFT assigns a 128-dimensional descriptor vector constructed as local histograms of image gradient orientations around the point. The descriptor itself is oriented by the dominant gradient direction, which makes it rotation-invariant [Lowe 2004]. A similar approach, Speeded Up Robust Feature (SURF) transform [Bay et al. 2006], uses a 64-dimensional descriptor, computed effi-

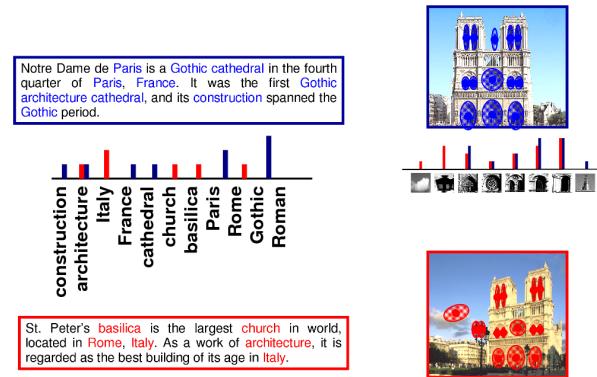


Fig. 3. Representation of text (left) and images (right) using the bags of features paradigm.

ciently using integral images. At this stage, the image can be compactly represented by specifying the spatial coordinates of the detected feature points together with the corresponding descriptors, which can be presented as vectors. This information allows, for example, finding correspondence between images by matching their descriptors [Lowe 2004].

In order to reduce the representation size, a vocabulary is constructed by performing vector quantization in the descriptor space. Descriptors can be replaced by indices in the vocabulary representing visual “words”. Typical vocabulary size can vary from a few thousand [Sivic and Zisserman 2003] up to one million words [Chum et al. 2007]. Aggregating all the indices into a histogram by counting the frequency of appearance of each visual word, the *bag of features* (sometimes also called *bag of visual terms* or *bag of visterns*) is constructed (Figure 3).

After the feature detection and description stages, two images can be compared this way by comparing their bags of features. This way, the image similarity problem is reduced to the problem of comparing vectors of feature frequencies. Typically, weighted correlation or weighted Euclidean distance is used to measure similarity of bags of features. The weights can be chosen in such a way that features frequent in the query shape (high *term frequency*) and infrequent in the entire database (low *document frequency*) are assigned a

large weight. The weight is expressed as the ratio of the term frequency and the document frequency (referred to as *term-frequency inverse document frequency* or *tf-idf* in search engine literature). It was shown in [Sivic and Zisserman 2003; Chum et al. 2007] that this type of weighted distance is superior to simple, non-weighted, approach. In [Jain et al. 2008], it was shown that an optimal weighted distance between bags of features on a given database can be learned by supervised learning from examples of similar and dissimilar images.

A few recent papers tried to extend bags of features by taking into consideration spatial information about the features. Marszalek and Schmid [2006] used spatial weighting to reduce the influence of background clutter (a similar approach was proposed in [Leibe et al. 2004]). Grauman and Darrell [2005] proposed comparing distributions of local features using *earth mover's distance* (EMD) [Rubner et al. 2000], which incorporates spatial distances. In [Lazebnik et al. 2006], the spatial structure of features was captured using a multiscale bag of features construction. The representation proposed in [Amores et al. 2007] used spatial relations between parts.

Behmo et al. [2008] proposed a generalization of bags of features that takes into consideration the spatial relations between the features in the image. In this approach, following the stage of feature detection and description, a *feature graph* is constructed. The connectivity of the graph is determined by the spatial distance and visual similarity of the features (spatially and visually-close features are connected). Next, the graph is collapsed by grouping together features whose descriptors are quantized to the same index in the visual vocabulary. The connections in the collapsed graph represent the *commute time* between the graph nodes. This graph can be considered an extension of a bag of features, where there is additional information about the relations between the visual words. In [Bronstein and Bronstein 2010a], images were described as histograms of *pairs* of features and the spatial relations between them (referred to as *visual expressions*), using a visual vocabulary and a vocabulary of spatial relations.

### 3. LOCAL FEATURES IN SHAPES

Trying to adapt feature-based approached to 3D shapes, one needs to have the following in mind. First, the type of invariance in non-rigid shapes is different from one required in images. Typically, feature detectors and descriptors in images are made invariant to affine transformations, which account for different possible views of an object captured in the image. In case of non-rigid shapes, the richness of transformations is much larger, including changes in pose, bending, and connectivity changes. Since many natural shape deformations such as articulated motion can be approximated by isometries, basing the shape descriptors on intrinsic properties of the shape will make it invariant to such deformations. Second, shapes are typically less rich in features than images, making it harder to detect a large number of stable and repeatable feature points. This poses a challenging tradeoff in feature detection between the number of features required to describe a shape on one

hand and the number of features that are repeatable on the other, and motivates our choice to avoid feature detection at all and use dense descriptors instead. Third, unlike images which in the vast majority of applications appear as matrices of pixels, shapes may be often represented as triangular meshes, point clouds, voxels, level sets, etc. Therefore, it is desirable to have local features computable across multiple representations. Finally, since shapes usually do not have a global system of coordinates, the construction of spatial relations between features is a challenging problem.

There exists a plethora of local shape detection and description algorithms, and below we overview some of them. The reader is referred to the review paper [Bustos et al. 2005] and the recent benchmarks [Bronstein et al. 2010a; Bronstein et al. 2010b] for additional details.

#### 3.1 Feature detectors

**Harris 3D.** An effective feature detection method, called the Harris operator, first proposed in images [Harris and Stephens 1988] was extended to 3D shapes by Glomb [2009] and Sipiran and Bustos [2010]. This method is based on measuring variability of the shape in a local neighborhood of the point, by fitting a function to the neighborhood, and identifying feature points as points where the derivatives of this function are high [Bronstein et al. 2010a].

**Mesh DOG.** Several methods for feature detection have been inspired by the the *difference of Gaussians* (DOG), a classical feature detection approach used in computer vision. Zaharescu et al. [2009] introduce the *mesh DOG* approach by first applying Gaussian filtering to functions (e.g. mean or Gauss curvature) defined on the shape. This creates a representation of the function in scale space, and feature points are prominent maxima of the scale space across scales. Castellani et al. [2008] apply Gaussian filtering directly on the mesh geometry, and use a robust method inspired by [Itti et al. 1998] to detect feature points as points with greatest displacement in the normal direction.

**Heat kernel feature detectors.** Recently, Sun et al. [2009] and Gebal et al. [2009] introduced feature detection methods based on the heat kernel. These methods define a function on the shape, measuring the amount of heat remaining at a point  $x$  after large time  $t$  given a point source at  $x$  at time 0, and detect features as local maxima of this function. As these methods are intimately related to our work, we discuss in depth the properties of heat kernels in the following section.

#### 3.2 Feature descriptors

**Shape context.** Though originally proposed for 2D shapes and images [Belongie et al. 2002], *shape context* has also been generalized to 3D shapes. For a point  $x$  on a shape  $X$ , the shape context descriptor is computed as a log-polar histogram of the relative coordinates of the other points  $(x' - x)$  for all  $x' \in X$ . Such a descriptor is translation-invariant and can be made rotation-invariant. It is computable on any kind of shape representation, including point clouds, voxels, and tri-

angular meshes. It is also known to be insensitive to small occlusions and distortion, but in general is not deformation-invariant, which makes it disadvantageous in non-rigid shape analysis applications.

**Spin images.** Perhaps one of the best known classes of feature descriptors are *spin images* [Johnson and Hebert 1999; Andreetto et al. 2004; Assfalg et al. 2007], which describe the neighborhood of a point by fitting a tangent plane to the surface at the point, and accumulating information about the neighborhood into 2D images which can then be directly compared. Although these methods can be robust with respect to noise and changes in triangulation, they were originally developed for rigid shape comparison, and are thus very sensitive to non-rigid shape deformations.

**Mesh HOG.** Zaharescu et al. [2009] use the histogram of gradients of a function defined in a local neighborhood of a point, as a point descriptor (similar to the *histogram of gradients* (HOG) [Dalal and Triggs 2005] technique used in computer vision). Though Zaharescu et al. show insensitivity of their descriptor to non-rigid deformations, the fact that it is constructed based on k-ring neighborhoods makes it theoretically triangulation-dependent.

**Heat kernel signatures (HKS).** Recently, there has been increased interest in the use of *diffusion geometry* for shape recognition [Rustamov 2007; Ovsjanikov et al. 2008; Mateus et al. 2008; Mahmoudi and Sapiro 2009; Bronstein et al. 2010d; Raviv et al. 2010]. This type of geometry arises from the *heat equation*,

$$\left( \Delta_X + \frac{\partial}{\partial t} \right) u = 0, \quad (1)$$

which governs the conduction of heat  $u$  on the surface  $X$  (here,  $\Delta_X$  denotes the negative semi-definite *Laplace-Beltrami operator*, a generalization of the Laplacian to non-Euclidean domains). The fundamental solution  $K_t(x, z)$  of the heat equation, also called the *heat kernel*, is the solution of (1) with a point heat source at  $x$  (see Figure 4). Probabilistically, the heat kernel can also be interpreted as the transition density function of a Brownian motion (continuous analog of a random walk) [Hsu 2002; Coifman and Lafon 2006; Lafon 2004].

Sun et al. [Sun et al. 2009] proposed using the diagonal of the heat kernel as a local descriptor, referred to as the *heat kernel signatures* (HKS). For each point  $x$  on the shape, its heat kernel signature is an  $n$ -dimensional descriptor vector of the form

$$p(x) = c(x)(K_{t_1}(x, x), \dots, K_{t_n}(x, x)), \quad (2)$$

where  $c(x)$  is chosen in such a way that  $\|p(x)\|_2 = 1$ .

The HKS descriptor has many advantages, which make it a favorable choice for shape retrieval applications. First, the heat kernel is intrinsic (i.e., expressible solely in terms of the Riemannian structure of  $X$ ), and thus invariant under isometric deformations of  $X$ . This makes HKS deformation-invariant (Figure 5, bottom row left). Second, such a descriptor captures information about the neighborhood of a point  $x$  on the shape *at a scale* defined by  $t$ . It captures differential information in

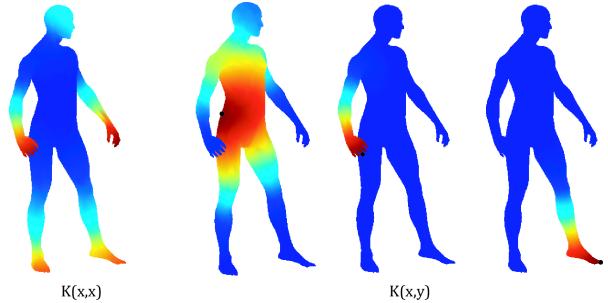


Fig. 4. Values of  $K_t(x, x)$  mapped on the shape (left) and values of  $K_t(x, y)$  for three different choices of  $y$  (marked with black dots in three rightmost figures). The value  $t = 1024$  is used. Hotter colors represent smaller values.

a small neighborhood of  $x$  for small  $t$ , and global information about the shape for large values of  $t$ . Thus, the  $n$ -dimensional feature descriptor vector  $p(x)$  can be seen as analogous to the multi-scale feature descriptors used in the computer vision community. Third, for small scales  $t$ , the HKS descriptor takes into account local information, which makes topological noise have only local effect (Figure 5, bottom row right). Fourth, Sun et al. prove that if the Laplace-Beltrami operator of a shape is non-degenerate (does not contain repeated eigenvalues), then any continuous map that preserves the HKS at every point must be an isometry. This latter property led Sun et al. to call the HKS *provably informative*. Finally, as will be shown next, the computation of the HKS descriptor relies on the computation of the first eigenfunctions and eigenvalues of the Laplace-Beltrami operator, which can be done efficiently and across different shape representations. This makes HKS applicable to different geometric data, though we focus in this paper on shapes represented as triangular meshes.

