Deformable Shape Retrieval using Bag-of-Feature techniques

Hedi Tabia[a] Mohamed Daoudi[b] Jean-Philippe Vandeborre[b] and Olivier Colot[a]

[a]LAGIS FRE CNRS 3303, University of Lille 1, Villeneuve d'Ascq, France
[b]LIFL UMR CNRS 8022, Institut TELECOM, TELECOM Lille 1, Villeneuve d'Ascq, France

ABSTRACT

We present a novel method for 3D-shape matching using Bag-of-Feature techniques (BoF). The method starts by selecting and then describing a set of points from the 3D-object. Such descriptors have the advantage of being invariant to different transformations that a shape can undergo. Based on vector quantization, we cluster those descriptors to form a shape vocabulary. Then, each point selected in the object is associated to a cluster (word) in that vocabulary. Finally, a BoF histogram counting the occurrences of every word is computed. These results clearly demonstrate that the method is robust to non-rigid and deformable shapes, in which the class of transformations may be very wide due to the capability of such shapes to bend and assume different forms.

Keywords: 3D Shape retrieval, feature extraction, bag of feature.

1. INTRODUCTION

The 3D-object matching has emerged as an important area in computer vision and multimedia computing. Many organizations have large 3D-collections in digital format, available for on-line access. Organizing these libraries into categories and providing effective indexing is imperative for real time browsing and retrieval.

In recent years, a large number of papers are concerned in 3D-shape matching.[1] Most of the current 3D-shape matching approaches are based on global similarity. Some methods directly analyze the 3D-mesh — such as Antini et al.[2] who use curvature correlograms — and some others use a 2D-view based approach — such as Filali et al.[3] who propose an adaptive nearest neighbor-like framework to choose the characteristic views of a 3D-model. These approaches enable retrieval of similar objects in spite of rigid transformations. Other approaches are also enabling retrieval of objects that differ from non-rigid transformations such as shape bending or character articulation.[4] They are generally based on the graph or skeleton computation of a 3D-object.[5] Other case of shape matching is encountered when partially similar shapes are presented. This kind of shape matching plays a crucial role in many applications such as indexing or modeling by example. It is usually approached using the *recognition by part* idea:[6,7] segmentation of the shape in significant parts, and matching pairs of parts as whole shapes. Bronstein et al.[8] also proposed to consider the 3D-matching problem as a multi-criterion optimization problem trying to simultaneously maximize the similarity and the significance of the matching parts.

Bag-of-Features (BoF), which is a popular approach in areas of computer vision and pattern recognition, have recently gained great popularity in shape analysis community. Liu et al.[7] presented a 3D-shape descriptor named "Shape Topics" and applied it to 3D partial shape retrieval. In their method, a 3D-object is considered as a word histogram obtained by vector quantizing Spin images of the object. Ohbuchi et al.[9] introduced a view-based method using salient local features. They represented 3D-objects as word histograms derived from the vector quantization of salient local descriptors extracted on the depth-buffer views captured uniformly around the objects. Ovsjanikov et al.[10] presented an approach to non-rigid shape retrieval similar in its spirit to text retrieval methods used in search engines. They used the heat kernel signatures to construct shape descriptors that are invariant to non-rigid transformations.

Inspired by the work presented in[10] and with analogy to methods using the bag-of-words representation for text retrieval.[11] In this paper, we propose a new visual similarity based 3D shape retrieval approach. An overview of the method is detailed in section 2, Section 3 presents the experimental results. Conclusion and future work end the paper.

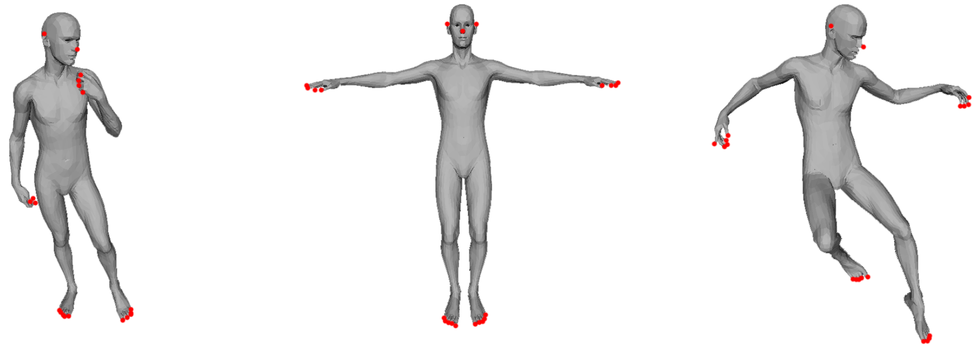


Figure 1. Feature points extracted from different poses of 3D-model.

