A comparison of methods for non-rigid 3D shape retrieval

Zhouhui Lian, Afzal Godil, Benjamin Bustos, Mohamed Daoudi, Jeroen Hermans, Shun Kawamura, Yukinori Kurita, Guillaume Lavoué, Hien Van Nguyen, Ryutarou Ohbuchi, et al.

A comparison of methods for non-rigid 3D shape retrieval

Zhouhui Lian[a,b,*], Afzal Godil[b], Benjamin Bustos[c], Mohamed Daoudi[d], Jeroen Hermans[e], Shun Kawamura[f], Yukinori Kurita[f], Guillaume Lavoué[g], Hien Van Nguyen[h], Ryutarou Ohbuchi[f], Yuki Ohkita[f], Yuya Ohishi[f], Fatih Porikli[i], Martin Reuter[j], Ivan Sipiran[c], Dirk Smeets[e], Paul Suetens[e], Hedi Tabia[k], Dirk Vandermeulen[e]

[a] *Institute of Computer Science and Technology, Peking University, Beijing 100871, China*
[b] *National Institute of Standards and Technology, Gaithersburg, USA*
[c] *Department of Computer Science, University of Chile, Chile*
[d] *Institut TELECOM, France*
[e] *Katholieke Universiteit Leuven, Belgium*
[f] *University of Yamanashi, Japan*
[g] *Université de Lyon, CNRS, France*
[h] *University of Maryland, College Park, USA*
[i] *Mitsubishi Electric Research Laboratories, Cambridge, USA*
[j] *Martinos Center for Biomedical Imaging, Massachusetts General Hospital / Harvard Medical / MIT, USA*
[k] *University Lille 1, France*

Abstract

Non-rigid 3D shape retrieval has become an active and important research topic in content-based 3D object retrieval. The aim of this paper is to measure and compare the performance of state-of-the-art methods for non-rigid 3D shape retrieval. The paper develops a new benchmark consisting of 600 non-rigid 3D watertight meshes, which are equally classified into 30 categories, to carry out experiments for 11 different algorithms, whose retrieval accuracies are evaluated using 6 commonly-utilized measures. Models and evaluation tools of the new benchmark are publicly available on our web site [1].

Keywords: 3D Shape Retrieval; Non-rigid; Benchmark

1. Introduction

With the recent advancement in computer science and technology, 3D models have become widely-used in many application areas, such as computer aided design, multimedia entertainment, electronic commerce, digital library, and so on. Since the number of 3D objects grows rapidly, there exist increasing demands to retrieve them based on their shapes. In the last few

*Corresponding author. Phone: (+86) 10 82529245, fax: (+86) 10 82529207.
Email address: lianzhouhui@yahoo.com.cn (Zhouhui Lian)

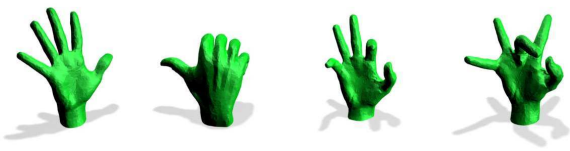

Figure 1: Examples of non-rigid 3D models.

years, the problem of *Non-rigid 3D Shape Retrieval* has become an active research topic in 3D shape retrieval and attracted more and more researchers from several research communities including pattern recognition [2] [3], computer graphics [4] [5], computer vision [6] [7], and applied mathematics [8] [9]. In fact, how to quickly and accurately compare non-rigid 3D shapes is not only important in practice but also interesting in theory. On the one hand, deformable objects are widely-seen in both real and virtual worlds. Take Fig. 1 for an example, a hand can appear in many different poses by articulating around its joints. Those articulated hands are very likely to be recognized as different kinds of objects using many traditional rigid-shape analyzing techniques (e.g., methods compared in the paper [10] that proposed the Princeton Shape Benchmark). On the other hand, many elegant mathematical tools, such as Singular Value Decomposition [11], Multidimensional Scaling [12], Heat Kernel diffusion [13], Laplace-Beltrami operator [14], etc., are well suited for the analysis of non-rigid 3D shapes. Usually, creating an isometry-invariant 3D shape descriptor can be formulated as an interesting mathematical problem.

As the number of algorithms for non-rigid 3D shape retrieval increases rapidly, it is often required to compare them in a fair and effective way. However, most of these methods need to be implemented on watertight manifolds, while both collecting and creating large amounts of those kinds of deformable models are not trivial. Until recently, the most commonly-used non-rigid 3D shape benchmark (i.e., McGill 3D Shape Database [15]) contains only 255 models. That somehow hinders the further investigation in this research direction. To address the problem, we designed an efficient processing framework to build a new large-scale benchmark consisting of 600 watertight triangle meshes and organized a contest called *SHREC'11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes*. In this contest, we asked each participant to submit up to 5 distance matrices obtained using their methods within one week. Finally, 11 different algorithms were proposed by 9 research groups and their retrieval accuracies were

evaluated and compared based on 6 standard measures. This article is an extended version of the conference paper [16] presented for the contest.

Major contributions of this paper are threefold. First, we propose an efficient processing framework to generate large numbers of non-rigid 3D watertight meshes, based on which we build a new non-rigid 3D shape benchmark. Second, we describe and implement a set of methods that roughly represent the state of the art in non-rigid 3D shape retrieval. Third, we evaluate and compare the performance of these methods using the new benchmark.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 describes how to generate the database for our benchmark. Section 4 mentions how to evaluate the retrieval performance and Section 5 provides the information of contributors for the benchmark. Then, we present 11 non-rigid 3D shape retrieval methods in Section 6, followed by Section 7 which demonstrates and analyzes experimental results. Finally, we conclude the paper in Section 8.

2. Related Work

2.1. 3D Shape Retrieval Techniques

The explosion in the number of available 3D models has led to the rapid development of 3D shape retrieval systems that, given a query object, retrieve similar 3D objects [8]. Up to now, a large number of methods for 3D shape retrieval have been proposed, such as Shape Distribution (D2) [4], Spherical Harmonic Descriptor (SHD) [17], Light Field Descriptor (LFD) [18], Elevator Descriptor (ED) [19], Shape Impact Descriptor (SID) [20], etc. However, most of these algorithms are only suitable for the retrieval of rigid 3D shapes and how to accurately and efficiently calculate the similarity between non-rigid models is still considered to be a challenging problem. For more information about 3D shape retrieval, we refer the reader to some recent surveys [8] [10].

In general, existing methods for non-rigid 3D shape retrieval can be roughly classified into algorithms employing local features, topological structures, isometry-invariant global geometric properties, direct shape matching, or canonical forms. The first solution is to measure the dissimilarity between two models based on their local features that are insensitive to isometric transformations. For instance, the well-known Spin Images [21] was utilized

in [22], where they described a 3D object as a word histogram by the vector quantization of all local features (Spin Images) extracted on the surface. Ovsjanikov *et al.* [23] made use of the Heat Kernel Signature (HKS) [13], which is based on the properties of the heat diffusion process on a 3D shape, and designed a spatially-sensitive bag-of-features approach to retrieve non-rigid models in large databases. Ohbuchi *et al.* [24] proposed a view-based method using salient local features (SIFT [25]). They represented a whole object as a histogram by using bag-of-features for 2D salient local descriptors extracted from a set of depth-buffer views captured uniformly around the object. More recently, Wang *et al.* [26] presented Intrinsic Spin Images (ISIs) by generalizing the traditional Spin Images [21] from 3D space to N-dimensional intrinsic shape space, in which their ISIs shape descriptors are calculated on MDS embedding representations of original 3D surfaces.

The second solution is to use topological structures to compare deformable 3D objects. For example, Hilaga *et al.* [27] developed the Topology Matching technique to compute the similarity between two models via the shape matching of their Multiresolutional Reeb Graphs (MRGs), while Sundar *et al.* [28] compared 3D objects by applying graph matching techniques to match their skeletons. Better retrieval performance can be obtained [29] by using topological and geometric features together.

For the third category, isometric-invariant global geometric properties (e.g., geodesic distance) are utilized for non-rigid 3D shape retrieval. Reuter *et al.* [30] suggested using the model's Laplace-Beltrami spectra, while Jain and Zhang [31] proposed to use eigenvalues of the geodesic distance matrix of a 3D object to generate 3D shape descriptors that are isometry-invariant. Also, Mahmoudi and Sapiro [32] designed six such signatures based on the distributions of intrinsic distances including diffusion distance, geodesic distance, a curvature weighted distance, etc.

Many investigations have also been made trying to measure the exact dissimilarity between non-rigid 3D models. For instance, Mémoli and Sapiro [9] introduced a theoretical framework to directly compare non-rigid 3D shapes based on the Gromov-Hausdorff (GH) distance. Since calculating the exact value of the GH distance is computationally expensive, Mémoli [33] proposed to approximate the GH distance by solving a mass transportation problem, which is a quadratic optimization problem with linear constraints. Bronstein *et al.* [34] formulated the GH distance as a MDS-like continuous optimization problem, leading to a numerically exact

calculation of the GH distance between surfaces. Apparently, an ideal and complete solution for the comparison of two non-rigid shapes is to match them directly. However, due to its high computational complexity, direct shape matching is impractical for real shape retrieval systems that require instant responses.

The utilization of canonical forms is also a promising solution for non-rigid 3D shape retrieval. Indeed, with canonical forms, any shape searching algorithm can be applied for the retrieval of non-rigid models. As we know, excellent performance, in term of both accuracy and efficiency, has been achieved for rigid 3D shape retrieval. Obviously, if it is possible to construct canonical forms with well-preserved features, the problem of non-rigid 3D shape retrieval could be well resolved. The idea of generating canonical forms in 3D domain was initially proposed in [35], where the authors introduced an invariant representation for isometric surfaces by applying MDS embedding to map the original surface to a small dimensional Euclidean space in which geodesic distances can be approximated by Euclidean ones. In [35], three MDS techniques were discussed and compared to construct such 3D canonical forms. To verify the effectiveness of their canonical forms, they [35] computed a moment-based shape descriptor from embedded surfaces and carried out a simple experiment for object classification. More recently, Lian *et al.* [36] presented a framework for non-rigid 3D shape retrieval based on the combination of Least Square MDS embedding and a visual similarity based approach. Thanks to the utilization of canonical forms, superior performance was obtained in [36] compared to other existing methods.