**Scale-invariant heat kernel signatures (SI-HKS).** A disadvantage of the HKS is its dependence on the global scale of the shape. If  $X$  is globally scaled by  $\beta$ , the corresponding HKS is  $\beta^{-2} K_{\beta^{-2}t}(x, x)$ . It is possible in theory to perform global normalization of the shape (e.g. normalizing the area or Laplace-Beltrami eigenvalues), but such a normalization is impossible if the shape has e.g. missing parts. As an alternative, a local normalization was proposed in [Bronstein and Kokkinos 2010] based on the properties of the Fourier transform. By using a logarithmic scale-space  $t = \alpha^\tau$ , global scaling results in HKS amplitude scaling and shift by  $2 \log_\alpha \beta$  in the scale-space. This effect is undone by the following sequence of transformations,

$$\begin{aligned} p_{dif}(x) &= (\log K_{\alpha^{\tau_2}}(x, x) - \log K_{\alpha^{\tau_1}}(x, x), \dots, \\ &\quad \log K_{\alpha^{\tau_m}}(x, x) - \log K_{\alpha^{\tau_{m-1}}}(x, x)), \\ \hat{p}(x) &= |(\mathcal{F}p_{dif}(x))(\omega_1, \dots, \omega_n)|, \end{aligned} \quad (3)$$

where  $\mathcal{F}$  is the discrete Fourier transform, and  $\omega_1, \dots, \omega_n$  denotes a set of frequencies at which the transformed vector is sampled. Taking differences of logarithms removes the scaling

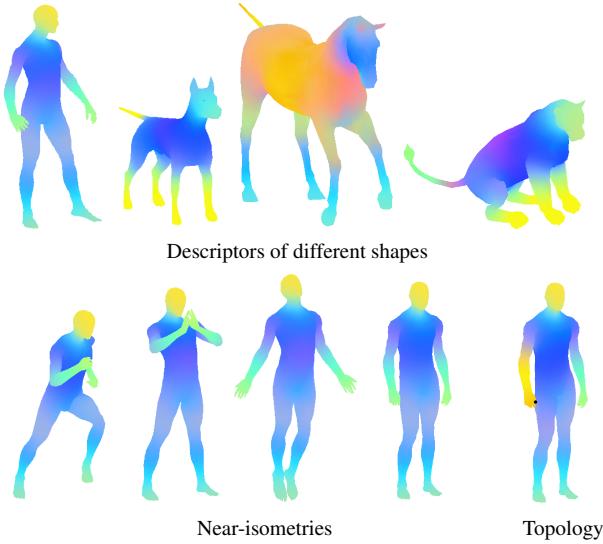


Fig. 5. An RGB visualization of the first three components of the HKS descriptor of different shapes (top row) and transformations of the same shape (bottom row). Four leftmost shapes in the bottom row are approximately isometric, and the descriptor appears to be invariant to these deformations. The rightmost shape in the bottom row has different topology (hand and leg are glued at point marked with black dot). Note that though the descriptor changes in the place of the topological change, the discrepancy is localized.

constant, and the Fourier transform converts the scale-space shift into a complex phase, which is removed by taking the absolute value. Typically, a large  $m$  is used to make the representation insensitive to large scaling factors and edge effects. Such a descriptor was dubbed *scale-invariant HKS* (SI-HKS) [Bronstein and Kokkinos 2010].

### 3.3 Numerical computation of HKS

For compact manifolds, the Laplace-Beltrami operator has a discrete eigendecomposition of the form

$$-\Delta_X \phi_l = \lambda_l \phi_l, \quad (4)$$

where  $\lambda_0 = 0 \geq \lambda_1 \geq \lambda_2 \dots$  are eigenvalues and  $\phi_0 = \text{const}$ ,  $\phi_1, \dots$  are the corresponding eigenfunctions. The heat kernel can be written in the following form, [Jones et al. 2008]

$$K_t(x, x') = \sum_{l=0}^{\infty} e^{-\lambda_l t} \phi_l(x) \phi_l(x'). \quad (5)$$

Since the coefficients decay exponentially as  $\lambda_i$  increase, for large values of  $t$  we can approximate the HKS as

$$K_t(x, x) \approx \sum_{l=0}^k e^{-\lambda_l t} \phi_l(x)^2. \quad (6)$$

Thus, in practice, the computation of the HKS descriptor boils down to computing the first largest eigenvectors and eigenvalues of the Laplace-Beltrami operator. For  $t \approx 0$ , this computation of HKS is numerically unstable, an issue that has been recently addressed in [Vaxman et al. 2010] using a multi-resolution approach.

**Discrete eigendecomposition problem.** In the discrete setting, having the shape represented by a finite sampling  $\hat{X} = \{x_1, \dots, x_N\}$ , several variants of the Laplace-Beltrami operator can be expressed in the following generic form,

$$(\Delta_{\hat{X}} f)_i = \frac{1}{a_i} \sum_j w_{ij} (f_i - f_j), \quad (7)$$

where  $f : \hat{X} = f(x_i)$  is a scalar function defined on  $\hat{X}$ ,  $w_{ij}$  are weights, and  $a_i$  are normalization coefficients. In matrix notation, Equation (7) can be written as

$$\Delta_{\hat{X}} f = A^{-1} W f, \quad (8)$$

where  $A = \text{diag}(a_i)$  and  $W = \text{diag}(\sum_{l \neq i} w_{il}) - (w_{ii})$ , allowing one to find the discrete eigenfunctions and eigenvalues by solving the *generalized eigendecomposition* [Lévy 2006]

$$W\Phi = A\Phi\Lambda, \quad (9)$$

where  $\Lambda$  is the  $(k+1) \times (k+1)$  diagonal matrix of eigenvalues and  $\Phi$  is an  $N \times (k+1)$  matrix of the corresponding eigenvectors, such that  $\Phi_{il} \approx \phi_l(x_i)$ .

**Laplace-Beltrami operator discretization.** Different discretizations of the Laplace-Beltrami lead to different choice of  $A$  and  $W$  [Zhang 2004; Floater and Hormann 2005; Bobenko and Springborn 2007]. For triangular meshes, a popular choice adopted in this paper is the *cotangent weight* scheme [Pinkall and Polthier 1993] and its variants [Meyer et al. 2003], in which  $w_{ij} = (\cot \alpha_{ij} + \cot \beta_{ij})/2$  for  $j$  in the 1-ring neighborhood of vertex  $i$  and zero otherwise, where  $\alpha_{ij}$  and  $\beta_{ij}$  are the two angles opposite to the edge between vertices  $i$  and  $j$  in the two triangles sharing the edge. It can be shown [Wardetzky et al. 2008] that this discretization preserves many important properties of the continuous Laplace-Beltrami operator, such as positive semi-definiteness, symmetry, and locality. For shapes represented as point clouds, the discretization of [Belkin et al. 2009] can be used.

**Finite elements.** Direct computation of the eigenfunction without explicit discretization of the Laplace-Beltrami operator is possible using the *finite elements method* (FEM). By the Green formula, the Laplace-Beltrami eigenvalue problem  $\Delta_X \phi = \lambda \phi$  can be expressed in the *weak form* as

$$\langle \Delta_X \phi, \alpha \rangle_{L_2(X)} = \lambda \langle \phi, \alpha \rangle_{L_2(X)} \quad (10)$$

for any smooth  $\alpha$ , where  $\langle f, g \rangle_{L_2(X)} = \int_X f(x)g(x)d\mu(x)$  and  $\mu(x)$  is the standard area measure on  $X$ . Given a finite basis  $\{\alpha_1, \dots, \alpha_q\}$  spanning a subspace of  $L_2(X)$ , the solution  $\phi$  can be expanded as  $\phi(x) \approx u_1\alpha_1(x) + \dots + u_q\alpha_q(x)$ . Substituting this expansion into (10) results in a system of equations

$$\sum_{j=1}^q u_j \langle \Delta_X \alpha_j, \alpha_r \rangle_{L_2(X)} = \lambda \sum_{j=1}^q u_j \langle \alpha_j, \alpha_r \rangle_{L_2(X)},$$

for  $r = 1, \dots, q$ , which, in turn, is posed as a generalized eigenvalue problem

$$Au = \lambda Bu. \quad (11)$$

(here  $A$  and  $B$  are  $q \times q$  matrices with elements  $a_{rj} = \langle \Delta_X \alpha_j, \alpha_r \rangle_{L_2(X)}$  and  $b_{rj} = \langle \alpha_j, \alpha_r \rangle_{L_2(X)}$ ). Solution of (11) gives eigenvalues  $\lambda$  and eigenfunctions  $\phi = u_1\alpha_1 + \dots + u_q\alpha_q$  of  $\Delta_X$ .

As the basis, linear, quadratic, or cubic polynomials defined on the mesh can be used. The FEM approach is quite general, and in particular, the cotangent scheme can be derived as its instance by using piecewise linear hat functions centered on the vertices and supported within the 1-ring neighborhood. Since the inner products in FEM are computed on the surface, the method can be less sensitive to the shape discretization than the direct approach based on the discretization of the Laplace-Beltrami operator. This is confirmed by numerical studies performed by Reuter *et al.* who showed the advantage in accuracy of higher-order FEM schemes at the expense of computational and storage complexity [Reuter et al. 2005]. In [Bronstein et al. 2010b], linear FEM method produced comparable results in the discretization of HKS compared to cotangent weights (note that linear FEM method defines the same Laplace-Beltrami operator as the cotangent weight Laplacian, with the sole difference that we use the lumped mass matrix to get a diagonal matrix  $A$  for the latter).

## 4. BAGS OF FEATURES

Given local descriptor computed at a set of stable feature points (or alternatively, a dense descriptor), similarly to feature-based approaches in computer vision, our next step is to quantize the descriptor space in order to obtain a compact representation in a vocabulary of “geometric words”. A vocabulary  $\mathcal{P} = \{p_1, \dots, p_V\}$  of size  $V$  is a set of representative vectors in the descriptor space, obtained by means of unsupervised learning (vector quantization through  $k$ -means).

Given a vocabulary  $\mathcal{P}$ , for each point  $x \in X$  with the descriptor  $p(x)$ , we define the *feature distribution*  $\theta(x) = (\theta_1(x), \dots, \theta_V(x))^T$ , a  $V \times 1$  vector whose elements are:

$$\theta_i(x) = c(x) e^{-\frac{\|p(x) - p_i\|_2^2}{2\sigma^2}}, \quad (12)$$

and the constant  $c(x)$  is selected in such a way that  $\|\theta(x)\|_1 = 1$ .  $\theta_i(x)$  can be interpreted as the probability of the point  $x$  to be associated with the descriptor  $p_i$  from the vocabulary  $\mathcal{P}$ .

Equation (12) is a “soft” version of vector quantization. “Hard” vector quantization is obtained as a particular case of (12) by choosing  $\sigma \approx 0$ , in which case  $\theta_i(x) = 1$  (where  $i$  is the index of the vocabulary element  $p_i$  closest to  $p$  in the descriptor space) and zero otherwise.

Integrating the feature distribution over the entire shape  $X$  yields a  $V \times 1$  vector

$$f(X) = \int_X \theta(x) d\mu(x), \quad (13)$$

which we refer to as a *bag of features* (or *BoF* for short). Using this representation, we can define a distance between two shapes  $X$  and  $Y$  as a distance between bags of features in  $\mathbb{R}^V$ ,

$$d_{BoF}(X, Y) = \|f(X) - f(Y)\|. \quad (14)$$

An example of bags of features using a vocabulary of size 64 is shown in Figure 7 (top).

### 4.1 Spatially-sensitive bags of features

The disadvantage of bags of features is the fact that they consider only the distribution of the words and lose the relations between them. Resorting again to a text search example, in a document about “matrix decomposition” the words “matrix” and “decomposition” are frequent. Yet, a document about the movie *Matrix* and a document about decomposition of organic matter will also contain these words, which will result in a similar word statistics and, consequently, similar bags of features. In the most pathological case, a random permutation of words in a text will produce identical bags of words. In order to overcome this problem, text search engines commonly use vocabularies consisting not only of single words but also of combinations of words or *expressions*. The combination of words “matrix decomposition” will be thus frequently found in a document about the algebraic notion, but unlikely in a document about the *Matrix* movie (Figure 6).<sup>1</sup>

In case of shapes, the phenomenon may be even more pronounced, as shapes, being poorer in features, tend to have many similar geometric words. The analogy of expressions in shapes would be sets of spatially-close geometric words. Instead of looking at the frequency of individual geometric words, we look at the frequency of word pairs, thus accounting not only for the frequency but also for the spatial relations between features. For this purpose, we define the following generalization of a bag of features, referred to as a *spatially-sensitive bags of features* (SS-BoF):

$$F(X) = \int_{X \times X} \theta(x)\theta^T(y)K_t(x, y)d\mu(x)d\mu(y). \quad (15)$$

The resulting representation  $F$  is a  $V \times V$  matrix, representing the frequency of appearance of nearby geometric words

<sup>1</sup>For this reason, web search engines return different results when the search string is written with quotation marks (“matrix decomposition”) and without (matrix decomposition).