2. METHOD OVERVIEW

Our method consists of four main steps. 1) Detection and description of 3D-object patches, 2) Assigning patch descriptors to a set of predetermined clusters (a vocabulary) with a vector quantization algorithm, 3) Constructing a bag of keyshapes, which counts the number of patches assigned to each cluster and 4) Matching between two objects, treating their bag of keyshapes as the feature vector, and thus determine their dissimilarity.

Ideally these steps are designed to maximize matching accuracy while minimizing computational effort. Thus, the descriptors extracted in the first step should be invariant to variations that are irrelevant to the retrieval task (deformable shapes) but rich enough to carry enough information to be discriminative between dissimilar objects. The vocabulary used in the second step should be large enough to distinguish relevant changes in 3D-object patches, but not too large as to distinguish irrelevant variations such as noise. We refer to the quantized feature vectors (cluster centers) as “keyshape” by analogy with “keywords” in text retrieval. However, in our case “words” do not necessarily have a repeatable meaning such as “legs”, or “head”, nor is there an obvious best choice of vocabulary. Rather, our goal is to use a vocabulary that allows good matching performance. We now discuss the choices made for each step in more detail.

2.1 Feature extraction

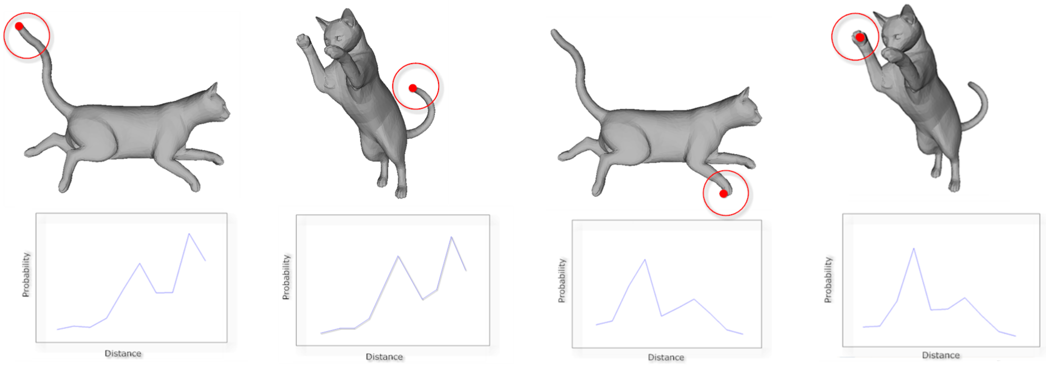
In this section, we detail the two major steps of feature extraction: 3D patches detection and 3D patches description.

2.1.1 3D patches detection

Here, we present the algorithm developed for the feature extraction process. Using a diversity of 3D-objects, the proposed algorithm produces robust and well-localized feature points. The concept of feature points has been introduced by several authors.[6, 12] However it is difficult to find a formal definition that characterizes this concept. In Katz et al.[12] feature points refer to the points localized at the extremity of a 3D-object’s components. Formally, feature points are defined as the set of points that are the furthest away (in the geodesic sense) from all the other points of the surface. They are equivalent to the local extrema of a geodesic distance function which is defined on the 3D-object. Figure 1 shows some 3D-objects with their corresponding feature points.

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.