2.2. 3D Shape Retrieval Benchmarks

To evaluate and compare the performance of methods for 3D shape retrieval, a number of benchmarks have been developed. Meanwhile, the 3D SHape REtrieval Contest (SHREC) [37], an annual event started from 2006, also provides researchers various resources to compare their algorithms. In this section, we discuss several representative benchmarks that are commonly-used in the area of 3D shape retrieval.

The Princeton Shape Benchmark (PSB) [10] is widely acknowledged as the most prominent 3D shape retrieval benchmark. Evaluation measures they proposed [10] have already become standard methods to evaluate retrieval performance. The PSB database contains 1814 polygonal models that are divided into separate training and test sets, and it also in-

cludes a set of hierarchical classifications. The PSB test set with base classification, which is mostly-utilized in the literature, consists of 907 generic models that are classified into 92 categories. The maximum number of objects in a class is 50, while the minimum number is 4.

Although the PSB benchmark was built rigorously, it still contains many limitations, such as too few models in some classes, evaluation biases caused by unequal number of models in each class, and so on. To solve those problems, Fang *et al.* [38] proposed the NIST (National Institute of Standards and Technology) Shape Benchmark (NSB). The NSB benchmark is composed of 800 generic models that are classified into 40 categories mainly based on their visual similarities, and each class contains an equal number (20) of models. They also carried out experiments to validate the reliability of their benchmark.

The PSB and NSB benchmark were both designed for the retrieval of generic 3D models that can be any kinds of objects in the world. However, there are also requirements for many specific applications, like the retrieval of CAD models, architectural models, protein models, etc. Among them, how to search for desired CAD models is one of the most important applications in content-based 3D object retrieval. Jayanti *et al.* [39] developed the Engineering Shape Benchmark (ESB) that consists of 867 3D CAD models. They classified the database into 45 categories. It has maximum 58 and minimum 4 objects in a class.

Since many 3D feature extraction methods employ elegant mathematical tools which can only be applied on watertight manifolds, there exist increasing demands for such benchmarks that are purely composed of 3D watertight meshes. Giorgi *et al.* [40] organized the watertight model retrieval track under SHREC 2007 by using their database that is made of 400 watertight polygonal models, classified into 20 classes of 20 models each. They manually built the ground truth to ensure that the classes exhibit sufficient and diverse variations, from pose change to shape variability in the same semantic group.

As mentioned above, non-rigid 3D shape retrieval has become an active topic in many research communities. The most commonly-used benchmark for this topic is the McGill articulated 3D shape Benchmark (McGill) [15]. However, the McGill database merely contains 10 categories, totally 255 articulated 3D meshes. When performing statistical analysis, the size of the database is typically too small to reduce possible bias and establish accurate evaluation. Furthermore, the largest class in the McGill database has 31 models while the

smallest class contains 20 objects. The unequal number of models in each class could also cause evaluation bias. In this paper, we propose an effective scheme to create articulated watertight meshes and develop a large-scale database consisting of 600 non-rigid watertight meshes with 20 models in each class.

3. Data Creation

The new benchmark consists of 600 watertight triangle meshes that are derived from 30 original models, among which 26 objects are collected from several freely-accessible repositories (e.g., PSB database [10], McGill database [15], TOSCA shapes [12], etc.) while the other 4 models (i.e., *lamp, paper, scissor*, and *twoballs*) are created by us using *Autodesk 3d Max*. Given a 3D mesh, we use *Autodesk 3d Max* to build its skeleton and then generate 19 deformed versions of the mesh by articulating parts around its joints in different ways (see Fig. 3). To remove the inner structures of those articulated models, as shown in Fig. 4, we implement our own software to first capture 18 depth-buffer views for the normalized object on the vertices of a unit geodesic sphere, and then convert those images into a point cloud. Finally, we wrap the point cloud into a polygon surface and fix it to form a watertight 3D manifold without any topological errors by using *Geomagic*, which can be automatically implemented with recorded macros. As shown in Fig. 2, those 600 non-rigid models have been equally classified into 30 categories.

4. Evaluation

When running algorithms on the database, we first calculate the dissimilarity value between every two objects, and then generate a distance matrix for each method. The matrix is composed of 600×600 floating point numbers, where the number at position (i, j) represents the dissimilarity between models i and j. Analyzing those distance matrices, we evaluate their retrieval performance based on the Precision-recall curve and other five quantitative measures (see [10] for detailed definitions):

- **Precision-recall curve:** A curve illustrating the relationship between the precision and recall of a retrieval algorithm. Precision is the percentage of retrieved objects that are relevant, while recall is the percentage of relevant models that are retrieved.

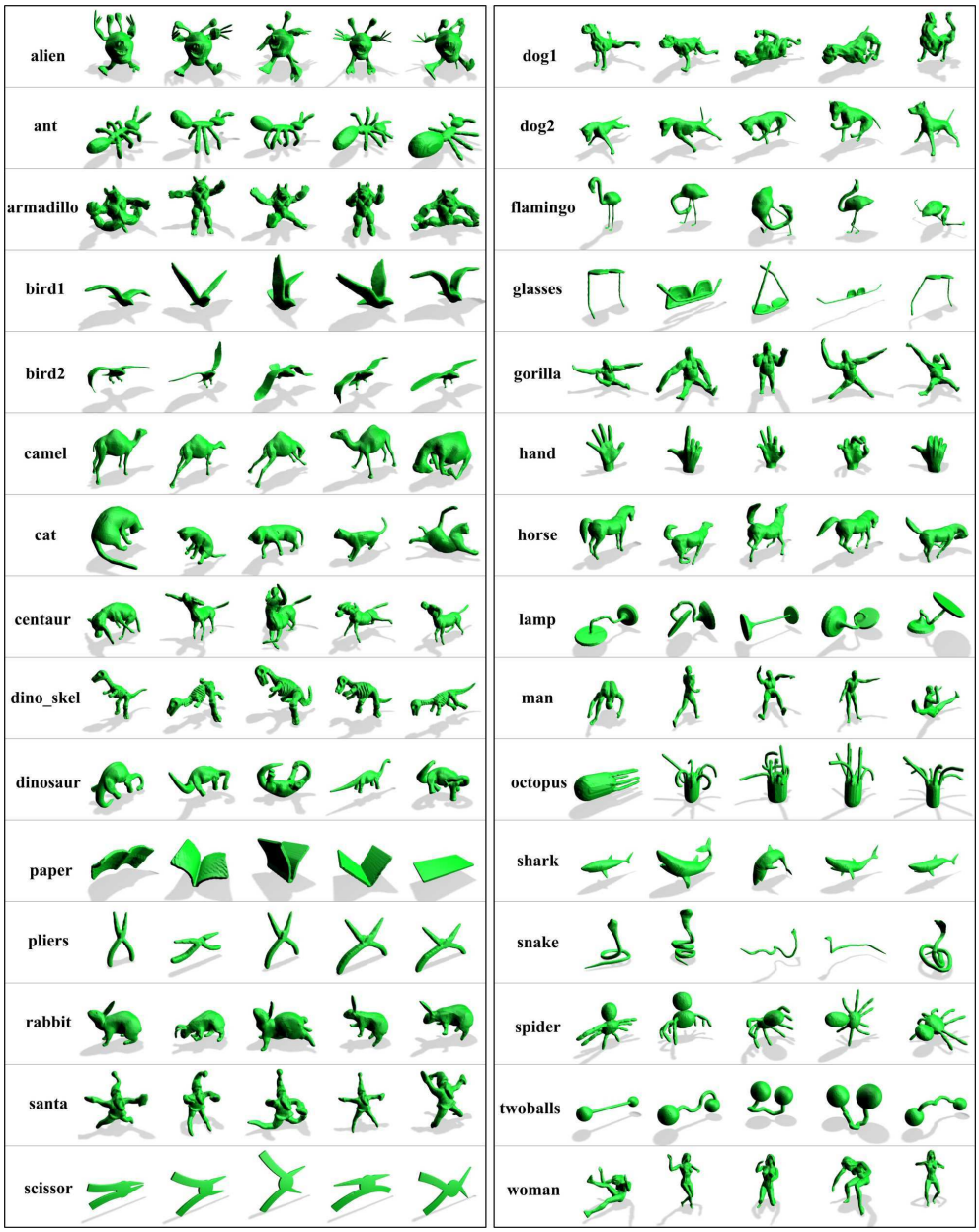

Figure 2: Examples of models in our database that is classified into 30 categories.

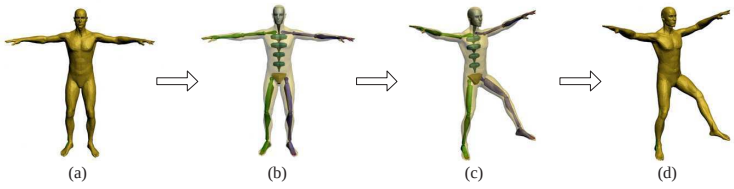

Figure 3: Generating an articulated model for a 3D mesh. Given the original mesh (a), we first build its skeleton (b), and then articulate (c) the model around its joins to obtain the new mesh (d).

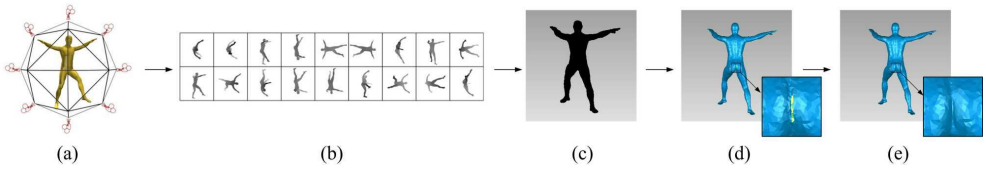

Figure 4: Creating the watertight manifold for a 3D mesh. We first normalize the original model (a) so that it is located inside a unit geodesic sphere, and then a set of depth-buffer views (b) of the 3D mesh are captured on the 18 vertices of the bounding sphere. Afterwards, we convert those depth-buffer images into a point cloud (c), which is then utilized to generate a polygonal mesh (d). Finally, by fixing holes and other errors on the surface (d), we obtain a watertight manifold (e) without inner structures.