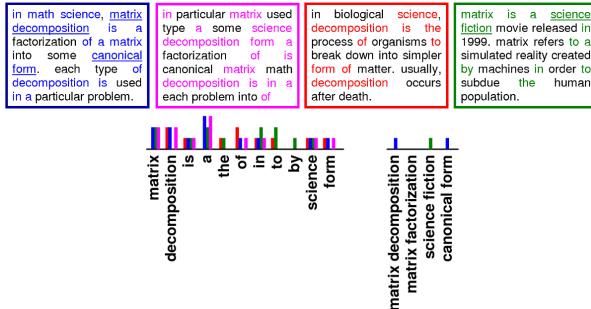


Fig. 6. Example visualizing the importance of expressions. Shown left to right: text about matrix decomposition in math; same text with randomly shuffled words; text on biological process of decomposition; text about the movie *Matrix*. All the texts have similar or even identical distributions of words, shown as the left histogram. Yet, adding a vocabulary with combinations of words (expressions) allows one to distinguish between the texts (right histogram).

or “geometric expressions”  $i, j$ . It can be considered as a bag of features in a vocabulary of size  $V^2$  consisting of pairs of words (see Figure 7, bottom). When hard quantization is used (the vectors  $\theta$  are binary), the element  $F_{ij}$  would be large if the instances of the words  $i$  and  $j$  are spatially close to each other on  $X$  (the diffusion distances between words  $i$  and  $j$  is small, or alternatively, the heat kernel  $K$  is large).  $f_{ij}$  can also be interpreted as the proximity of the words  $i, j$  in the descriptor space and on the surface  $X$ , which is similar to the spirit of [Behmo et al. 2008; Bronstein and Bronstein 2010a].

We define a distance between two shapes  $X$  and  $Y$  as a distance between  $F(X)$  and  $F(Y)$ ,

$$d_{\text{SS-BoF}}(X, Y) = \|F(X) - F(Y)\|. \quad (16)$$

## 4.2 Statistical weighting

An important question is how to compare the bag-of-features representations of two shapes. While deferring the question of constructing an optimal metric to the next section, we note that not all geometric words are equally important for the purpose of shape retrieval. In text retrieval, it is common to assign different weights to words according to their statistical importance. Frequent words in a document are likely to be informative; on the other hand, in order to be discriminative, they should be rare in the corpus of documents to which the document belongs. Down-weighting common words like prepositions and articles increases the performance of text search engines. Similar approaches have been successfully used for object retrieval in video [Sivic and Zisserman 2003], as well as three-dimensional shape comparison [Shilane and Funkhauser 2006; Bronstein et al. 2009].

Here, we use statistical weighting of bags of geometric features. In the case when hard quantization is used, a bag of

features represents the occurrence frequency of different geometric words in the shape, referred to as *term frequency*. Assume we have a shape database of size  $D$ , containing the shapes  $X_1, \dots, X_D$ . The *inverse document frequency* of geometric word  $i$  is defined as the logarithm of the inverse fraction of the shapes in the database in which this word appears,

$$w_i = \log \left( \frac{D}{\sum_{j=1}^D \delta(f_i(X_j) > 0)} \right), \quad (17)$$

where  $\delta$  is an indicator function, and  $f_i(X_j)$  counts the number of occurrences of word  $i$  in shape  $j$ . The smaller is  $w_i$ , the more common the term  $i$  is in the database, and so it is less likely to be able to discriminate between shapes [Chum et al. 2007]. This information can be taken into consideration when comparing bags of features, by down-weighting common words. This leads to a weighted  $L_1$  distance:

$$d_{\text{BoFw}}(X, Y) = \sum_{i=1}^V w_i |f_i(X) - f_i(Y)|, \quad (18)$$

referred to as *tf-idf weighting*.

Statistical weighting can be applied in the same way to SS-BoFs. In the computation of inverse document frequency, instead of geometric words, geometric expressions (pairs of words) are used. This way, common expressions are down-weighted.

## 5. SHAPES AS BINARY CODES

Assume that we are given a class of transformations  $\mathcal{T}$  (such as bending), invariance to which is desired in the shape retrieval problem. Theoretically, it is possible to construct the local descriptors to be invariant under  $\mathcal{T}$ , making the resulting bag of features representation, up to quantization errors, invariant as well. However, only simple geometric transformation can be modeled explicitly, and thus the descriptor would rarely be truly invariant to real shape transformations.

**Metric learning.** As an alternative, we propose *learning* the invariance from examples. Assume that we are given a set of bags of features (or spatially-sensitive bags of features)  $\mathcal{F}$  describing different shapes. We denote by  $\mathcal{P} = \{(f, f \circ \tau) : f \in \mathcal{F}, \tau \in \mathcal{T}\}$  the set of *positive pairs* (bags of features of identical shapes, differing up to some transformation), and by  $\mathcal{N} = \{(f, f') : f \neq f' \in \mathcal{F}\}$  the set of *negative pairs* (bags of features of different shapes). Negative pairs are modeled by taking different shapes which are known to be distinct. For positive pairs, it is usually possible to simulate representative transformations from class  $\mathcal{T}$ . Our goal is to find a metric between bags of features that is as small as possible on the set of positives and as large as possible on the set of negatives.

**Similarity-sensitive hashing (SSH).** Shakhnarovich [Shakhnarovich 2005] considered metrics parameterized as

$$d_{A,b}(x, x') = d_{\mathbb{H}}(\text{sign}(Ax + b), \text{sign}(Ax' + b)), \quad (19)$$

where  $d_{\mathbb{H}}(y, y') = \frac{s}{2} - \frac{1}{2} \sum_{i=1}^s \text{sign}(y_i y'_i)$  is the *Hamming metric* in the  $s$ -dimensional *Hamming space*  $\mathbb{H}^s = \{-1, +1\}^s$

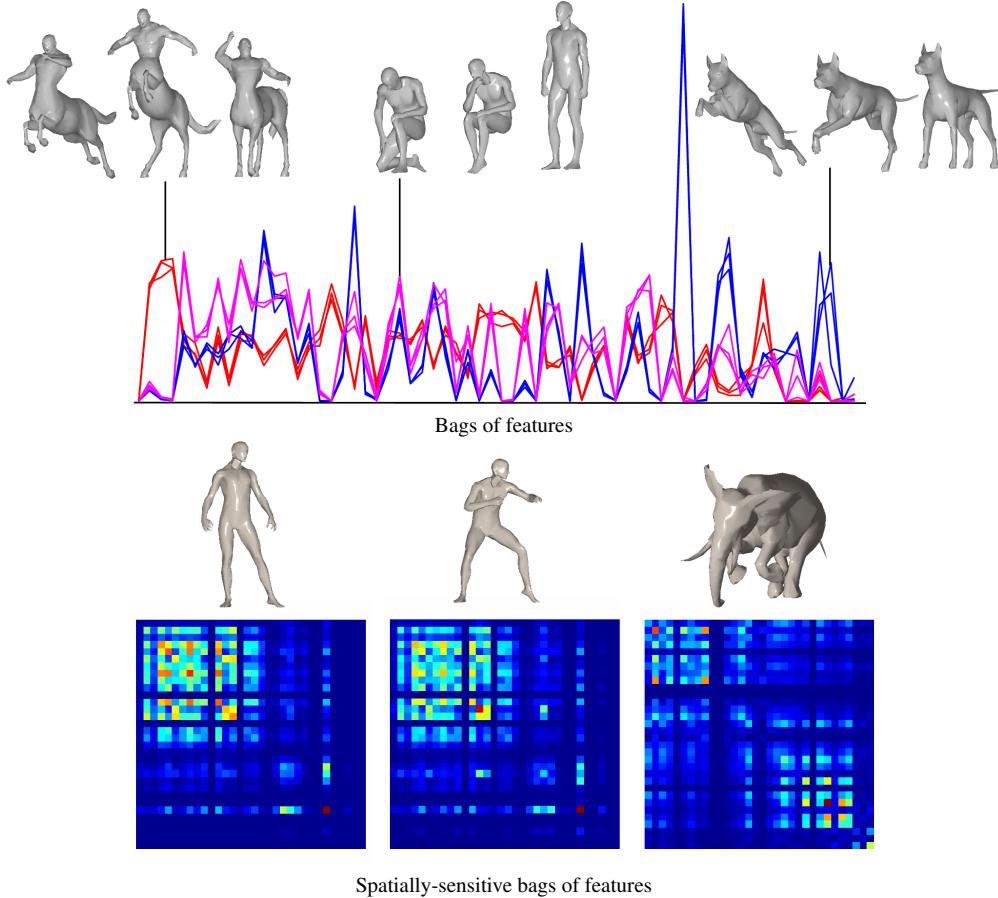


Fig. 7. Top row: examples of bags of features computed for different deformations of centaur (red), dog (blue), and human (magenta). Note the similarity of bags of features of different transformations and dissimilarity of bags of features of different shapes. Also note the overlap between the centaur and human bags of features due to partial similarity of these shapes. Bottom row: examples of spatially-sensitive bags of features computed for different deformations of human (left and center) and elephant (right).

of binary sequences of length  $s$ .  $A$  and  $b$  are an  $s \times V$  matrix and an  $s \times 1$  vector, respectively, parameterizing the metric. Our goal is to find  $A$  and  $b$  such that  $d_{A,b}$  reflects the desired similarity of pairs of bags of features  $f, f'$  in the training set.

Ideally, we would like to achieve  $d_{A,b}(f, f') \leq d_0$  for  $(f, f') \in \mathcal{P}$ , and  $d_{A,b}(f, f') > d_0$  for  $(f, f') \in \mathcal{N}$ , where  $d_0$  is some threshold. In practice, this is rarely achievable as the distributions of  $d_{A,b}$  on  $\mathcal{P}$  and  $\mathcal{N}$  have cross-talks responsible for false positives ( $d_{A,b} \leq d_0$  on  $\mathcal{N}$ ) and false negatives

( $d_{A,b} > d_0$  on  $\mathcal{P}$ ). Thus, optimal  $A, b$  should minimize

$$\min_{A, b} \quad \frac{1}{|\mathcal{P}|} \sum_{(f, f') \in \mathcal{P}} \left\{ e^{\text{sign}(d_{A,b}(f, f') - d_0)} \right\} \\ + \frac{1}{|\mathcal{N}|} \sum_{(f, f') \in \mathcal{N}} \left\{ e^{\text{sign}(d_0 - d_{A,b}(f, f'))} \right\}. \quad (20)$$

In [Shakhnarovich 2005], the learning of optimal parameters  $A, b$  was posed as a boosted binary classification problem, where  $d_{A,b}$  acts as a strong binary classifier, and each dimension of the linear projection  $\text{sign}(A_i x + b_i)$  is a weak classifier. This way, the AdaBoost algorithm can be used to progressively construct  $A$  and  $b$ , which is a greedy solution of (20). At the  $i$ -th iteration, the  $i$ -th row of the matrix  $A$  and the  $i$ -th element

---

**Algorithm 1:** Metric learning between bags of features using similarity sensitive hashing (SSH).

---

**Input:**  $P$  pairs of bags of features  $(\mathbf{f}_p, \mathbf{f}'_p)$  labeled by  $s_p = s(\mathbf{f}_p, \mathbf{f}'_p) \in \{\pm 1\}$ .

**Output:** Parameters  $A, b$  of the optimal projection.

- 1 Initialize weights  $w_p^1 = 1/P$ .
- 2 **for**  $i = 1, \dots, d$  **do**
- 3   Set the  $i$ th row of  $A$  and  $b$  by solving the optimization problem
- $$(\mathbf{A}_i, b_i) = \min_{\mathbf{A}_i, b_i} \sum_{p=1}^P w_p^i s_p (2 - \text{sign}(\mathbf{A}_i \mathbf{f}_p + b_i)) (2 - \text{sign}(\mathbf{A}_i \mathbf{f}'_p + b_i)).$$
- 4   Update weights  
         $w_p^{i+1} = w_p^i e^{-s_p \text{sign}(\mathbf{A}_i \mathbf{f}_p + b_i) \text{sign}(\mathbf{A}_i \mathbf{f}'_p + b_i)}$  and normalize by sum.

---

of the vector  $\mathbf{b}$  are found minimizing a weighted version of (20). Weights of false positive and false negative pairs are increased, and weights of true positive and true negative pairs are decreased, using the standard AdaBoost reweighting scheme [Freund and Schapire 1995], as summarized in Algorithm 1.