As mentioned by Cole-McLaughlin et al.,[13] 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. Reciprocally, 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. We define the set of feature points F of the



(a) (b) (c) (d)

Figure 2. Distance-distributions on four different feature points.

triangulated surface S as the closest *intersecting* points in the sets F_1 and F_2. In practice, f_1 and f_2 local extrema do not appear exactly on the same vertices but in the same geodesic neighborhood. Consequently, we define the intersection operator $\cap$ with the following set of constraints, where d_n stands for the normalized geodesic distance function (to impose scale invariance):

$$V \in F = F_1 \cap F_2 \Leftrightarrow \begin{cases} \exists v_{F_1} \in F_1 \ / \ d_n\left(V, v_{F_1}\right) < \epsilon \\ \exists v_{F_2} \in F_2 \ / \ d_n\left(V, v_{F_2}\right) < \epsilon \\ d_n\left(V, v_i\right) > \epsilon \ \forall v_i \in F \\ \epsilon, d_n \in [0,1] \end{cases}$$

This algorithm detects all the feature points required in the subsequent analysis. They are accurately localized and their localization is robust with respect to rigid and non-rigid transformations, because of the use of geodesic distance in f_1 and f_2 functions.

2.1.2 3D patches description

Once feature points extraction process is performed, we get a set F of all the feature points. Then, we proceed to construct a descriptor that can be representative to each feature point. In our system, this descriptor is a probability distribution that represents the shape of the 3D-object (as other similar approaches for 3D-object recognition[14, 15]). The distribution is sampled from an intrinsic distance function on the 3D-patch surface. The distance function used in our system is based on the geodesic distances between the feature point and all the points on the 3D-object surface. It is in turn invariant to the rigid and non-rigid transformations of the surface. The descriptor constructed with a feature point as starting point of a distribution is called *a 3D-patch*. Figure 2 shows the descriptor corresponding to four feature points extracted from two different cats. We can notice that the tail-patch descriptor Figure 2(a) of the first cat is similar to the tail-patch descriptor Figure 2(b) of the second cat. The leg-patches of the two cats (Figure 2(c) and Figure 2(d)) also have similar distance distributions.

More formally, given a 3D-object O, for every feature point $F_i \in F$, we define a descriptor $P(F_i)$ associated to F_i and consider the geodesic distances $\{d(F_i, v); \forall v \in V\}$ with V is the set of all the vertices on the 3D-object surface. Considering f the distribution of vertices according to these distances, we define the descriptor $P(F_i)$ as an R-dimensional vector:

$$P(F_i) = (p_1, ..., p_R)$$

where

$$p_r = \int_{(r-1)/R}^{r/R} f(d) \delta d$$

$P(F_i)$ is an R-bin histogram of vertex distribution of geodesic distances measured from F_i. In order to make the descriptors comparable between different shapes, we have to scale the geodesic function d by the geodesic diameter of the shape.

2.2 Shape vocabulary construction

The vocabulary used in our method is a way of constructing a feature vector for classification that relates descriptors in 3D-object to be classified to descriptors previously seen in training. One extreme of this approach would be to compare each object descriptor to be classified to all training descriptors: with the huge number of training descriptors involved, this seems impractical. Another extreme would be to try to identify a small number of large "clusters" which sufficiently discriminate between different shape classes. In practice we find that the best trade-offs of accuracy and computational efficiency are obtained for intermediate sizes of clustering. Vector quantization algorithms are generally based on iterative square-error partitioning or on hierarchical techniques. Square-error partitioning algorithms attempt to obtain the partition which minimizes the within-cluster distance or maximizes the between-cluster distance. Hierarchical techniques organize data in a nested sequence of groups which can be displayed in the form of a tree. They need some heuristics to form clusters and hence are less frequently used than square-error partitioning techniques in pattern recognition. We chose to use the simplest square-error partitioning method: k-means.[16] This algorithm proceeds by iterated assignments of points to their closest cluster centers and re-computation of the cluster centers. Two difficulties are that the k-means algorithm converges only to local optima of the squared distortion, and that it does not determine the parameter k. There exist methods allowing to automatically estimate the number of clusters. For example, Pelleg et al[17] use cluster splitting to do it, where the splitting decision is done by computing the Bayesian Information Criterion. However, in the present case we do not really know anything about the density or the compactness of our clusters. Moreover, we are not even interested in a "correct clustering" in the sense of feature distributions, but rather in accurate categorization. We therefore run k-means several times with different number of desired representative vectors (k) and different sets of initial cluster centers. We select the final clustering giving the lowest empirical risk in categorization.[18]

2.3 Shape histogram computing

The final step of the BoF method is the computing of the dissimilarity between two objects. To this end, we compute the cumulative occurrences of keyshapes in both objects. Descriptors in the 3D-objects 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. Finally, the dissimilarity between two objects can be computed using several metrics, as D_{Max}, D_{Min}, ... L_2. We use the L_2 to measure the distance between two objects.