- **Nearest Neighbor (NN):** The percentage of closest matches that are in the same class as the query.

- **First Tier (FT) and Second Tier (ST):** The percentage of models belonging to the query's class that appear within the top N_T retrieved models, where N_T depends on the size of the query's class. More specifically, assume that the query's class contains N_C models, $N_T = N_C - 1$ for the First Tier and $N_T = 2(N_C - 1)$ for the Second Tier.

- **E-measure (E):** A composite measure of the precision and recall for a fixed number of retrieved models. The E-Measure is defined as $E = \frac{2}{\frac{1}{P}+\frac{1}{R}}$, where P and R are the precision and recall, respectively, computed for the first 32 retrieved models.

- **Discounted Cumulative Gain (DCG):** A statistic that measures the usefulness (i.e., gain) of a retrieved model based on its position in the ranked list. The gain is accumulated from the top of the ranked list to the bottom with the gain of each result reduced logarithmically proportional to the position of the result.

5. Contributors

The first two authors of this paper built the benchmark and organized the *SHREC'11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes*. Totally, 9 groups took part in the contest and implemented the following 11 methods.

1. *FOG*: submitted by Shun Kawamura, Yukinori Kurita and Ryutarou Ohbuchi from University of Yamanashi, Japan.

2. *BOW-LSD*: submitted by Guillaume Lavoué from Université de Lyon, CNRS, France
3. *MDS-CM-BOF*: submitted by Zhouhui Lian from Peking University, China, and Afzal Godil from National Institute of Standards and Technology, USA.
4. *BOGH*: submitted by Hien Van Nguyen from University of Maryland, College Park, USA and Fatih Porikli from Mitsubishi Electric Research Laboratories, USA.
5. *LSF*: submitted by Yuki Ohkita, Yuya Ohishi, Shun Kawamura and Ryutarou Ohbuchi from University of Yamanashi, Japan.
6. *ShapeDNA*: submitted by Martin Reuter from Martinos Center for Biomedical Imaging, Massachusetts General Hospital / Harvard Medical / MIT, USA.
7. *Harris3DGeoMap* and *HKS*: submitted by Ivan Sipiran and Benjamin Bustos from University of Chile, Chile.
8. *MeshSIFT* and *SD-GDM*: submitted by Dirk Smeets, Jeroen Hermans, Dirk Vandermeulen and Paul Suetens from Katholieke Universiteit Leuven, Belgium.
9. *PatchBOF*: submitted by Hedi Tabia from University Lille 1, France and Mohamed Daoudi from Institut TELECOM, France.

6. Methods

This section describes the methods we compared.

6.1. FOG: Features on Geodesics, by S. Kawamura, Y. Kurita and R. Ohbuchi

The Features on Geodesics (FOG) algorithm is based on a diffusion-like distance on 3D mesh surface to achieve robustness against articulation. In addition, the FOG is designed to accept diverse surface-based 3D models, e.g., non-watertight mesh or polygon-soup.

To compute features, the FOG method first resamples the surface of a model by uniformly and quasi-randomly generating N_{sp} oriented points ($N_{sp} \approx 3000$). These points are then reconstructed into a mesh by using k-nearest neighbor connectivity. This remeshing gains invariances to shape representation and tessellation, in exchange for retrieval accuracy.

After remeshing, the algorithm computes a set of local-FOG features at N_k ($N_k \approx 500$) randomly-selected key-points on the mesh by using the Manifold Ranking algorithm developed by Zhou *et al.* [41]. The manifold ranking algorithm is originally designed to compute distances among features in high dimensional feature space. The k-nearest neighbor meshing in the

feature space of the original manifold ranking algorithm is replaced with the mesh resampling mentioned above.

For each key-point, a local-FOG is computed as a set of geodesic-like distances for vertices that lie within a radius r sphere of interest (using 3D Euclidian distance). A local-FOG feature centered at the key-point captures local geometry at multiple scales, by having multiple radius of interest r and multiple parameters σ that controls the diffusion speed during the computation of manifold ranking.

A histogram of these distances coupled with a local geometrical feature within the same sphere becomes the local-FOG feature at the key-point. A set of N_k local FOG features are integrated into a feature vector per 3D model by using the bag-of-words approach. For the FOG algorithm, *Kullback-Leibler Divergence* is used to compute the distance between two features.

6.2. BOW-LSD: Bag of Words with Local Spectral Descriptors, by G. Lavoué

The method [42] is based on the Bag of Words (BoW) paradigm. For a given 3D shape, the proposed approach considers a set of feature points uniformly sampled on the surface and associated with local Fourier descriptors; this descriptor is computed in the neighborhood of each feature point by projecting the geometry onto the eigenvectors of the Laplace-Beltrami operator. In a preliminary step, a visual dictionary is built by clustering a large set of feature descriptors, then each 3D shape is described by an histogram of occurrences of these visual words.

In this method, a uniform sampling is first utilized to generate feature points on the mesh surface; for this goal, a random set of n_p vertices on the mesh is considered as an initial set of *seeds*, and then Lloyd relaxation iterations are implemented. Lloyd's algorithm [43] is a fixed-point iteration that simply consists of iteratively moving the seeds to the centroids of their Voronoi cells. Each feature point p_i is then associated with a local patch P_i on which a descriptor is calculated. For each feature point, this local patch is extracted by considering the connected set of facets belonging to a given sphere of center p_i and of a given radius r.

After that, each feature point is associated to a descriptor computed on its patch. The Fourier spectra of the patch are computed by projecting the geometry onto the eigenvectors of the Laplace-Beltrami operator. The Laplace-Beltrami operator Δ is the counterpart of

the Laplace operator in Euclidian space. It is defined as the divergence of the gradient for functions defined over manifolds. The eigenfunction and eigenvalue pairs (H^k, λ_k) of this operator satisfy the following relationships:$-\Delta H^k = \lambda_k H^k$. In the case of a 2-manifold triangular mesh the above eigen-problem can be discretized and simplified within the finite element modeling framework [44]:$-Q\mathbf{h}^k = \lambda_k D\mathbf{h}^k$, in which $\mathbf{h}^k$ denotes the vector $[H_1^k, ...H_m^k]$ where m is the number of vertices of the patch. D is the Lumped Mass matrix and Q is the Stiffness matrix. To resolve this discrete eigenproblem, the fast algorithm from Vallet and Lévy [45], based on a band-by-band approach and an efficient eigen-solver, is adopted; hence the eigenvectors $\mathbf{h}^k$ (i.e., the manifold harmonic bases) and the associated eigenvalues are obtained. The spectral coefficients are then calculated as the inner product between the geometry of the surface and the sorted eigenvectors. For $\mathbf{x}$ (resp. $\mathbf{y}$,$\mathbf{z}$):

$$\tilde{x}_k = < \mathbf{x}, \mathbf{h}^k > = \sum_{i=1}^{m} x_i D_{i,i} H_i^k \tag{1}$$

The k^{th} $(k = 1..m)$ spectral coefficient amplitude is then defined as:

$$c_k = \sqrt{(\tilde{x}_k)^2 + (\tilde{y}_k)^2 + (\tilde{z}_k)^2} \tag{2}$$

Thus, for a given patch P_i around a feature point p_i, the descriptor is the spectral amplitude vector $\mathbf{c}^i = [c_1^i, ...c_{n_c}^i]$, with c_k^i, the k^{th} spectral coefficient amplitude of the patch P_i. Here, only the n_c first spectral coefficients are considered to limit the descriptor to low/medium frequencies.

Given a 3D object containing a set of patches P_i associated with descriptors $\mathbf{c}^i$, the next step is to represent it as a distribution of visual words from a given dictionary. First, the visual dictionary is created by clustering a huge dataset of descriptors and keep the n_w centroids $\bar{\mathbf{c}}^k$ of the clusters as visual words. Then, each patch P_i is associated with its closest visual word and the bag of words $\mathbf{b}^M$ of the whole model M is a n_w-dimensional vector containing the distribution of the visual words over all its patches. The matching between two bags of words is simply done using the L_1 distance.

In this paper, settings of the BOW-LSD algorithm are as follows:

- The size n_w of the dictionary was set to 200 and the number of patchs n_p was set to 200.

- The visual vocabulary was computed from the database.
- The radius of the patches is selected as $r = 0.15$ and the number of spectral coefficients is $n_c = 40$.

6.3. MDS-CM-BOF: Visual Similarity based Non-rigid 3D Shape Retrieval Using MDS, by Z. Lian and A. Godil

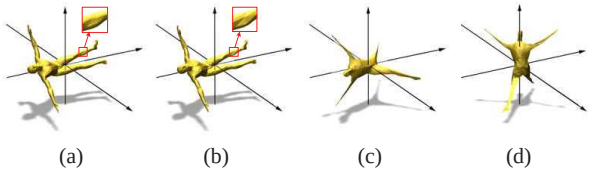

Figure 5: Procedures of the canonical form computation. (a) and (d) show the original 3D model and its canonical form, respectively.

The method [36] performs step by step as follows:

1) Canonical Form Computation: Calculate the canonical form for a 3D model based on MDS and PCA. As shown in Fig. 5, the least squares technique with the SAMCOF algorithm is chosen to implement the MDS embedding (Fig. 5(c)), and before that the number of vertices on the mesh has been reduced to about 1000 (Fig. 5(b)).

2) Local Feature Extraction: Capture 66 depth-buffer views for the canonical form on the vertices of a given geodesic sphere, and then extract salient SIFT descriptors [25] from these views (Fig. 6).

3) Word Histogram Construction: Generate a word histogram by vector quantizing each view's local features against a pre-specified codebook, such that the shape can be represented by a set of histograms. It should be pointed out that the codebook is built by using K-means to create 256 clusters for large numbers of local features randomly sampled from MDS embedded McGill database, and a particular data structure (Fig. 6) is designed to represent the histogram in a more efficient and effective way [46].

4) Dissimilarity Calculation: Carry out an efficient multi-view shape matching (Clock Matching) scheme [6] to measure the dissimilarity between two models by calculating the minimum distance of their 24 matching pairs.