**Projection selection.** Finding an optimal projection  $\mathbf{A}_i$  at Step 3 of Algorithm 1 is difficult because of the sign function non-linearity. In [Shakhnarovich 2005], a random projection was used at each step. In [Bronstein et al. 2010c], it was shown empirically that a better way is to select a projection by using the minimizer of the exponential loss of a simpler problem,

$$\mathbf{A}_i = \underset{\|\mathbf{A}\|=1}{\text{argmax}} \frac{\mathbf{A}^T C_N \mathbf{A}}{\mathbf{A}^T C_P \mathbf{A}}, \quad (21)$$

where  $C_P$  and  $C_N$  are the covariance matrices of the positive and negative pairs, respectively. It can be shown that  $\mathbf{A}_i$  maximizing (21) is the largest generalized eigenvector of  $C_N^{\frac{1}{2}} \mathbf{A}_i = \lambda_{\max} C_P^{\frac{1}{2}} \mathbf{A}_i$ . This approach is similar in its spirit to *linear discriminative analysis* (LDA) and was used in [Bronstein et al. 2010c; Bronstein et al. 2010e]. Since the minimizers of (20) and (21) do not coincide exactly, in our implementation, we select a subspace spanned by the largest ten eigenvectors, out of which the direction as well as the threshold parameter  $\mathbf{b}$  minimizing the exponential loss are selected. Such an approach appears to be more efficient (in terms of bits required for the hashing to discriminate between dissimilar data) than [Shakhnarovich 2005].

**Advantages.** There are a few advantages to invariant metric learning in our problem. First, unlike the tf-idf weighting scheme, which constructs a metric between bags of features based on an empirical evidence, the metric  $d_{A,b}$  is constructed to achieve the best discriminativity and invariance on the training set. The Hamming metric in the embedding space can be therefore thought of as an *optimal* metric on the training set.

---

**Algorithm 2:** ShapeGoogle algorithm for shape retrieval.

---

**Input:** Query shape  $X$ ; geometric vocabulary  $\mathcal{P} = \{p_1, \dots, p_V\}$ , database of shapes  $\{X_1, \dots, X_D\}$ .

**Output:** Shapes from  $\{X_1, \dots, X_D\}$  most similar to  $X$ .

- 1 **if** Feature detection **then**
- 2   | Compute a set  $X'$  of stable feature points on  $X$ .
- 3 **else**
- 4   |  $X' = X$ .
- 5 Compute local feature descriptor  $p(x)$  for all  $x \in X'$ .
- 6 Quantize the local feature descriptor  $p(x)$  in the vocabulary  $\mathcal{P}$ , obtaining for each  $x$  a distribution  $\theta(x) \in \mathbb{R}^V$ .
- 7 **if** Geometric expressions **then**
- 8   | Compute a spatially-sensitive  $V \times V$  bag of geometric words
- $$F(X) = \int_{X \times X} \theta(x) \theta^T(y) K_t(x, y) d\mu(x) d\mu(y),$$
  
and parse it into a  $V^2 \times 1$  vector  $f(X)$ .
- 9 **else**
- 10   | Compute a  $V \times 1$  bag of geometric words
- $$f(X) = \int_X \theta(x) d\mu(x)$$
- 11 Embed the bag of words/expressions into the  $s$ -dimensional Hamming space  
 $y(X) = \text{sign}(Af(X) + b)$  using pre-computed parameters  $A, b$ .
- 12 Compare the bitcode  $y(X)$  to bitcodes  $y(X_1), \dots, y(X_D)$  of the shapes in the database using Hamming metric and find the closest shape.

---

If the training set is sufficiently representative, such a metric generalizes well. Secondly, the projection itself has an effect of dimensionality reduction, and results in a very compact representation of bags of features as *bitcodes*, which can be efficiently stored and manipulated in standard databases. Thirdly, modern CPU architectures allow very efficient computation of Hamming distances using bit counting and SIMD instructions. Since each of the bits can be computed independently, shape similarity computation in a shape retrieval system can be further parallelized on multiple CPUs using either shared or distributed memories. Due to the compactness of the bitcode representation, search can be performed in memory. Fourth, invariance to different kinds of transformations can be learned. Finally, since our approach is a meta-algorithm, it works with any local descriptor, and in particular, can be applied to descriptors most successfully coping with particular types of transformations and invariance.

We summarize our shape retrieval approach in Algorithm 2.

ACM Transactions on Graphics, Vol. , No. , Article , Publication date: .

## 6. RESULTS

### 6.1 Experimental setup

**Dataset.** In order to assess our method, we used the SHREC 2010 robust large-scale shape retrieval benchmark, simulating a retrieval scenario, in which the queries include multiple modifications and transformations of the same shape [Bronstein et al. 2010b]. The dataset used in this benchmark was aggregated from three public domain collections: TOSCA shapes [Bronstein et al. 2008], Robert Sumner’s collection of shapes [Sumner and Popović 2004], and Princeton shape repository [Shilane et al. 2004]. The shapes were represented as triangular meshes with the number of vertices ranging approximately between 300 and 30,000. The dataset consisted of two parts: 715 shapes from 13 shape classes with simulated transformation (55 per shape) used as queries and the remaining 456 shapes, as shown in Figure 8. The total dataset size was 1184.<sup>2</sup>

**Queries.** The query set consisted of 13 shapes taken from the dataset (null shapes), with simulated transformations applied to them. For each null shape, transformations were split into 11 classes shown in Figure 9: *isometry* (non-rigid almost inelastic deformations), *topology* (welding of shape vertices resulting in different triangulation), *micro holes* and *big holes*, *global* and *local scaling*, additive Gaussian *noise*, *shot noise*, *partial occlusion* (resulting in multiple disconnected components), *down sampling* (less than 20% of original points), and *mixed* transformations. In each class, the transformation appeared in five different versions numbered 1–5. In all shape categories except scale and isometry, the version number corresponded to the transformation strength levels: the higher the number, the stronger the transformation (e.g., in noise transformation, the noise variance was proportional to the strength number). For scale transformations, the levels 1–5 corresponded to scaling by the factor of 0.5, 0.875, 1.25, 1.625, and 2. For the isometry class, the numbers did not reflect transformation strength. The total number of transformations per shape was 55, and the total query set size was 715. Each query had one correct corresponding null shape in the dataset.

**Evaluation criteria.** Evaluation simulated matching of transformed shapes to a database containing untransformed (null) shapes. As the database, all 469 shapes with null transformations were used. Multiple query sets according to transformation class and strength were used. For transformation  $x$  and strength  $n$ , the query set contained all the shapes with transformation  $x$  and strength  $\leq n$ . In each transformation class, the query set size for strengths 1,2,..., 5 was 13, 26, 39, 52, and 65. In addition, query sets with all transformations broken down according to strength were used, containing 143, 286, 429, 572, and 715 shapes (referred to as *average* in the following).

<sup>2</sup>The datasets are available at [http://tosca.cs.technion.ac.il/book/shrec\\_robustness.html](http://tosca.cs.technion.ac.il/book/shrec_robustness.html)

Performance was evaluated using precision/recall characteristic. *Precision P(r)* is defined as the percentage of relevant shapes in the first  $r$  top-ranked retrieved shapes. In the present benchmark, a single relevant shape existed in the database for each query. *Mean average precision* (mAP), defined as

$$mAP = \sum_r P(r) \cdot rel(r),$$

(where *rel(r)* is the relevance of a given rank), was used as a single measure of performance. Intuitively, mAP is interpreted as the area below the precision-recall curve. Ideal performance retrieval performance results in first relevant match with mAP=100%. Retrieval performance results between similar-class positive shapes (males and females, centaur, horse, and human shapes) were ignored and did not participate in the statistics, since these shapes can be considered either positive or negative depending on the application.

### 6.2 ShapeGoogle

In the first experiment, we studied the performance of the proposed bag of features approach (without SSH). The shapes were represented as bags of geometric words or expressions based on HKS or SI-HKS local descriptors. The computation of the HKS descriptors was performed as follows. The cotangent weight scheme with Dirichlet boundary conditions was used to discretize  $\Delta_X$ . The heat kernel was approximated using  $k = 100$  largest eigenvalues and eigenvectors. The HKS was computed at  $n = 6$  scales, with  $t_1, \dots, t_6$  were chosen as 1024, 1351, 1783, 2353, 3104 and 4096 (these are setting identical to [Ovsjanikov et al. 2009]). For the computation of SI-HKS descriptors, we used the same heat kernel discretization; the values of  $\tau$  were from 1 to 25 with increments of 1/16 ( $m = 385$ ), and  $\alpha = 2$ . The first six discrete frequencies of the Fourier transform ( $n = 6$ ) were taken (these are settings identical to [Bronstein and Kokkinos 2010]). Approximate nearest neighbors approach [Arya et al. 1998] was used to perform clustering in the descriptor space and construct the vocabularies. The soft quantization parameter  $\sigma$  was set to twice the median size of the clusters in the geometric vocabulary. The  $L_1$  distance was used to compare between bags of features and spatially-sensitive bags of features.

**Complexity.** The code was implemented in MATLAB with some parts written in C with MEX interface. The experiments were run on a laptop with a 2GHz Intel Pentium Core2 Duo CPU and 3GB of RAM. Computational time for a bag of feature was in the range of 1-10 seconds, depending mostly on the number of vertices. Typical time of computing the discrete Laplacian matrix and finding its eigenvectors was a few seconds; computation of HKS descriptors took negligible time; vector quantization took less than 1 second; and computation of histogram was negligible. Comparison time of two bags of features using the  $L_1$  norm was also negligible. Pre-computation of the vocabulary was performed offline by clustering a set of approximately  $5 \times 10^5$  representative descriptors



Fig. 8. Null shapes used in the benchmark. Transformations of the first 13 shapes (top left) were used as queries in our experiments.

using [Arya et al. 1998] and took between 25 to 50 minutes depending on the number of clusters (vocabulary size).

**Choice of the descriptor.** Fixing the vocabulary size to 48, we compared the performance produced by different choices of the descriptor. Tables I and II show the performance (mAP in percent) across transformation classes and strengths of ShapeGoogle using bags of features based on HKS and SI-HKS local descriptors, respectively (for each descriptor, a separate vocabulary was computed). The use of SI-HKS shows significant performance boosting on global Scale transformations (from 27.42% to 98.21%), as well as improvement on Local scaling and Mixed transformations, and also overall performance. At the same time, SI-HKS shows slight degradation in the Topology transformations class, which is explained by the fact that we have to take larger scales of the heat kernel (up to  $2^{25}$ ) in order to provide sufficient support for the Fourier transform. This experiments visualizes how the choice of the descriptor results in invariance to different transformations.

Figure 10 shows a few examples of retrieved shapes, ordered by relevance, which is inversely proportional to the distance from the query shape. This figure visualizes how the performance of shape retrieval depends on the invariance of the local descriptor. Using HKS, all the matches for Scale and Mixed transformations (rows 1-2 and 4) are incorrect (middle column). On the other hand, using the SI-HKS, which is scale-invariant, dramatically improves the results in these transformation classes (right column).

**Sparse vs dense descriptor.** We compared the performance produced by using dense descriptor computed at each point of the shape as opposed to a sparse descriptor, computed at a set of points produced by a feature detection algorithm. We used two feature detection algorithms which produced the best results in the SHREC’10 feature detection benchmark [Bronstein et al. 2010a]. Both methods are based on the heat kernel signature [Sun et al. 2009]. In the first method (*HKS scale-space peaks*), feature points are detected as local peaks of the

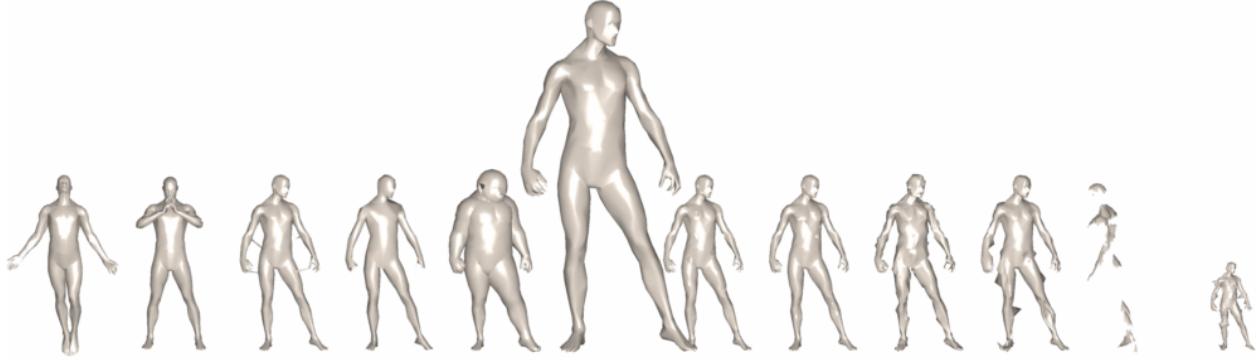


Fig. 9. Transformations of the human shape used as queries (shown in strength 5, left to right): null, isometry, topology, sampling, local scale, scale, holes, micro holes, noise, shot noise, partial, mixed.