3. EXPERIMENTS

In this section, we present the experimental results, some state-of-the-art shape-matching algorithms to compare with. The robustness to shape deformations is discussed in the sequel. The algorithms that we have described in the previous sections have been implemented using Matlab software. The framework encloses an off-line feature extraction algorithm and a vector quantization algorithm, and an on-line retrieval process.

The proposed approach has been tested on tow databases the first is the SHREC07 dataset and the second is composed of shapes from the TOSCA and the Sumner datasets.

The SHREC07 dataset[19] contains 400 3D-objects in 20 classes. It is a challenging dataset, not only because of the large number of classes, but also because it contains shapes with highly variable poses and non-rigid or isometric transformations.

The TOSCA dataset has been proposed by Bronstein et al.[20] for non-rigid shape correspondence measures. The Sumner dataset has been proposed by Sumner and Popovic[21] for deformable shape correspondence. The total set size of the second dataset is 450 shapes.

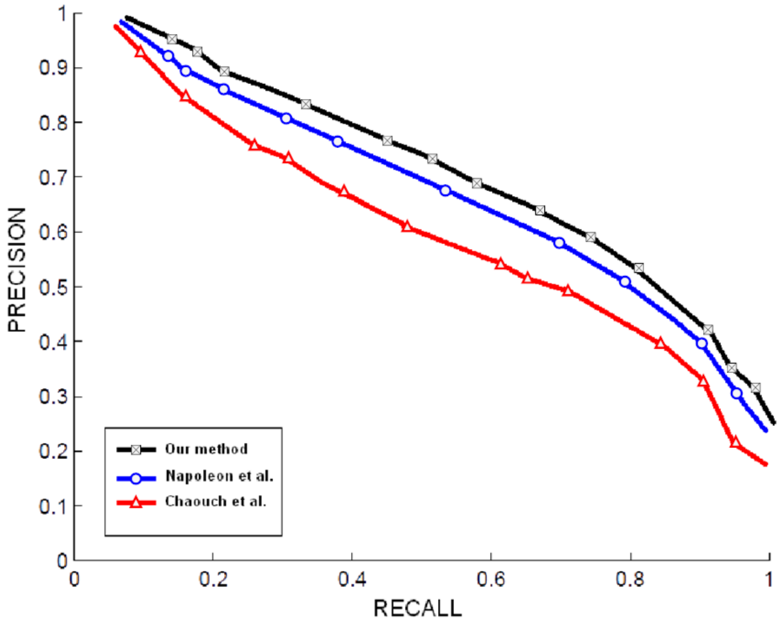


Figure 3. Precision vs Recall plots for our method and some stat of art methods On the Shrec07 dataset.

We test the robustness of our approach using the Precision vs Recall plots. They are well known in the literature of content-based search and retrieval. The precision and recall are defined as follow:

$$Precision = \frac{N}{A}, Recall = \frac{N}{Q}$$

where N is the number of relevant models retrieved in the top A retrievals, Q is the number of relevant models in the collection, which is the number of models to which the query belongs to.