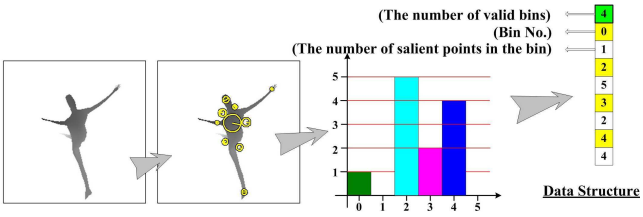

Figure 6: Represent a depth-buffer view as a word histogram by the vector quantization of its SIFT local features.

Since the method is mainly based on *Multidimensional Scaling*, *Clock Matching*, and *Bag-of-Features*, for the sake of convenience, it is denoted as "MDS-CM-BOF". More details of this method can be found in [36] [46] [6].

6.4. BOGH: Bag of Geodesic Histograms, by H.V. Nguyen and F. Porikli

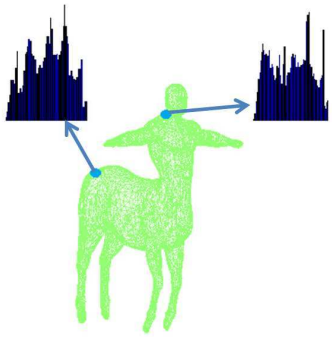

Figure 7: Illustration for the histogram of geodesic distances computed at two points on the centaur model.

The method uses a Bag-of-Feature approach and Normalized Geodesic Distances to retrieval non-rigid 3D shapes.

Consider a shape to be a closed set $\mathcal{S} \in \mathcal{R}^n$, the geodesic distance $\gamma(p,q)$ between two points p and q is defined to be the shortest path among all paths connecting these two points on the shape. Let $\mathrm{h}(p) = [h_1(p), h_2(p), \ldots, h_n(p)]$ denotes the histogram of geodesic distances (see Fig. 7) from the point p to all points in $\mathcal{S}$, which is defined as follows:

$$h_i(p) = \frac{\mathcal{Q}_i}{\mathcal{S}} \tag{3}$$

$$\mathcal{Q}_i = \left\{ q \in \mathcal{S} | (i-1)\Delta \leq \frac{\gamma(p,q)}{\overline{\gamma}_p} \leq i\Delta \right\} \tag{4}$$

where $\overline{\gamma}_p$ is the mean of geodesic distance from p to all points, and Δ is the separation between histogram bins. Here, $n = 100$ and $\Delta = 0.025$. Since the descriptor is based on the geodesic distances, they are robust to various 3D non-rigid articulations. In addition, the normalization with respect to average geodesic distances take into account the scaling effects.

For each shape, N points (here $N = 300$) are randomly chosen and a bag of descriptors is computed. Shape matching is done by first finding the optimal correspondences between their bags of descriptors using the Hungarian algorithm.

Let two sets of the descriptors for two shapes A and B be $\Lambda^A : h_1^A, h_2^A, ..h_N^A$ and $\Lambda^B : h_1^B, h_2^B, ..h_N^B$. The correspondence is established through a one-to-one mapping function τ such that $\tau : \Lambda^A \leftrightarrow \Lambda^B$. If a descriptor h_i^A is matched to another h_i^B then $\tau(i^A) = j^B$ and $\tau(j^B) = i^A$. The cost function is defined as

$$E(h) = \sum_{1 \le i \le N} \epsilon(\tau(i), i) \tag{5}$$

where the distance between two descriptors is computed using χ^2 statistic

$$\epsilon(\tau(i), i) = \sum_{1 \le k \le N} \frac{[h_{\tau(i)}^A(k) - h_i^B(k)]^2}{h_{\tau(i)}^A(k) + h_i^B(k)} \tag{6}$$

Finally, the optimal cost $E(h)$ is used as the similarity measure between two shapes.

6.5. LSF: Localized Statistical Features (LSF), by Y. Ohkita, Y. Ohishi, S. Kawamura and R. Ohbuchi

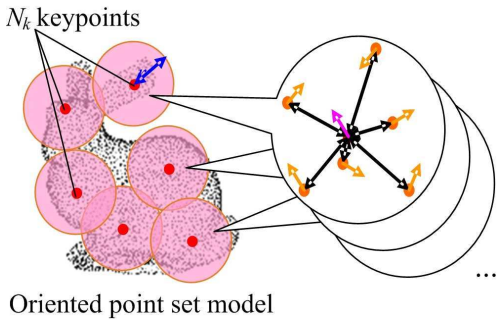

Figure 8: Localized Statistical Features (LSF).

The Localized Statistical Features (LSF) (see Fig. 8) is a very simple 3D shape descriptor that has a set of good robustness properties [47]. The LSF is robust against shape representations; the LSF can handle 3D models represented as polygon soup, oriented point set,

watertight mesh, water leaking manifold mesh, etc. The LSF is robust against similarity transformation without requiring any pose normalization. It is also fairly robust against geometrical/topological noise. Finally, the LSF is robust against articulation.

The LSF computes a set of N_k ($N_k \approx 500$) localized 3D statistical features, which are then combined into a feature vector per 3D model by using the bag-of-words approach. Each statistical feature is a derivative of the Surflet-Pair-Relation Histograms (SPRH) feature by Wahl *et al.* [48]. The SPRH feature accepts a 3D model in oriented point set representation. From the point set, the SPRH computes a 4D joint histogram consisting of three angles (inner product, etc.) and a distance among all the pairs of the oriented points.

For the LSF, the SPRH descriptor is made to be local. Each LSF is computed from the point set within the sphere of radius r about the N_k keypoints quasi-randomly and uniformly placed on the surfaces of the model. In LSF, histogram is computed from point pairs in which one of the points is the keypoint. If there are n points in the sphere, there are $(n-1)$ pairs of points filling the histogram.

After the set of local features are computed, they are combined into a feature vector per 3D model by using the bag-of-features approach. Here, the LSF feature is used as is, i.e., without Manifold Ranking and other distance metric learning.

6.6. ShapeDNA: Laplace Spectra for Non-Rigid Shape Analysis, by M. Reuter

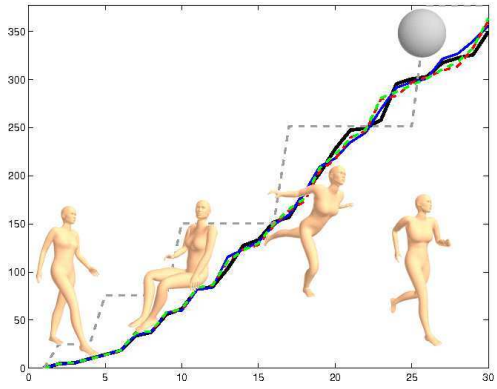

Figure 9: First 30 eigenvalues plotted for the girl in four different postures as opposed to the spectrum of the sphere (gray) with same surface area (the position of the shapes in the plot is irrelevant).

ShapeDNA is the normed beginning sequence of the spectrum, i.e. the first eigenvalues, of the Laplace-Beltrami operator (LBO) for 2D surfaces or 3D solids. It has been introduced

in 2005 [30, 14] as the first spectral shape signature for non-rigid shape analysis. Extensions have later been employed for local shape analysis of structures in the human brain to detect and quantify disease effects [49] and for automatic shape segmentation and part correspondence [50]. Over the past half decade, spectral methods have gained much attention due to their beneficial properties, most importantly their *isometry invariance* making them robust with respect to pose differences (see Fig. 9). Many methods based on the LBO have emerged for shape processing and to generate local and global shape signatures, commonly based on both eigenvalues and eigenfunctions, e.g., to approximate the heat kernel.

The relation between shape and sound has been of great interest in the past. The mathematician Bers first asked the famous question "Can one hear the Shape of a Drum?" in a talk [51]. This question inspired Kac who later published an article [52] with the same title, asking if the shape of a planar domain is in fact determined by the spectrum of the Laplacian. He proved this to be true for disks. The idea to connect the eigenvalues (the spectrum) with geometric entities dates back at least to Weyl [53] who showed that the asymptotic behavior of the eigenvalues depends on the surface area of the drum (or on the volume of a 3D solid). Obviously the natural frequencies and thus the sound of a drum are determined by its shape (and material properties, ignored here). Except for a limited class of shapes known to be spectrally determined, however, the reverse is not necessarily true (see e.g. [54] for artificially constructed isospectral twins).

In addition to the isometry invariance, the beginning sequence of the Laplace spectra has many desirable properties. This descriptor is insensitive to noise, which influences mainly the higher eigenvalues. Potential switching of eigenvalues due to small non-isometric deformations is not problematic (as opposed to comparing eigenfunctions), since the values must have been close to begin with. As a vector of numbers the spectrum can be compared easily and can be computed for many different shape representations and dimensions. It can deal with objects containing cavities (when using 3D solids), depends continuously on shape deformations and can easily be made scaling invariant. Note that the ShapeDNA does not rely on any prior knowledge and in contrast to other methods involving eigenfunctions or the heat kernel, it yields a very simple and robust, isometry invariant shape descriptor.

Mathematically the eigenvalues λ and eigenfunctions u are defined as the solutions of the Laplacian eigenvalue problem $\Delta u = -\lambda u$, where $\Delta u := div(grad(u))$ with $grad$ being the gra-

dient and *div* the divergence with respect to the underlying domain or Riemannian manifold in general. Here the normed first smallest N eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ are taken as the shape descriptor (ShapeDNA). The spectrum of a given shape can be approximated very efficiently even if the shape exhibits not-flat geometry. Many different discretizations of the Laplace-Beltrami operator exist ([55] show a comparison of some common discretizations). These range from simple graph Laplace operators considering only the connectivity, metric aware linear mesh operators, to higher order discretizations based on finite elements methods (FEM), introduced in [14] (see also [49] appendix for implementation details on triangle meshes). In this paper, the simple *linear FEM* discretization is utilized to compute the first eigenvalues of the LBO. Since for shape retrieval only a small number of eigenvalues is needed linear approaches should be sufficient. Note that in order to compute a large number of eigenvalues and eigenfunctions, e.g. to approximate the heat kernel, higher order approximations may be required due to their superior accuracy [55].