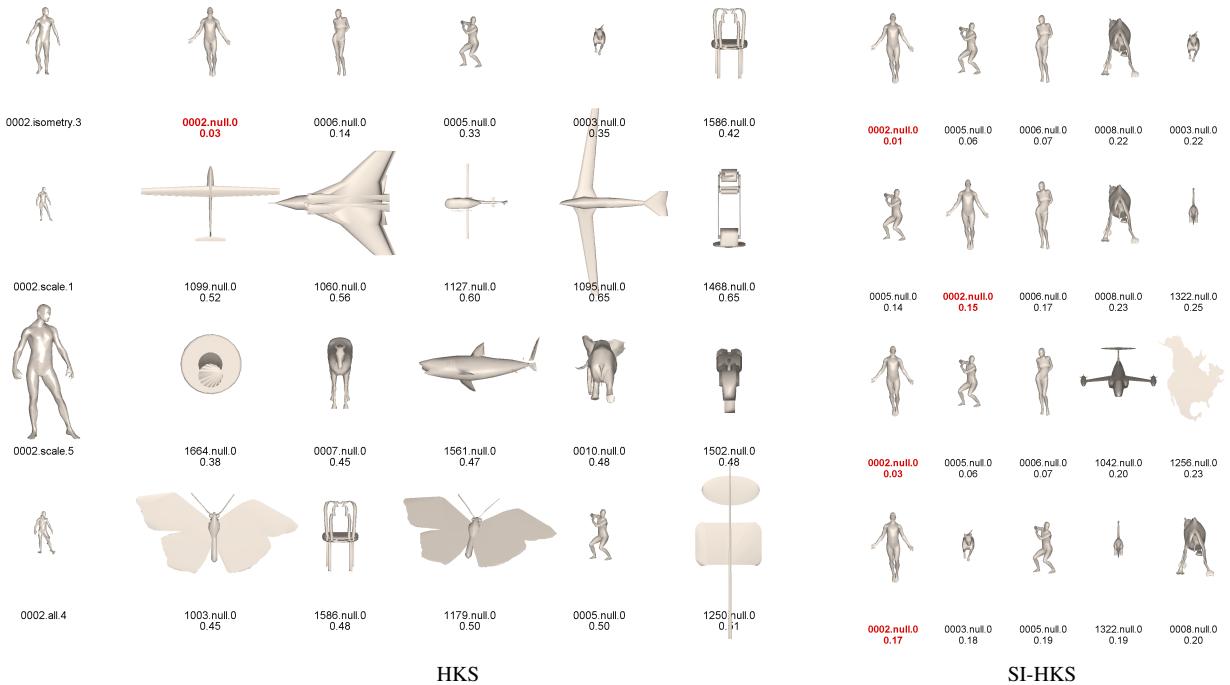


Fig. 10. Retrieval results with ShapeGoogle. Left: query shapes, middle: first five matches obtained with HKS descriptors, right: first five matches obtained with SI-HKS descriptors. Shape annotation follows the convention *shapeid.transformation.strength*; numbers below show distance from query. Only a single correct match exists in the database (marked in red), and ideally, it should be the first one.

HKS descriptor. Given the HKS function  $K_t(x, x)$  computed on a mesh for a large fixed value of  $t$ , a point  $x$  is declared as a feature point if  $K_t(x, x) > K_t(x', x')$  for all  $x'$  in a 2-ring neighborhood of  $x$ . The second method (*topological persistence*) follows the same basic procedure described above, but differs in the final step of selecting feature points. After defin-

ing the function  $K_t(x, x)$  for  $t = 0.1$ , persistent homology was used to filter out unstable feature points. For this, the 0-dimensional persistence diagram of this function [Edelsbrunner et al. 2000; Chazal et al. 2009] was computed. A point was declared a feature, if it is a local maximum of this function and furthermore, if the difference between the death and birth

Table I. Performance (mAP in %) of ShapeGoogle using bags of features of size 48 based on HKS local descriptor computed with cotangent weight discretization.

Transform.	Strength				
	1	$\leq 2$	$\leq 3$	$\leq 4$	$\leq 5$
Isometry	100.00	100.00	100.00	100.00	100.00
Topology	100.00	98.08	97.44	96.79	96.41
Holes	100.00	100.00	97.44	95.19	90.13
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	0.98	40.68	43.31	33.72	27.42
Local scale	100.00	100.00	98.72	89.38	80.22
Sampling	100.00	100.00	100.00	100.00	99.23
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	100.00	100.00	100.00	100.00	100.00
Partial	7.54	5.70	4.51	3.58	2.95
Mixed	53.13	55.86	47.77	37.54	30.34
Average	94.94	93.12	90.84	87.82	85.00

Table II. Performance (mAP in %) of ShapeGoogle using bags of features of size 48 based on SI-HKS local descriptor computed with cotangent weight discretization.

Transform.	Strength				
	1	$\leq 2$	$\leq 3$	$\leq 4$	$\leq 5$
Isometry	100.00	100.00	100.00	100.00	100.00
Topology	96.15	96.15	94.87	93.27	92.69
Holes	100.00	100.00	100.00	94.71	89.97
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	91.03	95.51	97.01	97.76	98.21
Local scale	100.00	100.00	97.44	89.38	82.08
Sampling	100.00	100.00	100.00	100.00	97.69
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	100.00	100.00	100.00	100.00	100.00
Partial	17.43	10.31	9.57	8.06	6.61
Mixed	56.47	57.44	63.59	67.47	65.07
Average	97.05	95.16	94.03	92.54	90.79

times of the corresponding connected component is above a threshold  $\alpha$  (see [Edelsbrunner et al. 2000] for details). A uniform  $\alpha = 0.1$  was used for all shapes, which was chosen by manually examining the persistence diagram of one of the null shapes. At the detected points, the HKS descriptor was computed. The bags of features were created using the same vocabulary of size 48.

Tables III and IV show the results obtained using both methods. We conclude that the overall performance of the dense descriptor (Table I) is superior.

**Vocabulary size.** To study the influence of the vocabulary size, we used the HKS local descriptor, changing the vocabulary size from 16 to 64 geometric words. Table V shows the obtained performance across transformation classes over all strengths ( $\leq 5$ ). The overall performance as well as performance almost in all classes improves with the increase of the

Table III. Performance (mAP in %) of ShapeGoogle using bags of features of size 48 based on feature points detected using HKS scale-space peaks and HKS local descriptor computed with cotangent weight discretization.

Transformation	Strength				
	1	$\leq 2$	$\leq 3$	$\leq 4$	$\leq 5$
Isometry	100.00	97.44	96.05	96.07	95.63
Topology	88.81	81.45	78.73	74.70	72.26
Holes	96.15	87.50	80.56	74.47	67.06
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	0.86	36.30	35.20	28.60	23.14
Local scale	100.00	94.23	90.26	77.69	67.93
Sampling	100.00	96.15	95.51	93.00	82.47
Noise	100.00	100.00	98.08	95.80	93.15
Shot noise	100.00	100.00	100.00	98.40	93.26
Partial	15.95	8.63	6.64	5.19	4.24
Mixed	31.68	31.13	24.55	19.15	15.65
Average	94.34	90.80	87.21	83.17	78.85

Table IV. Performance (mAP in %) of ShapeGoogle using bags of features of size 48 based on feature points detected using topological persistence and HKS local descriptor computed with cotangent weight discretization.

Transformation	Strength				
	1	$\leq 2$	$\leq 3$	$\leq 4$	$\leq 5$
Isometry	78.79	72.08	74.65	67.51	65.22
Topology	63.62	60.45	55.22	52.92	51.21
Holes	53.85	45.70	41.85	38.94	36.68
Micro holes	79.74	79.74	75.07	73.63	72.80
Scale	0.85	19.28	26.35	23.11	19.72
Local scale	53.88	50.63	51.57	50.31	47.47
Sampling	72.19	70.59	69.09	63.69	57.17
Noise	81.73	74.68	65.90	60.79	58.15
Shot noise	76.50	67.74	63.23	58.91	53.31
Partial	3.04	1.71	1.51	1.21	1.03
Mixed	22.00	25.21	21.47	18.48	16.50
Average	89.09	81.67	75.94	70.50	65.92

vocabulary size, coming at the expense of the representation size (length of the bag of features vector).

**Geometric expressions.** To test the influence of geometric expressions, we used the same geometric vocabularies to compute the spatially-sensitive bags of features. The time scale of the heat kernel in (15) was fixed to  $t = 1024$ .

Table VI shows the performance of ShapeGoogle with  $48 \times 48$  spatially-sensitive bags of features based on HKS local descriptors. Comparing to the performance of bags of features computed with the same vocabulary (Table I), we see consistent improvement in almost all classes of transformations except global Scale. Especially dramatic improvement is observed with Partial transformation (from 2.95% to 10.28%), which makes us believe that the use of geometric expressions can be especially helpful in coping with large missing parts. On the other hand, Scale performance deteriorates (from

Table V. Performance (mAP in %) of ShapeGoogle using bags of features of varying size based on HKS local descriptor computed with cotangent weight discretization. Performance is measured over all strengths ( $\leq 5$ ).

Transform.	Vocabulary size				
	16	24	32	48	64
<i>Isometry</i>	100.00	100.00	100.00	100.00	100.00
<i>Topology</i>	88.07	94.28	94.36	96.41	97.95
<i>Holes</i>	76.35	79.40	83.79	90.13	92.31
<i>Micro holes</i>	100.00	100.00	100.00	100.00	100.00
<i>Scale</i>	22.86	25.78	26.55	27.42	29.71
<i>Local scale</i>	75.67	79.70	78.05	80.22	81.63
<i>Sampling</i>	96.28	97.69	99.23	99.23	99.23
<i>Noise</i>	100.00	100.00	100.00	100.00	100.00
<i>Shot noise</i>	100.00	100.00	100.00	100.00	100.00
<i>Partial</i>	1.82	2.16	2.36	2.95	3.08
<i>Mixed</i>	26.61	27.86	29.30	30.34	30.26
<b>Average</b>	82.85	83.91	84.28	85.00	85.41

Table VI. Performance (mAP in %) of ShapeGoogle using spatially-sensitive bags of features of size  $48 \times 48$  based on HKS local descriptor computed with cotangent weight discretization.

Transform.	Strength				
	1	$\leq 2$	$\leq 3$	$\leq 4$	$\leq 5$
<i>Isometry</i>	100.00	100.00	100.00	100.00	100.00
<i>Topology</i>	100.00	100.00	100.00	100.00	100.00
<i>Holes</i>	100.00	100.00	100.00	100.00	99.23
<i>Micro holes</i>	100.00	100.00	100.00	100.00	100.00
<i>Scale</i>	0.73	43.44	40.32	31.69	26.01
<i>Local scale</i>	100.00	100.00	98.72	93.88	85.06
<i>Sampling</i>	100.00	100.00	97.44	98.08	97.70
<i>Noise</i>	100.00	100.00	100.00	100.00	100.00
<i>Shot noise</i>	100.00	100.00	100.00	100.00	100.00
<i>Partial</i>	24.13	19.28	16.24	12.53	10.28
<i>Mixed</i>	75.64	78.85	59.89	45.55	36.64
<b>Average</b>	95.76	94.54	91.86	89.10	86.55

27.42% to 26.01%), which is explained by the fact that our definition of spatial distance relations (15) using a fixed time scale is not scale-invariant.

Table VII shows the performance of spatially-sensitive bags of features as function of vocabulary size (compare to Table V). All the aforementioned phenomena are consistently manifested with different vocabularies. Overall, the spatially-sensitive bags of features show better performance with the same vocabulary, coming at the expense of larger storage complexity.

### 6.3 Similarity-sensitive hashing

In the second experiment, we evaluated the contribution of similarity sensitive hashing to the performance of our shape retrieval approach. A separate set with a total of 624 shapes

ACM Transactions on Graphics, Vol. , No. , Article , Publication date: .

Table VII. Performance (mAP in %) of ShapeGoogle using spatially-sensitive bags of features of varying size based on HKS local descriptor computed with cotangent weight discretization. Performance is measured over all strengths ( $\leq 5$ ).