For the ShapeDNA, in addition to the LBO discretization, several parameters can be specified. Earlier tests showed that usually $N = 10 \cdots 15$ eigenvalues are a good number (less have often not enough power to distinguish shapes, while including higher values increases influence of noise and non-isometric deformations). The first eigenvalue is omitted as it is zero for closed manifolds. Another parameter is the distance metric to compare the spectra, where the simple *Euclidean distance* on the N dimensional vector of numbers is chosen. Other meaningful distances (Hausdorff, correlation, different p-norms) did not improve retrieval rates in prior tests. Finally, in order to compare shape rather than size of the objects, the spectra need to be normalized. One option is to multiply the spectrum by the *surface area* (applied here), which is the same as normalizing the area of the shapes before computation. Another option is to divide the sequence by the *first non-zero eigenvalue*, which has the same effect in perfectly isometric cases. However, shapes are usually not isometric and dividing by the first non-zero eigenvalue may help to identify similar shapes in spite of noise or near-isometric deformations. On the other hand it puts a large emphasis on the first value.

Software to compute eigenvalues and eigenvectors of the Laplace-Beltrami operator with up to cubic FEM on triangle meshes has been made freely available for non-profit research at [56].

6.7. Harris3DGeoMap and HKS: by I. Sipiran and B. Bustos

This section presents two techniques, including Harris 3D and Heat Kernel Signatures methods, to tackle the problem of non-rigid 3D shape retrieval.

6.7.1. Harris3DGeoMap: Harris 3D Geodesic Map

The idea behind this method is to compute a characteristic distribution of geodesic distances between the interest points of a shape. So the method starts by detecting interest points of a shape using the Harris 3D method [57]. For this paper, adaptive neighborhoods with $\delta = 0.01$ are utilized and the 0.01% of the number of vertices with the highest Harris response are selected as interest points.

Let F be the set of interest points detected, the complete set of geodesic distances between each pair of interest points is computed. This set is represented by the matrix D of dimension $|F| \times |F|$. Values in the matrix are normalized through dividing each entry by the maximum value. This makes the values invariant against scale.

Next, a histogram is created with n bins, which divides the interval $[0, 1]$ of possible normalized geodesic distances. Then, m samples are randomly selected from the matrix D, accumulating a vote in their corresponding bins. Configurations chosen to compute the histograms are as follows: $n = 32$, $m = 2000$. The distance between two histograms is measured using the Euclidean distance.

6.7.2. HKS: Heat Kernel Signatures based Point-to-point Matching

Heat kernel signatures method (HKS) [13] has proven to be an interesting mesh analysis tool. Unlike Harris 3D, HKS computes a descriptor for each vertex on a mesh. These descriptors are invariant to non-rigid transformations, allowing to detect interest points too.

The method starts by detecting the interest points using the Heat kernel signatures. In this paper, descriptors of length 100 are used and $t = 0.1$ of the area of the surface is considered as the value for comparing the HKS for interest point detection. Once the interest points have been detected, each interest point has an associated HKS descriptor. Then, a shape is represented by a set of HKS descriptors associated to the interest points.

As HKS is based on an intrinsic formulation of a mesh, the descriptors are expected to be very similar in presence of non-rigid transformations. Based on this fact, the set of descriptors

of two shapes are compared. Let $S = \{s_1, s_2, \cdots, s_n\}$ and $P = \{p_1, p_2, \cdots, s_m\}$ be the sets of descriptors of two shapes. The dissimilarity between S and P is defined as

$$d(S, P) = \frac{\sum_{k=1}^{n} d_{min}(s_k, P)}{n} \tag{7}$$

where

$$d_{min}(s_i, P) = \min_{s_j \in P} \|s_i - s_j\|_2 \tag{8}$$

6.8. SD-GDM and MeshSIFT: by D. Smeets, J. Hermans, D. Vandermeulen and P. Suetens

In this section, two methods are presented for non-rigid 3D object recognition. The first, which we will call *SD-GDM*, is a global feature method based on an intrinsic object representation, invariant for isometric deformations (Sect. 6.8.1). The second method, *MeshSIFT*, on the other hand is a local feature method describing local neighborhoods of interest points on the surface (Sect. 6.8.2).

6.8.1. SD-GDM: Spectral Decomposition of the Geodesic Distance Matrix

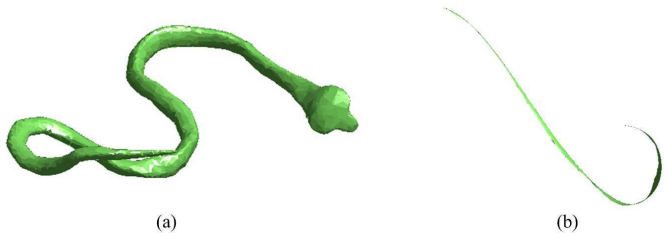

Figure 10: A snake model (a) after preprocessing and its canonical form (b).

The SD-GDM approach was introduced by Smeets et al. for 3D non-rigid object recognition [11] and for 3D face recognition [58].

As preprocessing, the surface meshes are downsampled to about 3000 points, by reducing the number of faces without moving the remaining points. Next, duplicated and isolated vertices are removed in the surface mesh. The preprocessed mesh of a snake model is shown in Fig. 10(a).

The 3D shapes are then represented by a geodesic distance matrix (GDM), $G = [g_{ij}]_{n \times n}$, which is a isometric deformation invariant matrix. It contains the geodesic distance g_{ij} between each pair of points on the surface. This distance is the length of the shortest path on the object surface between two points on the object. Isometric deformations leave these geodesic

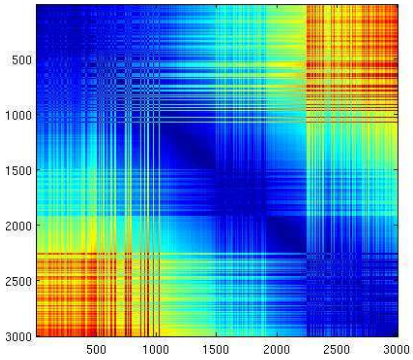

Figure 11: The normalized geodesic distance matrix of the preprocessed snake model, which was shown in Fig. 10 (a).

distances unchanged. The geodesic distances are calculated with a fast marching algorithm for triangulated meshes [59, 60]. To compensate for scale differences in the 3D shapes, the geodesic distances are normalized by the square root of the total surface area of the mesh. An example of such a normalized geodesic distance matrix is given in Fig. 11.

Next, spectral decomposition (SD) of the GDM provides a sampling order invariant global feature (shape descriptor). In [11], it is proved that the modal representation, i.e. the eigenvalue matrix, is invariant to the sampling order under the condition that each point on one surface has one corresponding point on the other surface, which can be assumed for watertight meshes after resampling. Object recognition reduces to direct comparison of the shape descriptors without the need to establish explicit point correspondences. As a trade-off between sensitivity for shape variations and noise robustness, the 40 largest eigenvalues are computed. As such, the complexity of the spectral decomposition is reduced to $\mathcal{O}(kn^2)$, with $k = 40$ and $n \approx 3000$. The modal representations of the normalzied GDMs are then compared using the mean normalized Manhattan distance as in [11].

Strictly speaking, the SD-GDM method is not a spectral embedding technique, since the canonical shape is not computed explicitly. Even if the eigenvectors would be computed, embedding is not straightforward because the normalized GDM is not positive (semi)definite (unlike the Laplacian matrix or the weighted Euclidean distance matrix). By taking the absolute value of the eigenvalues, this equivalent canonical shape can be computed and visualized (Fig. 10(b)).

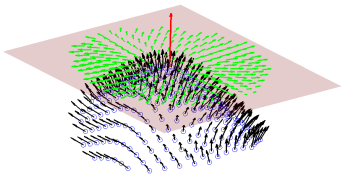

Figure 12: The neighborhood of a scale space extremum with normals and projected normals

The normalized GDM, however, has the advantage for non-rigid object recognition of the importance of the larger, and therefore less noise-sensitive, distances. Exponential weightings of the geodesic distance, $\exp(-g_{ij}^2/\sigma^2)$, generally provide positive (semi)definite matrices, but are shown to give worse results for non-rigid object recognition in [11].

6.8.2. MeshSIFT: Scale Invariant Feature Transform for meshes

Similar to the scale invariant feature transform (SIFT) algorithm [25], the meshSIFT algorithm [61] consists of four major components: keypoint detection, orientation assignment, the local feature description and feature matching.

The algorithm first identifies salient points on the mesh, by constructing a scale space that contains smoothed versions of the input mesh. These smoothed versions are obtained by approximating a Gaussian filter for meshes as subsequent convolutions of the mesh with a binomial filter. Next, for the detection of salient points in the scale space, the mean curvature H is computed for each vertex and at each scale in the scale space (H_i). Note that the mesh itself is smoothed and not the function on the mesh (H). Extrema (minima and maxima) in scale spaces of differences between subsequent scales ($dH_i = H_{i+1} - H_i$) are selected as local feature locations. Finally, the correct scale, which corresponds with some amounts of smoothing, is assigned to each scale space extremum, leading to a keypoint with an assigned scale.

In order to obtain an orientation-invariant descriptor, each keypoint is assigned a canonical orientation. By expressing the neighborhood size in function of the scale, we ensure a scale invariant descriptor as well. First, for each vertex within this region, the normal vector is computed (using [62]) and the geodesic distance to the respective keypoint (using [60]) is determined. Next, as shown in Fig. 12, all calculated normal vectors are projected onto the tangent plane to the mesh containing the keypoint. These projected normal vectors are

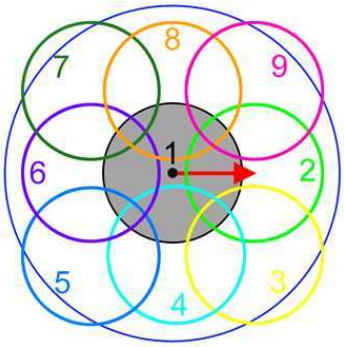

Figure 13: Location and order of the regions w.r.t. the canonical orientation, used for the construction of the feature vector.

gathered in a weighted histogram comprising 360 bins. Each histogram entry is Gaussian weighted with its geodesic distance to the keypoint. The resulting histogram is smoothed by convolving it three times with a Gaussian filter for a more accurate and robust localization of the canonical orientation. Finally, the highest peak in the histogram and every peak above 80% of this highest peak value is selected as a canonical orientation. If more than one canonical orientation exists for a keypoint, this results in multiple keypoints, each assigned one of the canonical orientations.