Transform.	Vocabulary size				
	16×16	24×24	32×32	48×48	64×64
<i>Isometry</i>	99.23	99.23	99.23	100.00	100.00
<i>Topology</i>	97.44	99.23	100.00	100.00	100.00
<i>Holes</i>	95.73	98.08	98.46	99.23	100.00
<i>Micro holes</i>	100.00	100.00	100.00	100.00	100.00
<i>Scale</i>	24.18	23.08	23.85	26.01	24.54
<i>Local scale</i>	82.68	84.95	84.86	85.06	86.24
<i>Sampling</i>	97.70	97.70	97.70	97.70	97.70
<i>Noise</i>	100.00	100.00	100.00	100.00	100.00
<i>Shot noise</i>	100.00	100.00	100.00	100.00	100.00
<i>Partial</i>	7.65	8.47	9.39	10.28	10.28
<i>Mixed</i>	31.48	35.29	36.08	36.64	36.73
<b>Average</b>	85.51	86.06	86.25	86.55	86.58

provided as part of the SHREC 2010 benchmark was used for training. The training set included representative transformations of different classes and strengths. Besides null shapes, the training set contained no shape instances from the test set. The positive and negative set size was  $|\mathcal{P}| = 10^4$  and  $|\mathcal{N}| = 10^5$ , respectively. Hashing of length varying from  $s = 16$  to 96 bit was applied to bags of features based on SI-HKS local descriptor quantized in a vocabulary of size 48.

**Complexity.** Training time on the above set was about 20 minutes and was performed offline. After having the optimal projection parameters ( $A$  and  $b$ ), the additional complexity of embedding the bag of feature vectors into the Hamming space was negligible.

Table IX shows the performance of ShapeGoogle+SSH as a function of the hash length ( $s$ ) across transformation classes. Even for a very short hash length (16 bit), the method outperforms bags of features without hashing (average mAP of 94.12% compared to 90.79%, see Table II). Thus, hashing has both metric learning and dimensionality reduction effects. The latter offers significant advantages in storage and search complexity (compare to  $\approx 1000$  bit codes used in [Mitra et al. 2006]). The performance of SSH increases with the hash length. Table VIII summarizes the performance of ShapeGoogle+SSH with  $s = 96$ .

### 6.4 Comparison to other methods

As a reference, in the third experiment we compared our method to three state-of-the-art methods for shape retrieval.

**Shape DNA.** Reuter *et al.* [2006] described shapes by the vector of their first Laplace-Beltrami eigenvalues, a representation referred to as *Shape DNA*. Due to intrinsic nature of the Laplace-Beltrami operator, its spectrum is deformation-invariant. In our experiments, we used linear FEM [Patané and Falcidieno 2010] to compute the first 50 eigenvalues and the

Table VIII. Performance (mAP in %) of ShapeGoogle using bags of features of size 48 based on SI-HKS local descriptor computed with cotangent weight discretization and 96 bit similarity sensitive hash.

Transform.	Strength				
	1	$\leq 2$	$\leq 3$	$\leq 4$	$\leq 5$
Isometry	100.00	100.00	100.00	100.00	100.00
Topology	100.00	100.00	100.00	100.00	100.00
Holes	100.00	100.00	100.00	100.00	100.00
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	100.00	100.00	100.00	100.00	100.00
Local scale	100.00	100.00	100.00	97.18	94.98
Sampling	100.00	100.00	100.00	100.00	99.23
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	100.00	100.00	100.00	100.00	100.00
Partial	96.15	88.81	86.52	86.80	78.86
Mixed	96.15	96.15	97.44	98.08	95.35
Average	99.84	99.48	99.30	99.10	98.27

Table IX. Performance (mAP in %) of ShapeGoogle+SSH using bags of features of size 48 based on SI-HKS local descriptor computed with cotangent weight discretization as function of the hash length ( $s$ ). Performance is measured over all strengths ( $\leq 5$ ).

Transform.	Hash length ( $s$ )				
	16	24	32	48	64
Isometry	98.08	100.00	100.00	100.00	100.00
Topology	98.46	100.00	100.00	100.00	100.00
Holes	96.89	97.69	99.23	99.23	98.46
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	98.65	98.46	100.00	100.00	99.23
Local scale	88.96	93.55	94.98	93.85	97.74
Sampling	92.05	96.21	97.31	97.44	100.00
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	99.23	100.00	100.00	100.00	100.00
Partial	37.67	40.73	48.72	57.00	62.93
Mixed	83.73	85.55	85.90	89.33	91.02
Average	94.12	95.18	95.94	96.53	97.22

$L_1$  distance to compare between the vectors, which were empirically found to produce the best performance.

**Clock matching bag of features.** Lian *et al.* [2010] presented the *clock matching bag of features* (CM-BOF) method, which was among the top performing algorithms in the SHREC'10 benchmark [Bronstein *et al.* 2010b]. In this method, following a PCA-based shape normalization, the shape is projected onto multiple directions, resulting in a set of 2D views. Each view is described as a word histogram obtained by the vector quantization of the view's local features. During shape matching, all possible poses are taken into account. The exact settings of this method in our experiments appear in [Bronstein *et al.* 2010b].

**Part-based bags of words.** Toldo *et al.* [2009] used an approach following the same philosophy of ShapeGoogle, representing the shape as a histogram of “geometric words” de-

Table X. Performance (mAP in %) of Shape DNA (Reuter *et al.* [2006]).

Transform.	Strength				
	1	$\leq 2$	$\leq 3$	$\leq 4$	$\leq 5$
Isometry	83.35	85.92	82.18	81.83	81.76
Topology	83.35	83.35	83.35	83.35	83.35
Holes	84.97	58.32	41.60	31.93	25.81
Micro holes	84.63	84.63	84.63	84.63	85.40
Scale	1.01	1.95	1.55	1.27	1.10
Local scale	88.48	73.13	62.78	51.45	42.48
Sampling	83.35	85.96	85.56	83.90	75.73
Noise	83.35	85.92	80.66	72.95	63.07
Shot noise	83.35	85.92	82.92	73.75	62.35
Partial	2.17	2.12	2.06	1.89	1.74
Mixed	3.37	9.01	7.82	6.25	5.23
Average	91.11	84.72	78.94	73.68	68.60

Table XI. Performance (mAP in %) of part-based BOF (Toldo *et al.* [2009]).

Transform.	Strength				
	1	$\leq 2$	$\leq 3$	$\leq 4$	$\leq 5$
Isometry	100.00	88.57	88.54	88.04	88.04
Topology	100.00	98.08	97.44	97.12	97.69
Holes	100.00	91.94	76.24	67.65	61.58
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	100.00	100.00	100.00	100.00	100.00
Local scale	100.00	97.12	94.40	85.97	77.61
Sampling	94.87	95.51	90.68	85.06	79.30
Noise	100.00	88.78	69.27	58.23	52.67
Shot noise	92.31	94.23	93.59	93.27	93.85
Partial	1.35	1.49	1.46	1.48	1.43
Mixed	39.13	40.30	38.20	36.59	33.55
Average	95.28	92.11	88.41	85.06	82.20

scribing object sub-parts. Spectral clustering combined with a region-growing approach is used to segment the object into parts; each part is then described using a rigid descriptor. A more detailed description and the exact settings of this method in our experiments appear in [Bronstein *et al.* 2010b].

Tables X–XII detail the performance of these methods on the same dataset. CM-BOF shows the highest performance among the three, but is outperformed by ShapeGoogle+SSH shown in Table VIII (which has mAP of 98.27% compared to 94.33% of CM-BOF). The difference in performance is especially pronounced on the Isometry and Partial transformation classes.

Figure 11 depict the receiver operation characteristic (ROC) curves of ShapeGoogle+SSH and the compared methods. The ROC curve represents the tradeoff between the percentage of similar shapes correctly identified as similar (*true positives rate* or TPR) and the percentage of dissimilar shapes wrongfully identified as similar (*false positive rate* or FPR) depending on a global threshold applied to the shape distance matrix. These curves show that our approach significantly outperform the other methods.

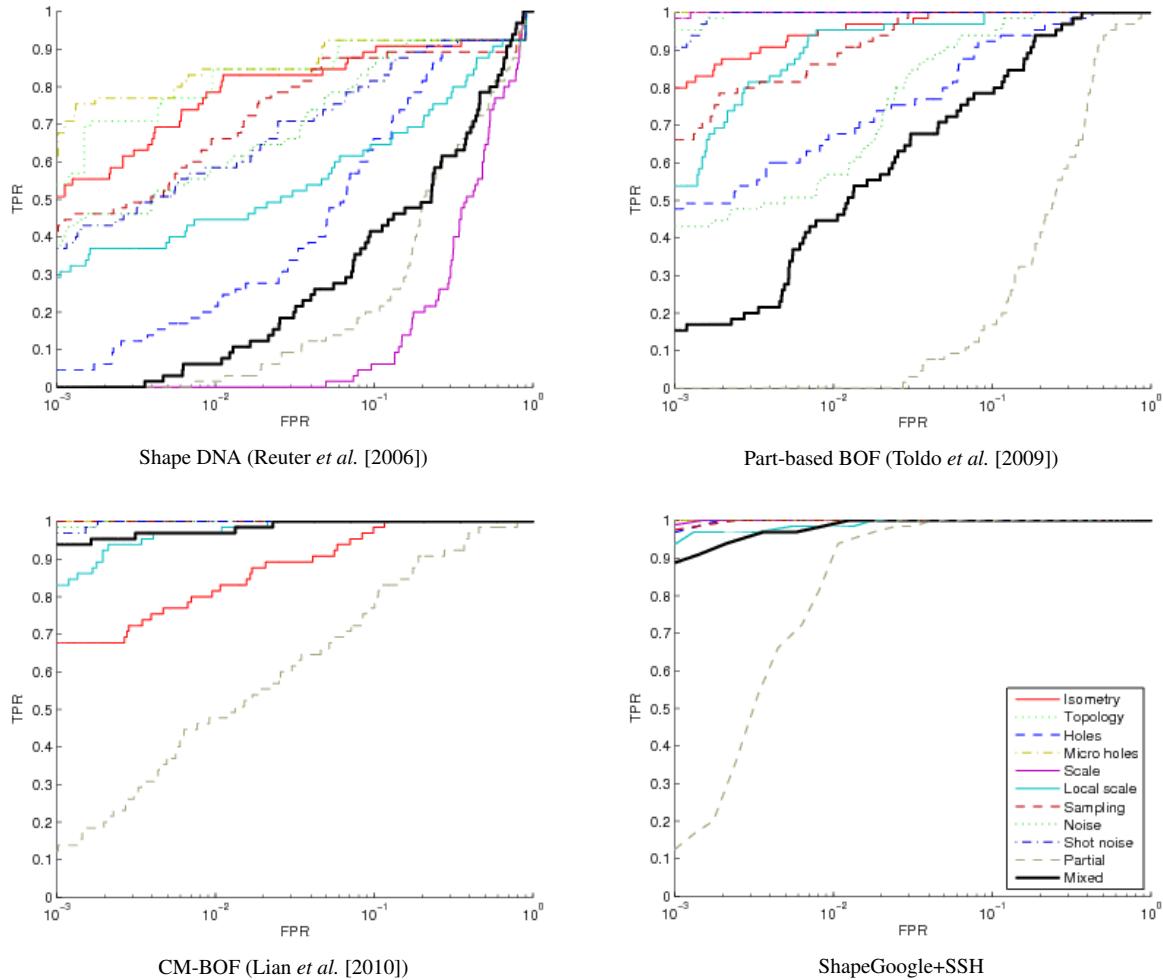


Fig. 11. ROC curves comparing the performance of different methods. TPR is the recognition rate (the higher the better).

Figure 12 visualizes a few examples of retrieved shapes, ordered by relevance, which is inversely proportional to the distance from the query shape.

## 7. CONCLUSIONS

We presented an approach to non-rigid shape retrieval similar in its spirit to text retrieval methods used in search engines. We drew analogies with feature-based image representations used in the computer vision community to construct shape descriptors that are invariant to a wide class of transformations on one hand and are discriminative on the other.

We have observed significant performance improvement in many classes of transformations when using bags of geometric

expressions rather than simple geometric words. Also, the use of similarity sensitive hashing showed dramatic improvement both in shape representation size and the quality of the shape metric. Our approach showed very high retrieval accuracy in a standard large-scale shape retrieval benchmark, exceeding the previous state-of-the-art approaches.