The local descriptor provides for each keypoint (with assigned scale and canonical orientation) a feature vector consisting of concatenated histograms. Each of these histograms is calculated over a small circular region, as shown in Fig. 13. In each region two histograms with 8 bins each are computed. The first contains the shape index, which is a combination of minimum and maximum curvature. The second contains the slant angles, which are defined as the angle between every projected normal and the canonical orientation. First, each entry for both histograms is Gaussian weighted with the geodesic distance to the keypoint and with the geodesic distance to the center of the region. Next, every histogram is normalized and clipped, reducing the influence of large histogram values. Finally, the histograms are concatenated in one feature vector.

In order to find corresponding features, two sets of feature vectors are compared using the angle as similarity measure. The angles of all candidates are then ranked in ascending order. If the ratio between the first and the second is smaller than 0.9, a match is accepted; other matches are rejected. The number of matches is simply used as similarity criterion. The

similarity matrix is converted into a dissimilarity matrix by subtracting the matrix from the maximum number of matches.

6.9. PatchBOF: Bag-of Densely-Sampled Local Visual Features, by H. Tabia and M. Daoudi

The method consists of the following four steps (see [63] for more details):

1) Detection and description of 3D patches: Let v_1 and v_2 be the farthest vertices (in the geodesic sense) on a connected triangulated surface S. Let f_1 and f_2 be two scalar functions defined on each vertex v of the surface S, as follows: $f_1(v) = d(v, v_1)$ and $f_2(v) = d(v, v_2)$ where $d(x, y)$ is the geodesic distance between points x and y on the surface. In a critical point classification, a local minimum of $f_i(v)$ is defined as a vertex v_{min} such that all its level-one neighbors have a higher function value. While, a local maximum is a vertex v_{max} such that all its level-one neighbors have a lower function value. Let F_1 be the set of local extrema (minima and maxima) of f_1 and F_2 be the set of local extrema of f_2. The set of feature points F of the triangulated surface S is defined as the closest intersecting points in the sets F_1 and F_2. Given a 3D object O, for every feature point $F_i \in F$, a descriptor $P(F_i)$ is defined for F_i and the geodesic distances $\{d(F_i, v); \forall v \in V\}$ with V is the set of all vertices on the surface are calculated. Consider f the distribution of vertices according to these distances, the descriptor $P(F_i)$ is defined as a R-dimensional vector: $P(F_i) = (p_1, \ldots, p_R)$ where $p_r = \int_{(r-1)/R}^{r/R} f(d)\delta d$. $P(F_i)$ is a R-bin histogram of vertex distribution of geodesic distances measured from F_i. In order to make the descriptors comparable between different shapes, the geodesic function d is scaled by the geodesic diameter of the shape.

2) Shape vocabulary construction: The vocabulary used in this method is a way of constructing a feature vector that relates descriptors in 3D-object query to descriptors previously seen in the indexing step. The k-means algorithm is chosen for clustering. In order to determine the parameter k, the k-means method is implemented several times with different number of desired k, and then the final clustering giving the lowest empirical risk is selected.

3) Shape histogram computing: Descriptors in the 3D object are assigned to the nearest neighbor keyshapes in the vocabulary. Then each object is represented using an histogram whose i^{th} bin contains the number of i^{th} keyshapes in that object.

4) Shape matching: Compare two objects, treating their bag of keyshapes as feature

Table 1: Retrieval performance of 11 methods evaluated using five standard measures on the whole database.

Authors	Methods	NN	FT	ST	E	DCG
Kawamura	FOG	96.8%	81.7%	90.3%	66.0%	94.4%
Lavoue	BOW-LSD	95.5%	67.2%	80.3%	57.9%	89.7%
Lian	MDS-CM-BOF	99.5%	91.3%	96.9%	71.7%	98.2%
Nguyen	BOGH	99.3%	81.1%	88.4%	64.7%	94.9%
Ohkita	LSF	99.5%	79.9%	86.3%	63.3%	94.3%
Reuter	ShapeDNA	99.2%	91.5%	95.7%	70.5%	97.8%
Sipiran	Harris3DGeoMap	56.2%	32.5%	46.6%	32.2%	65.4%
Sipiran	HKS	83.7%	40.6%	49.7%	35.3%	73.0%
Smeets	MeshSIFT	99.5%	88.4%	96.2%	70.8%	98.0%
Smeets	SD-GDM	100.0%	96.2%	98.4%	73.1%	99.4%
Tabia	PatchBOF	74.8%	64.2%	83.3%	58.8%	83.7%

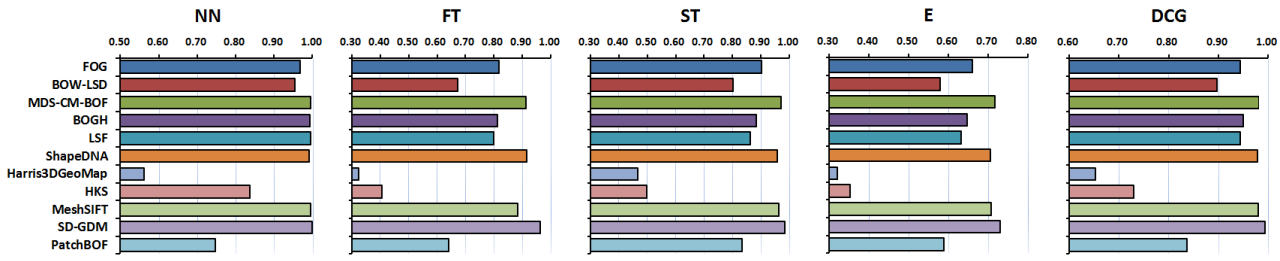

Figure 14: Bar charts of the retrieval accuracies of 11 methods evaluated on the whole database using five standard measures, respectively.

vectors, and thus determine their dissimilarity by calculating L_2 difference between two histograms.

7. Results

In this section, we present and compare the results of the above-mentioned 11 methods. Given the 11 dissimilarity matrices, evaluations for these approaches are carried out not only on the average performance of the whole database, but also on the result corresponding to each specific class. We evaluate the retrieval performance by using the five quantitative statistics *(i.e., NN, FT, ST, E, and DCG)* and the *Precision-recall curve* described in Section 4.

Table 1 shows the retrieval accuracies of all 11 algorithms evaluated on the whole database. We observe that most of these methods perform well in this benchmark. For instance, DCG values of 7 methods are greater than 0.940 and 8 methods have NN values that are above 0.950. In Fig. 14, we also provide bar charts to intuitively compare the results of those methods evaluated using five quantitative measures, respectively. As we can see from Table 1 and

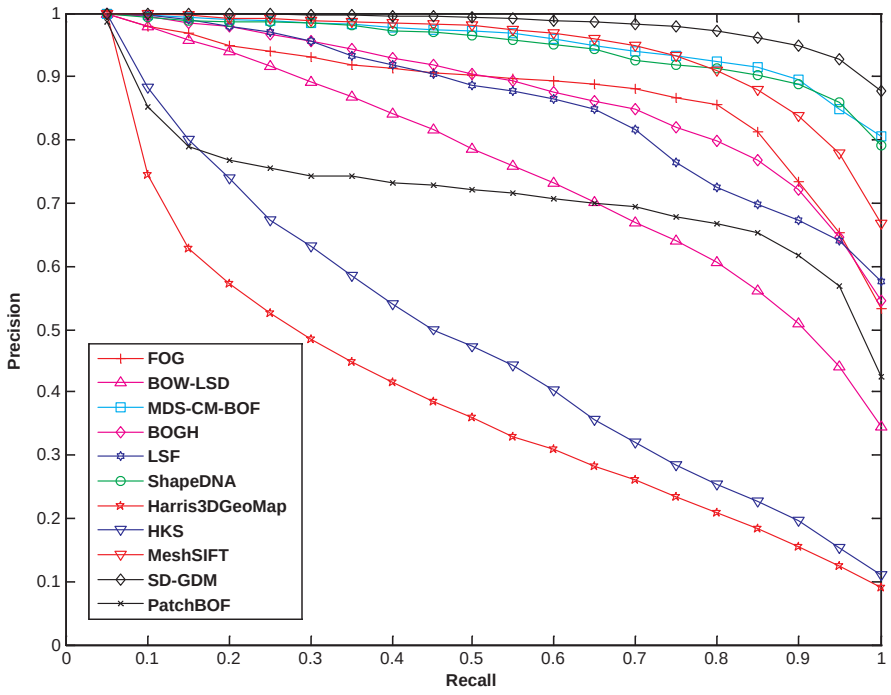

Figure 15: Precision-recall curves of 11 methods evaluated for the whole database.

Fig. 14, Smeets's SD-GDM clearly outperforms all other algorithms, while the second and third best methods are not so obvious. Considering the values of FT, Reuter's ShapeDNA method gets better performance than Lian's MDS-CM-BOF, but if we base the evaluation on NN, ST, E, and DCG, Lian's MDS-CM-BOF would take the second place. Similar observations can be made from Fig. 15, which shows Precision-recall curves of all these algorithms on the whole database.

Next, we show the Precision-recall curves of these 11 approaches evaluated for selected 12 classes of the database in Fig. 16. We find that none of these methods performs best for all kinds of objects. For example, Smeets's SD-GDM obtains the best results for lots of categories but not *ant, bird2, paper, pliers, spider, woman* models, etc., while although Tabia's PatchBOF performs worst in the retrieval of *woman* models, it outperforms others for *lamp* objects. As shown in Fig. 2, our database contains a set of models which have similar overall appearances but belong to various categories because they are different in the details of local regions or/and topological structures. This makes the new benchmark more challenging

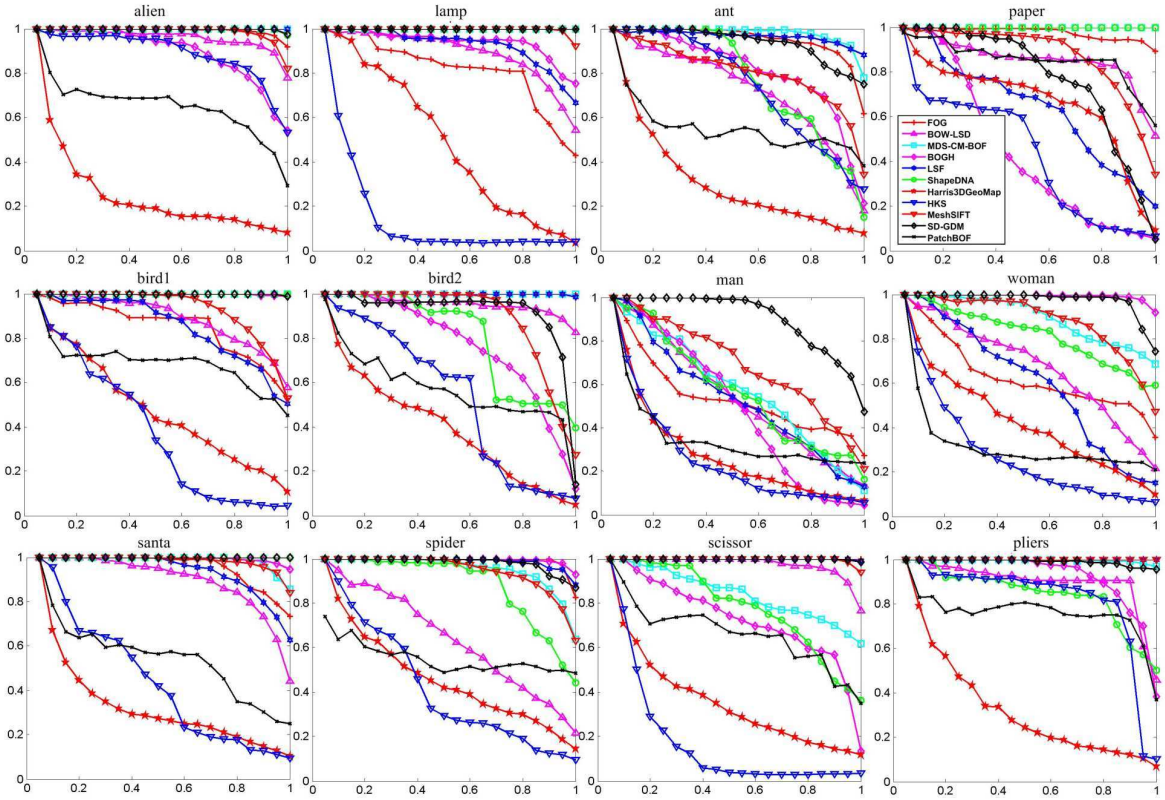

Figure 16: Precision-recall curves of 11 methods evaluated for 12 different classes, respectively.

than other non-rigid 3D databases. However, as we can see from Fig. 16, the challenge can be well resolved by several algorithms described in this paper. For example, Lian's MDS-CM-BOF are able to perfectly discriminate two types of bird models (i.e., *bird1* and *bird2*), which have slightly different skeletons, while Smeets's SD-GDM obtains considerably high retrieval accuracies for the bird models as well as the human models (i.e., *man* and *woman*) that possess dissimilar features based on gender. Generally speaking, most of these methods (e.g., Smeets's SD-GDM, Lian's MDS-CM-BOF, Reuter's ShapeDNA, etc.) work fairly well for every class in this database, as their precision-recall curves are all in the top right parts of these figures.

Analyzing the 11 methods compared in this paper, we find that the most popular approach (5 methods, i.e., FOG, BOW-LSD, MDS-CM-BOF, LSF, and PatchBOF) is to employ the bag-of-features method to quantize a model's local features into a word histogram. There also exist methods (2 runs including Smeets's MeshSIFT and Sipiran's HKS) that extract

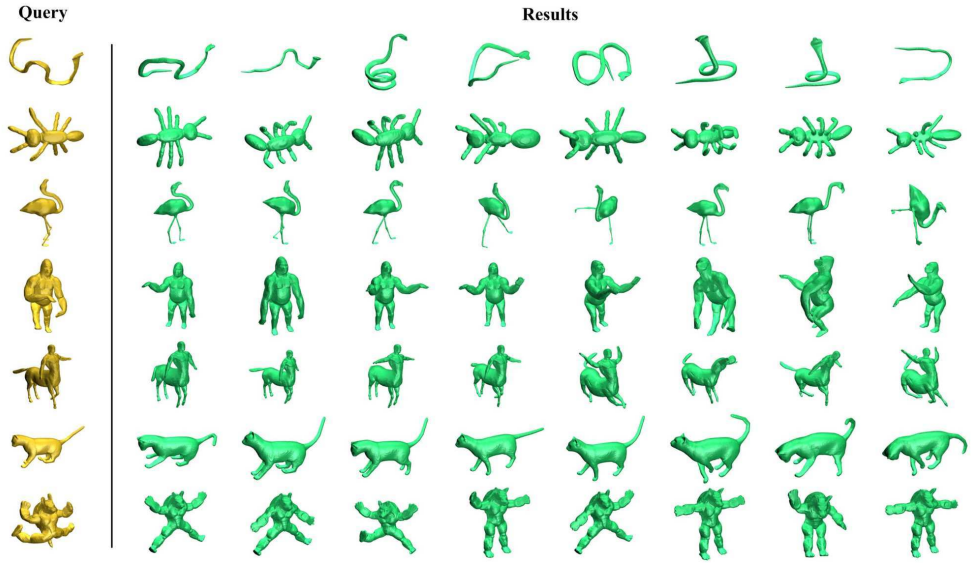

Figure 17: Examples of queries (first column) from our database and the corresponding top 8 retrieved models using Lian's MDS-CM-BOF method. The retrieved models are ranked from left to right based on the increasing order of dissimilarity.

salient local features and match them directly to compare 3D shapes. Lian's MDS-CM-BOF is basically a visual similarity based method. Mainly because of the utilization of both 3D Canonical Forms and local features, it becomes insensitive against isometric transformations. Other 4 methods (including BOGH, ShapeDNA, Harris3DGeoMap, and SD-GDM) utilize specifically-designed shape descriptors that are based on various isometry-invariant global properties of 3D models. We also observed that(see [16]), the combination of several different kinds of methods can result in better retrieval accuracies, and it is possible to further improve performance by applying some unsupervised Machine Learning algorithms (e.g., Manifold Ranking).

Fig. 17 shows some queries and their corresponding top 8 retrieved objects from this database using Lian's MDS-CM-BOF algorithm. As we can see from this figure, the retrieved 3D models in the top 8 positions of the rank lists all belong to the same categories of their corresponding queries, which again verifies the effectiveness of the evaluated method in non-rigid 3D shape retrieval applications. For more results, we refer the reader to our web site [1], where the new non-rigid 3D shape benchmark and the evaluation code are also freely-available for academic use.

8. Conclusion

In this paper, we first proposed an efficient approach to generate a large amount of non-rigid 3D watertight meshes, based on which a new non-rigid 3D shape benchmark was developed. We then presented a number of non-rigid 3D shape retrieval algorithms and compared their performance by carrying out experiments on the new benchmark. Our results demonstrated that SD-GDM, MDS-CM-BOF, and ShapeDNA are the three most discriminative methods among the 11 approaches we evaluated, but none of these 11 methods performs best for all kinds of objects.

This paper is based on the *SHREC'11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes* which is the second attempt in the history of SHREC to specifically focus on the performance evaluation of non-rigid 3D shape retrieval algorithms. Compared to the first SHREC non-rigid 3D shape retrieval track [64] (200 models and 3 groups) we organized in 2010, both the size of the database (600 models) and the number of participants (9 groups) tripled in 2011, which indicates that more and more researchers have become interested in analyzing non-rigid 3D shapes. We believe that, with so many participants taking part in this track, methods described in this paper constitute a good representation of the state of the art in this important research field, and we hope that the new benchmark would further promote the investigation of non-rigid 3D shape retrieval.

Several directions of future work are listed as follows: 1) Keep updating our benchmark by using the proposed method to generate more models for the database; 2) Build non-rigid 3D shape benchmarks for some specific applications (e.g., the retrieval of proteins, faces, animals, etc.); 3) Select different kinds of approaches carefully and integrate them optimally to develop more discriminative and efficient non-rigid 3D shape retrieval systems.

Disclaimer

Any mention of commercial products or reference to commercial organizations is for information only; it does not imply recommendation or endorsement by NIST nor does it imply that the products mentioned are necessarily the best available for the purpose.

ACKNOWLEDGMENTS

This work has been supported by China Postdoctoral Science Foundation (Grant No.: 2012M510274), the SIMA program and the Shape Metrology IMS. We would like to thank the anonymous reviewers for their constructive comments, and AIM@SHAPE, Cyberware, Kaleem Siddiqi, Philip Shilane, Michael Bronstein, Robert Sumner, and Daniela Giorgi for providing original 3D models.

References

[1] Http://www.itl.nist.gov/iad/vug/sharp/contest/2011/NonRigid/ (2011).

[2] D. Zhang, G. Lu, Review of shape representation and description techniques, Pattern Recognition 37 (1) (2004) 1–19.

[3] P. Papadakis, I. Pratikakis, T. T. S Perantonis, Efficient 3d shape matching and retrieval using a concrete radialized spherical projection representation, Pattern Recognition 40 (9) (2007) 2437–2452.

[4] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, ACM Transactions on Graphics (TOG) 21 (4) (2002) 807–832.

[5] R. Gal, A. Shamir, D. Cohen-Or, Pose-oblivious shape signature, IEEE Transactions on Visualization and Computer Graphics (TVCG) 13 (2) (2007) 261–271.

[6] Z. Lian, P. L. Rosin, X. Sun, Rectilinearity of 3D meshes, International Journal of Computer Vision (IJCV) 89 (2-3) (2010) 130–151.

[7] M. R. Ruggeri, G. patane, M. Spagnuolo, D. Saupe, Spectral-driven isometry-invariant matching of 3D shapes, International Journal of Computer Vision (IJCV) 89 (2-3) (2010) 248–265.