In the future works, we intend to explore in depth the analogy of our descriptors with scale-invariant feature transform (SIFT) in computer vision. Other applications possible using our approach are dense correspondence between non-rigid shapes. Extending metric learning methods to product vocabularies (like in our spatially-sensitive bags of features) may lead to further performance improvement. Finally, another promising direction is the use of multi-modal hashing [Bronstein et al.

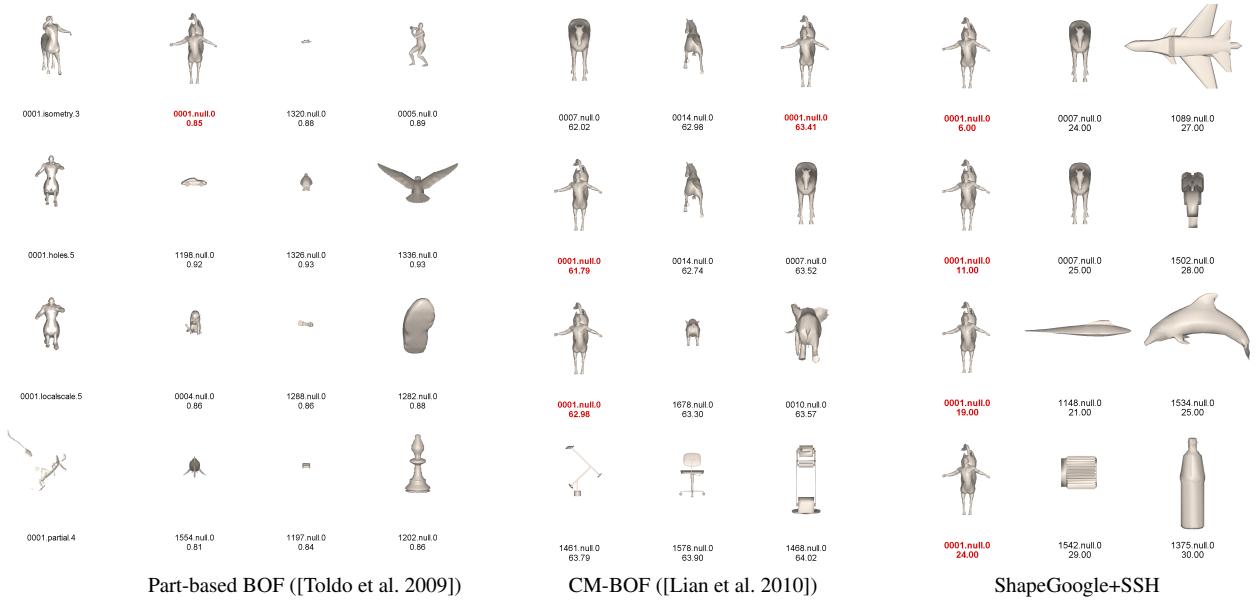


Fig. 12. Retrieval results using different methods. First column: query shapes, second column: first three matches obtained with the method of Toldo *et al.* [Toldo et al. 2009], third column: first three matches obtained with the method of Lian *et al.* [Lian et al. 2010], fourth column: first three matches obtained with ShapeGoogle+SSH. Shape annotation follows the convention *shapeid.transformation.strength*; numbers below show distance from query. Only a single correct match exists in the database (marked in red), and ideally, it should be the first one.

Table XII. Performance (mAP in %) of CM-BOF  
(Lian *et al.* [2010; 2010]).

Transform.	Strength				
	1	$\leq 2$	$\leq 3$	$\leq 4$	$\leq 5$
Isometry	100.00	86.67	79.24	77.46	72.58
Topology	100.00	100.00	100.00	100.00	100.00
Holes	100.00	100.00	100.00	100.00	100.00
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	100.00	100.00	100.00	100.00	100.00
Local scale	100.00	100.00	98.72	96.47	92.95
Sampling	100.00	100.00	100.00	100.00	100.00
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	100.00	100.00	100.00	100.00	98.46
Partial	54.22	47.45	46.28	40.57	35.49
Mixed	100.00	100.00	100.00	98.56	97.31
Average	99.03	97.73	96.71	95.66	94.33

2010e] to cope with the problem of cross-representation retrieval and different versions of vocabularies.

### Acknowledgment

We are grateful to Zhouhui Lian and Umberto Castellani for providing the performance of their algorithms on the SHREC'10 benchmark, and to Giuseppe Patané for providing the FEM computation of Laplace-Beltrami eigenvalues and eigenfunctions.

### REFERENCES

- AMORES, J., SEBE, N., AND RADEVA, P. 2007. Context-based object-class recognition and retrieval by generalized correlograms. *Trans. PAMI* 29, 10, 1818–1833.
- ANDRETTTO, M., BRUSCO, N., AND CORTELAZZO, G. M. 2004. Automatic 3D modeling of textured cultural heritage objects. *Trans. Image Processing* 13, 3, 335–369.
- ARYA, S., MOUNT, D. M., NETANYAHU, N. S., SILVERMAN, R., AND WU, A. Y. 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM* 45, 6, 891–923.
- ASSFALG, J., BERTINI, M., AND A. DEL BIMBO, P. P. 2007. Content-based retrieval of 3D objects using spin image signatures. *Trans. Multimedia* 9, 3, 589–599.
- BAY, H., TUYTELAARS, T., AND VAN GOOL, L. 2006. SURF: Speeded up robust features. In *Proc. ECCV*. 404–417.
- BEHMO, R., PARAGIOS, N., AND PRINET, V. 2008. Graph commute times for image representation. In *Proc. CVPR*.
- BELKIN, M. AND NIYOGI, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15, 6 (June), 1373–1396.
- BELKIN, M., SUN, J., AND WANG, Y. 2009. Constructing Laplace operator from point clouds in  $\mathbb{R}^d$ . In *Proc. Symp. Discrete Algorithms*. 1031–1040.

- BELONGIE, S., MALIK, J., AND PUZICHA, J. 2002. Shape matching and object recognition using shape contexts. *Trans. PAMI*, 509–522.
- BEN-CHEN, M., WEBER, O., AND GOTSMAN, C. 2008. Characterizing shape using conformal factors. In *Proc. 3DOR*.
- BÉRARD, P., BESSON, G., AND GALLOT, S. 1994. Embedding riemannian manifolds by their heat kernel. *Geometric and Functional Analysis* 4, 4, 373–398.
- BIASOTTI, S., MARINI, S., MORTARA, M., AND PATANÉ, G. 2003. An overview on properties and efficacy of topological skeletons in shape modeling. In *Proc. SMI*. 245–254.
- BOBENKO, A. I. AND SPRINGBORN, B. A. 2007. A discrete Laplace–Beltrami operator for simplicial surfaces. *Discrete and Computational Geometry* 38, 4, 740–756.
- BORG, I. AND GROENEN, P. 1997. *Modern multidimensional scaling - theory and applications*. Springer.
- BRONSTEIN, A. M. AND BRONSTEIN, M. M. 2010a. Affine-invariant spatially-sensitive vocabularies. In *Proc. ECCV*.
- BRONSTEIN, A. M., BRONSTEIN, M. M., BUSTOS, B., CASTELLANI, U., CRISANI, M., FALCIDIENO, B., GUIBAS, L. J., SIPIRAN, I., KOKKINOS, I., MURINO, V., OVSJANIKOV, M., PATANÉ, G., SPAGNUOLO, M., AND SUN, J. 2010a. SHREC 2010: robust feature detection and description benchmark. In *Proc. 3DOR*.
- BRONSTEIN, A. M., BRONSTEIN, M. M., CARMON, Y., AND KIMMEL, R. 2009. Partial similarity of shapes using a statistical significance measure. *Trans. Computer Vision and Applications* 1, 0, 105–114.
- BRONSTEIN, A. M., BRONSTEIN, M. M., CASTELLANI, U., FALCIDIENO, B., FUSIELLO, A., GODIL, A., GUIBAS, L. J., KOKKINOS, I., LIAN, Z., OVSJANIKOV, M., PATANÉ, G., SPAGNUOLO, M., AND TOLDO, R. 2010b. SHREC 2010: robust large-scale shape retrieval benchmark. In *Proc. 3DOR*.
- BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. 2006a. Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Scientific Computing* 28, 5, 1812–1836.
- BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. 2006b. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS* 103, 5, 1168–1172.
- BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. 2008. *Numerical geometry of non-rigid shapes*. Springer-Verlag New York Inc.
- BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. 2010c. The video genome. arXiv 1003.5320v1.
- BRONSTEIN, A. M., BRONSTEIN, M. M., KIMMEL, R., MAHMOUDI, M., AND SAPIRO, G. 2010d. A Gromov–Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *IJCV* 89, 2–3, 266–286.
- BRONSTEIN, M. M. AND BRONSTEIN, A. M. 2009. On a relation between shape recognition algorithms based on distributions of distances. Tech. Rep. CIS-2009-14, Dept. of Computer Science, Technion, Israel.
- BRONSTEIN, M. M. AND BRONSTEIN, A. M. 2010b. Shape recognition with spectral distances. *Trans. PAMI*. to appear.
- BRONSTEIN, M. M., BRONSTEIN, A. M., KIMMEL, R., AND YAVNEH, I. 2006. Multigrid multidimensional scaling. *Numerical Linear Algebra with Applications (NLAA)* 13, 149–171.
- BRONSTEIN, M. M., BRONSTEIN, A. M., MICHEL, F., AND PARAGIOS, N. 2010e. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *Proc. CVPR*.
- BRONSTEIN, M. M. AND KOKKINOS, I. 2010. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Proc. CVPR*.
- BUSTOS, B., KEIM, D. A., SAUPE, D., SCHRECK, T., AND VRANIĆ, D. V. 2005. Feature-based similarity search in 3D object databases. *ACM Computing Surveys* 37, 4, 387.
- CASTELLANI, U., CRISTANI, M., FANTONI, S., AND MURINO, V. 2008. Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum* 27, 643–652.
- CHAZAL, F., COHEN-STEINER, D., GUIBAS, L. J., MÉMOLI, F., AND OUDOT, S. Y. 2009. Gromov–Hausdorff stable signatures for shapes using persistence. 28, 5, 1393–1403.
- CHAZAL, F., GUIBAS, L. J., OUDOT, S. Y., AND SKRABA, P. 2009. Analysis of scalar fields over point cloud data. In *Proc. SODA*. 1021–1030.
- CHEN, D., OUHYOUNG, M., TIAN, X., AND SHEN, Y. 2003. On visual similarity based 3D model retrieval. *Computer Graphics Forum* 22, 3, 223–232.
- CHUM, O., PHILBIN, J., SIVIC, J., ISARD, M., AND ZISSERMAN, A. 2007. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. ICCV*.
- CLARENZ, U., RUMPF, M., AND TELEA, A. 2004. Robust feature detection and local classification for surfaces based on moment analysis. *Trans. Visualization and Computer Graphics* 10, 5, 516–524.
- COIFMAN, R. R. AND LAFON, S. 2006. Diffusion maps. *Applied and Computational Harmonic Analysis* 21, 5–30.
- DALAL, N. AND TRIGGS, B. 2005. Histograms of oriented gradients for human detection. In *Proc. CVPR*.
- DUBROVINA, A. AND KIMMEL, R. 2010. Matching shapes by eigen-decomposition of the Laplace–Beltrami operator. In *Proc. 3DPVT*.
- EDELSBRUNNER, H., LETSCHER, D., AND ZOMORODIAN, A. 2000. Topological persistence and simplification. In *Proc. IEEE Foundations of Computer Science*. 454–463.
- ELAD, A. AND KIMMEL, R. 2001. Bending invariant representations for surfaces. In *Proc. CVPR*. 168–174.
- ELAD, A. AND KIMMEL, R. 2003. On bending invariant signatures for surfaces. *Trans. PAMI* 25, 10, 1285–1295.
- FLOATER, M. S. AND HORMANN, K. 2005. Surface parameterization: a tutorial and survey. *Advances in Multiresolution for Geometric Modelling* 1.
- FREUND, Y. AND SCHAPIRE, R. E. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proc. European Conf. Computational Learning Theory*.
- FUNKHOUSER, T., MIN, P., KAZHDAN, M., CHEN, J., HALDERMAN, A., DOBKIN, D., AND JACOBS, D. 2003. A search engine for 3D models. *TOG* 22, 1, 83–105.
- GEBAL, K., BÆRENTZEN, J. A., AANÆS, H., AND LARSEN, R. 2009. Shape analysis using the auto diffusion function. *Computer Graphics Forum* 28, 5, 1405–1413.