[8] J. W. Tangelder, R. C. Veltkamp, A survey of content based 3D shape retrieval methods, Multimedia Tools and Applications 39 (3) (2008) 441–471.

[9] F. Mémoli, G. Sapiro, A theoretical and computational framework for isometry invariant recognition of point cloud data, Foundations of Computational Mathematics 5 (3) (2005) 313–347.

[10] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The princeton shape benchmark, in: Proc. Shape Modeling International (SMI'04), 2004, pp. 167–178.

[11] D. Smeets, T. Fabry, J. Hermans, D. Vandermeulen, P. Suetens, Isometric Deformation Modelling for Object Recognition, in: Proc. The 13th International Conference on Computer Analysis of Images and Patterns (CAIP'09), 2009, pp. 757–765.

[12] A. M. Bronstein, M. M. Bronstein, R. Kimmel, Numerical geometry of non-rigid shapes, Springer, 2008.

[13] J. Sun, M. Ovsjanikov, L. J. Guibas, A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion, Comput. Graph. Forum 28 (5) (2009) 1383–1392.

[14] M. Reuter, F. E. Wolter, N. Peinecke, Laplace-Beltrami spectra as shape-DNA of surfaces and solids, Computer-Aided Design 38 (4) (2006) 342–366.

[15] K. Siddiqi, J. Zhang, D. Maxrini, A. Shokoufandeh, S. Bouix, S. Dickinson, Retrieving articulated 3d models using medial surfaces, Machine Vision and Applications 19 (4) (2008) 261–274.

[16] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoue, H. V. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, D. Vandermeulen, SHREC'11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes, in: Proc. Eurographics Workshop on 3D Object Retrieval (3DOR'11), 2011, pp. 79–88.

[17] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3D shape descriptors, in: Proc. Symposium on Geometry Processing (SGP'03), 2003, pp. 156–164.

[18] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, M. Ouhyoung, On visual similarity based 3D model retrieval, in: Proc. Eurographics 2003, Vol. 22, 2003, pp. 223–232.

[19] J. Shih, C. Hsing, J. Wang, A new 3D model retrieval approach based on the elevation descriptor, Pattern Recognition 40 (1) (2007) 283–295.

[20] A. Mademlis, P. Daras, D. Tzovaras, M. G. Strintzis, 3d object retrieval using the 3d shape impact descriptor, Pattern Recognition 42 (11) (2009) 2447–2459.

[21] A. E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3d scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 21 (5) (1999) 433–449.

[22] Y. Liu, H. Zha, H. Qin, Shape topics: A compact representation and new algorithms for 3D partial shape retrieval, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006, pp. 2025–2032.

[23] M. Ovsjanikov, A. M. Bronstein, M. M. Bronstein, L. Guibas, Shape google: a computer vision approach to isometry invariant shape retrieval, in: Proc. NORDIA'09, 2009, pp. 320–327.

[24] R. Ohbuchi, K. Osada, T. Furuya, T. Banno, Salient local visual features for shape-based 3D model retrieval, in: Proc. Shape Modeling International (SMI'08), 2008, pp. 93–102.

[25] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision (IJCV) 60 (2) (2004) 91–110.

[26] X. Wang, Y. Liu, H. Zha, Intrinsic spin images: A subspace decomposition approach to understanding 3d deformable shapes, in: Proc. International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'10), 2010, pp. 17–20.

[27] M. Hilaga, Y. Shinagawa, T. Kohmura, T. L. Kunii, Topology matching for fully automatic similarity estimation of 3d shapes, in: Proc. SIGGRAPH, 2001, pp. 203–212.

[28] H. Sundar, D. Silver, N. Gagvani, S. Dickinson, Skeleton based shape matching and retrieval, in: Proc. Shape Modeling International (SMI'03), 2003, pp. 130–139.

[29] G. Tam, R. Lau, Deformable model retrieval based on topological and geometric signatures, IEEE Transactions on Visualization and Computer Graphics (TVCG) 13 (3) (2007) 470–482.

[30] M. Reuter, F. E. Wolter, N. Peinecke, Laplace-spectra as fingerprints for shape matching, in: Proc. SPM'05, 2005, pp. 101–106.

[31] V. Jain, H. Zhang, A spectral approach to shape-based retrieval of articulated 3D models, Computer Aided Design 39 (5) (2007) 398–407.

[32] M. Mahmoudi, G. Sapiro, Three-dimensional point cloud recognition via distributions of geometric distances, Graphical Models 71 (1) (2009) 22–31.

[33] F. Mémoli, On the use of gromov-hausdorff distances for shape comparison, in: Proc. SPG'07, 2007, pp. 81–90.

[34] A. M. Bronstein, M. M. Bronstein, R. Kimmel, Efficient computation of isometry-invariant distances between surfaces, SIAM Journal on Scientific Computing 28 (5) (2006) 1812–1836.

[35] A. Elad, R. Kimmel, On bending invariant signatures for surface, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 25 (10) (2003) 1285–1295.

[36] Z. Lian, A. Godil, X. Sun, H. Zhang, Non-rigid 3D shape retrieval using multidimensional scaling and bag-of-features, in: Proc. International Conference on Image Processing (ICIP 2010), 2010, pp. 3181–3184.

[37] http://www.aimatshape.net/event/SHREC/ (SHape REtrieval Contest (SHREC)).

[38] R. Fang, A. Godil, X. Li, A. Wagan, A new shape benchmark for 3D object retrieval, in: Proc. International Symposium on Visual Computing (ISVC'08), 2008, pp. 381–392.

[39] S. Jayanti, Y. Kalyanaraman, N. Iyer, K. Ramani, Developing an engineering shape benchmark for CAD models, Computer-Aided Design 38 (9) (2006) 939–953.

[40] D. Giorgi, S. Biasotti, L. Paraboschi, SHape REtrieval Contest 2007: watertight models track, in: SHREC competition, 2007.

[41] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: Proc. Neural Information Processing Systems Foundation (NIPS'03), 2003.

[42] G. Lavoué, Bag of words and local spectral descriptor for 3d partial shape retrieval, in: Proc. Eurographics Workshop on 3D Object Retrieval (3DOR'11), 2011, pp. 41–48.

[43] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Information Theory 28 (2) (1982) 129–137.

[44] B. Lévy, H. Zhang, Spectral mesh processing, Siggraph 2010 Course.

[45] B. Vallet, B. Lévy, Spectral geometry processing with manifold harmonics, Computer Graphics Forum 27 (2) (2008) 251–260.

[46] Z. Lian, A. Godil, X. Sun, Visual similarity based 3D shape retrieval using bag-of-features, in: Proc. Shape Modeling International (SMI'10), 2010, pp. 25–36.

[47] Y. Ohkita, T. Furuya, R. Ohbuchi, Sets of local 3d shape descriptors for 3d model retrieval, in: Proc. Visual Computing Symposium 2009 (in Japanese), 2009.

[48] E. Wahl, U. Hillenbrand, G. Hirzinger, Surflet-Pair-Relation Histograms: A Statistical 3D-Shape Representation for Rapid Classification, in: Proc. International Conference on 3D Digital Imaging and Modeling (3DIM'03), 2003, pp. 474–481.

[49] M. Reuter, F. E. Wolter, M. Shenton, M. Niethammer, Laplace-Beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis, Computer-Aided Design 41 (10) (2009) 739–755.

[50] M. Reuter, Hierarchical shape segmentation and registration via topological features of Laplace-Beltrami eigenfunctions, International Journal of Computer Vision (IJCV) 89 (2) (2010) 287–308.

[51] M. H. Protter, Can one hear the shape of a drum? revisited, SIAM Review 29 (2) (1987) 185–197.

[52] M. Kac, Can one hear the shape of a drum?, American Mathematical Monthly 73 (4) (1966) 1–23.

[53] H. Weyl, Über die asymptotische Verteilung der Eigenwerte, Nachrichten von der (königlichen) Gesellschaft der Wissenschaften zu Göttingen (1911) 110–117.

[54] C. Gordon, D. Webb, S. Wolpert, Isospectral plane domains and surfaces via riemannian orbifolds, Inventiones Mathematicae 110 (1992) 1–22.

[55] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, M. Spagnuolo, Discrete Laplace-Beltrami operators for shape analysis and segmentation, Computers & Graphics 33 (3) (2009) 381–390.

[56] Http://reuter.mit.edu/software (2011).

[57] I. Sipiran, B. Bustos, A robust 3D interest points detector based on Harris operator, in: Proc. Eurographics Workshop on 3D Object Retrieval (3DOR'10), 2010, pp. 7–14.

[58] D. Smeets, T. Fabry, J. Hermans, D. Vandermeulen, P. Suetens, Isometric deformation modeling using singular value decomposition for 3D expression-invariant face recognition, in: Proc. IEEE Fourth International Conference on Biometrics: Theory, Applications and Systems (BTAS'09), Washington DC, USA, 2009.

[59] G. Peyré, L. D. Cohen, Heuristically Driven Front Propagation for Fast Geodesic Extraction, IJVCB 1 (1) (2009) 55–67.

[60] G. Peyré, Toolbox mast marching, MATLAB Central File Exchange Select (2009).

[61] C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens, D. Vandermeulen, Feature detection on 3D face surfaces for pose normalisation and recognition, in: Proc. IEEE Fourth International Conference on Biometrics: Theory, Applications and Systems (BTAS'10), 2010.

[62] G. Peyré, Toolbox graph, MATLAB Central File Exchange Select (2009).

[63] H. Tabia, M. Daoudi, J. P. Vandeborreb, O. Colot, Deformable Shape Retrieval Using Bag-of-Feautre Techniques, in: Proc. 3D Image Processing (3DIP'11), 2011.

[64] Z. Lian, A. Godil, T. Fabry, T. Furuya, J. Hermans, R. Ohbuchi, C. Shu, D. Smeets, P. Suetens, D. Vandermeulen, S. Wuhrer, SHREC'10 Track: Non-rigid 3D Shape Retrieval, in: Proc. Eurographics Workshop on 3D Object Retrieval (3DOR'10), 2010, pp. 101–108.