- GELFAND, N., MITRA, N. J., GUIBAS, L. J., AND POTTMANN, H. 2005. Robust global registration. In *Proc. SGP*.
- GLOMB, P. 2009. Detection of interest points on 3D data: Extending the harris operator. In *Computer Recognition Systems 3. Advances in Soft Computing*, vol. 57. Springer Berlin / Heidelberg, 103–111.
- GRAUMAN, K. AND DARRELL, T. 2005. Efficient image matching with distributions of local invariant features. In *Proc. CVPR*. Vol. 2.
- GROMOV, M. 1981. *Structures Métriques Pour les Variétés Riemanniennes*. Number 1 in Textes Mathématiques.
- HARRIS, C. AND STEPHENS, M. 1988. A combined corner and edge detection. In *Proc. Fourth Alvey Vision Conference*. 147–151.
- HILAGA, M., SHINAGAWA, Y., KOHMURA, T., AND KUNII, T. 2001. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proc. Computer Graphics and Interactive Techniques*. 203–212.
- HSU, E. P. 2002. *Stochastic Analysis on Manifolds*. American Mathematical Society.
- ITTI, L., KOCH, C., AND NIEBUR, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *Trans. PAMI* 20, 11.
- JAIN, P., KULIS, B., AND GRAUMAN, K. 2008. Fast image search for learned metrics. In *Proc. CVPR*.
- JIANTAO, P., YI, L., GUYU, X., HONGBIN, Z., WEIBIN, L., AND UEHARA, Y. 2004. 3D model retrieval based on 2D slice similarity measurements. In *Proc. 3DPVT*. 95–101.
- JOHNSON, A. E. AND HEBERT, M. 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *Trans. PAMI* 21, 5, 433–449.
- JONES, P. W., MAGGIONI, M., AND SCHUL, R. 2008. Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels. *PNAS* 105, 6, 1803.
- KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proc. SGP*. 156–164.
- KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2004. Symmetry descriptors and 3D shape matching. In *Proc. SGP*. 115–123.
- KIMMEL, R., ZHANG, C., BRONSTEIN, A. M., AND BRONSTEIN, M. M. 2010. Are MSER features really interesting? *IEEE Trans. PAMI*. submitted.
- KOLOMENKIN, M., SHIMSHONI, I., AND TAL, A. 2009. On edge detection on surfaces,. In *Proc. CVPR*.
- LAFON, S. 2004. Diffusion maps and geometric harmonics. Ph.D. thesis, Yale.
- LAZEBNIK, S., SCHMID, C., AND PONCE, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*.
- LEIBE, B., LEONARDIS, A., AND SCHIELE, B. 2004. Combined object categorization and segmentation with an implicit shape model. In *Proc. Workshop Statistical Learning in Computer Vision*. 17–32.
- LÉVY, B. 2006. Laplace-Beltrami eigenfunctions towards an algorithm that “understands” geometry. In *Proc. Shape Modeling and Applications*.
- LI, Y., ZHA, H., AND QUI, H. 2006. Shape topics: a compact representation and new algorithm for 3D partial shape retrieval. In *Proc. CVPR*.
- LIAN, Z., GODIL, A., AND SUN, X. 2010. Visual similarity based 3D shape retrieval using bag-of-features. In *Proc. SMI*.
- LIAN, Z., ROSIN, P. L., AND SUN, X. 2010. Rectilinearity of 3D meshes. *IJCV*. in press.
- LIPMAN, Y. AND FUNKHOUSER, T. 2009. Möbius voting for surface correspondence. *TOG* 28, 3.
- LOWE, D. 2004. Distinctive image features from scale-invariant key-point. *IJCV* 60, 2, 91–110.
- MAHMOUDI, M. AND SAPIRO, G. 2009. Three-dimensional point cloud recognition via distributions of geometric distances. *Graphical Models* 71, 1, 22–31.
- MARSZAŁEK, M. AND SCHMID, C. 2006. Spatial weighting for bag-of-features. In *Proc. CVPR*.
- MATAS, J., CHUM, O., URBAN, M., AND PAJDLA, T. 2004. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing* 22, 10, 761–767.
- MATEUS, D., HORAUD, R. P., KNOSOW, D., CUZZOLIN, F., AND BOYER, E. 2008. Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. *Proc. CVPR*.
- MÉMOLI, F. 2007. On the use of Gromov-Hausdorff distances for shape comparison. In *Point Based Graphics*.
- MÉMOLI, F. 2009. Spectral Gromov-Wasserstein distances for shape matching. In *Proc. NORDIA*.
- MÉMOLI, F. AND SAPIRO, G. 2005. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics* 5, 313–346.
- MEYER, M., DESBRUN, M., SCHRODER, P., AND BARR, A. H. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and Mathematics III*, 35–57.
- MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. A comparison of text and shape matching for retrieval of online 3D models. *Research and Advanced Technology for Digital Libraries*, 209–220.
- MITRA, N., BRONSTEIN, A. M., AND BRONSTEIN, M. M. 2010. Intrinsic regularity detection in 3D geometry. In *Proc. ECCV*.
- MITRA, N. J., GUIBAS, L. J., GIESSEN, J., AND PAULY, M. 2006. Probabilistic fingerprints for shapes. In *Proc. SGP*.
- NAPOLÉON, T., ADAMEK, T., SCHMITT, F., AND O’CONNOR, N. E. 2007. Multi-view 3D retrieval using silhouette intersection and multi-scale contour representation. In *Proc. Shape Modeling and Applications*.
- NOVOTNÍ, M. AND KLEIN, R. 2003. 3D Zernike descriptors for content based shape retrieval. In *Proc. ACM Symp. Solid Modeling and Applications*. 216–225.
- OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2002. Shape distributions. *TOG* 21, 4, 807–832.
- OVSJANIKOV, M., BRONSTEIN, A. M., BRONSTEIN, M. M., AND GUIBAS, L. J. 2009. Shape Google: a computer vision approach to invariant shape retrieval. In *Proc. NORDIA*.
- OVSJANIKOV, M., SUN, J., AND GUIBAS, L. J. 2008. Global intrinsic symmetries of shapes. In *Computer Graphics Forum*. Vol. 27. 1341–1348.

- PAQUET, E., RIOUX, M., MURCHING, A., NAVEN, T., AND TABATABAI, A. 2000. Description of shape information for 2-D and 3-D objects. *Signal Processing: Image Communication* 16, 1-2, 103–122.
- PATANÉ, G. AND FALCIDIENO, B. 2010. Multi-scale feature spaces for shape analysis and processing. In *Proc. SMI*.
- PAULY, M., KEISER, R., AND GROSS, M. 2003. Multi-scale feature extraction on point-sampled surfaces. In *Computer Graphics Forum*. Vol. 22. 281–289.
- PINKALL, U. AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics* 2, 1, 15–36.
- RAVIV, D., BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. 2007. Symmetries of non-rigid shapes. In *Proc. NRTL*.
- RAVIV, D., BRONSTEIN, A. M., BRONSTEIN, M. M., KIMMEL, R., AND SAPIRO, G. 2010. Diffusion symmetries of non-rigid shapes. In *Proc. 3DPVT*.
- RAVIV, D., BRONSTEIN, M. M., BRONSTEIN, A. M., AND KIMMEL, R. 2010. Volumetric heat kernel signatures. In *Proc. ACM Multimedia Workshop on 3D Object Retrieval*.
- REUTER, M., BIASOTTI, S., GIORGI, D., PATANÉ, G., AND SPAGNUOLO, M. 2009. Discrete Laplace–Beltrami operators for shape analysis and segmentation. *Computers & Graphics* 33, 3, 381–390.
- REUTER, M., WOLTER, F.-E., AND PEINECKE, N. 2005. Laplace-spectra as fingerprints for shape matching. In *Proc. ACM Symp. Solid and Physical Modeling*. 101–106.
- REUTER, M., WOLTER, F.-E., AND PEINECKE, N. 2006. Laplace–Beltrami spectra as “shape-DNA” of surfaces and solids. *Computer Aided Design* 38, 342–366.
- RUBNER, Y., TOMASI, C., AND GUIBAS, L. J. 2000. The earth mover’s distance as a metric for image retrieval. *IJCV* 40, 2, 99–121.
- RUSTAMOV, R. M. 2007. Laplace–Beltrami eigenfunctions for deformation invariant shape representation. In *Proc. SGP*. 225–233.
- SHAKHNAROVICH, G. 2005. Learning task-specific similarity. Ph.D. thesis, MIT.
- SHILANE, P. AND FUNKHAUSER, T. 2006. Selecting distinctive 3D shape descriptors for similarity retrieval. In *Proc. Shape Modelling and Applications*.
- SHILANE, P., MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. The Princeton shape benchmark. In *Proc. SMI*. 167–178.
- SIPIRAN, I. AND BUSTOS, B. 2010. A robust 3D interest points detector based on Harris operator. In *Proc. 3DOR*. Eurographics, 7–14.
- SIVIC, J. AND ZISSERMAN, A. 2003. Video Google: a text retrieval approach to object matching in videos. In *Proc. CVPR*.
- STRECHA, C., BRONSTEIN, A. M., BRONSTEIN, M. M., AND FUÀ, P. 2010. LDA-hash: improved matching with smaller descriptors. *IEEE Trans. PAMI*. submitted.
- SUMNER, R. AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. In *Proc. Conf. Computer Graphics and Interactive Techniques*. 399–405.
- SUN, J., OVSJANIKOV, M., AND GUIBAS, L. J. 2009. A concise and provably informative multi-scale signature based on heat diffusion. In *Proc. SGP*.
- SUNDAR, H., SILVER, D., GAGVANI, N., AND DICKINSON, S. 2003. Skeleton based shape matching and retrieval. In *Proc. SMI*. Vol. 130.
- TAL, A., ELAD, M., AND AR, S. 2001. Content based retrieval of VRML objects - an iterative and interactive approach. In *Proc. Eurographics Workshop on Multimedia*.
- TANGELDER, J. W. H. AND VELTKAMP, R. C. 2008. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications* 39, 3, 441–471.
- THORSTENSEN, N. AND KERIVEN, R. 2009. Non-rigid shape matching using geometry and photometry. In *Proc. CVPR*.
- TOLA, E., LEPESTIT, V., AND FUÀ, P. 2008. A fast local descriptor for dense matching. In *Proc. CVPR*.
- TOLDÒ, R., CASTELLANI, U., AND FUSIELLO, A. 2009. Visual vocabulary signature for 3D object retrieval and partial matching. In *Proc. 3DOR*.
- TORRALBA, A., FERGUS, R., AND WEISS, Y. 2008. Small codes and large image databases for recognition. In *Proc. CVPR*.
- TORRESANI, L., KOLMOGOROV, V., AND ROTHER, C. 2008. Feature correspondence via graph matching: Models and global optimization. In *Proc. ECCV*. 596–609.
- TUNG, T. AND SCHMITT, F. 2005. The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes. *International Journal of Shape Modelling* 11, 1, 91–120.
- VAXMAN, A., BEN-CHEN, M., AND GOTSMAN, C. 2010. A multi-resolution approach to heat kernels on discrete surfaces. *ACM Trans. on Graphics* Vol.29, No. 4 (Proc. SIGGRAPH 2010)..
- VELTKAMP, R. C. AND HAGEDOORN, M. 2001. State of the art in shape matching. *Principles of visual information retrieval*, 87.
- WANG, C., BRONSTEIN, M. M., AND PARAGIOS, N. 2010. Discrete minimum-distortion correspondence problems in non-rigid shape analysis. Research Report 7333, INRIA.
- WARDETZKY, M., MATHUR, S., KÄLBERER, F., AND GRINSPUN, E. 2008. Discrete Laplace operators: no free lunch. In *Conf. Computer Graphics and Interactive Techniques*.
- ZAHARESCU, A., BOYER, E., VARANASI, K., AND HORAUD, R. 2009. Surface feature detection and description with applications to mesh matching. In *Proc. CVPR*.
- ZENG, Y., WANG, C., WANG, Y., GU, X., SAMARAS, D., AND PARAGIOS, N. 2010. Dense non-rigid surface registration using high-order graph matching. In *Proc. CVPR*.
- ZHANG, C. AND CHEN, T. 2001. Efficient feature extraction for 2D/3D objects in meshrepresentation. In *Proc. ICIP*. Vol. 3.
- ZHANG, H. 2004. Discrete combinatorial Laplacian operators for digital geometry processing. In *Conf. Geometric Design*. 575–592.