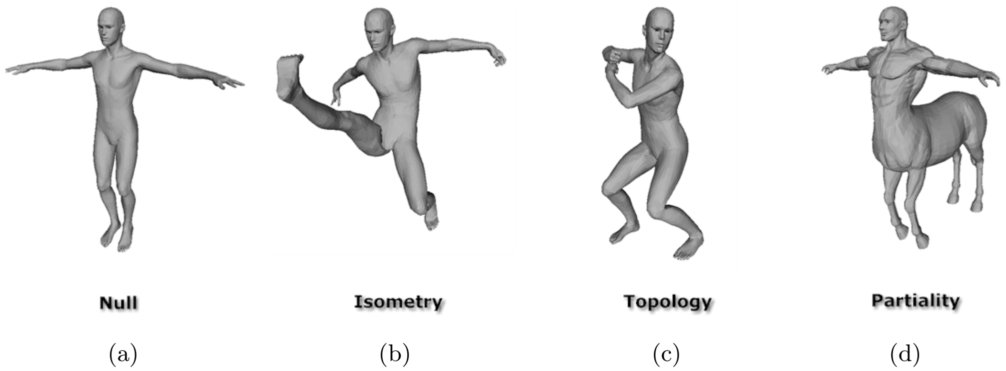
3.1 Comparison with related work

From a quantitative point of view, we compare the Precision vs Recall plot of our approach with some well-known stat-of-art methods on the SHREC07 dataset.[22]

- Chaouch et al. method:[23] the 3D shape retrieval system proposed in[23] compares 3D models based on their visual similarity using depth lines extracted from depth images. The process first normalizes and scales 3D model into a bounding box. Then, it computes the set of $N \times N$ depth-buffer images associated to the six faces of the bounding box. The system then generates $2 \times N$ depth lines per image, considering each depth image as a collection of N horizontal and N vertical depth lines. Finally, each depth line is encoded in a set of N states called sequence of observations. The shape descriptor consists in the set of $6 \times 2 \times N$ sequences, with $N = 32$.

- Napoleon et al. method:[22] their method is based on silhouettes intersection. First, the system captures a set of views of an object and extracts its silhouette from each view. The distance between two silhouettes is chosen as equal to the number of pixels that are not common to the two silhouettes intersection. The distance between two objects is just defined as the sum of the distance between their two sets of silhouettes.

Figure 3 shows the Precision vs Recall plots. Such a plot is the average of the set of Precision vs Recall plots corresponding to the models of the query-set. As the curve of our approach is higher than the others, it is obvious that it outperforms related methods.



(a) (b) (c) (d)

Figure 4. Examples of some transformations of 3D-object

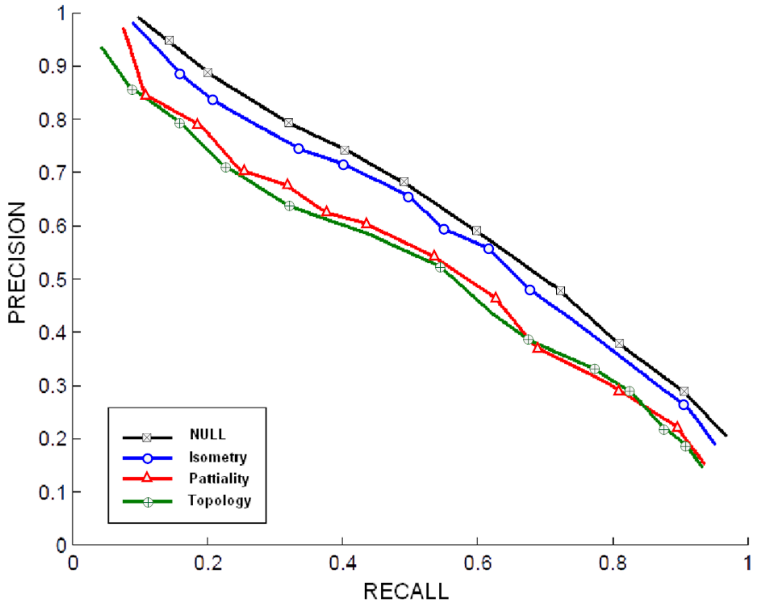


Figure 5. Precision vs Recall plots for some classes of shape transformations on the Tosca and Sumner dataset.

3.2 Robustness under shape deformations

In this section, we discuss the robustness of the retrieval method under deformed instances of the query shapes. For this end we conduct similar experiments as in.[10] First, we choose shapes in one of the 4 transformation categories (Null, isometry, topology and partiality shown in Figure 4) as a query set, and we kept the remaining transformation with the objects in the dataset. The results of this experiment are shown in Figure 5 as Precision vs Recall plots. The method is robust under isometry and partiality deformations, however it is less robust to topological ones. This is due to the feature point extraction step of the framework, which is based on geodesics. In practice, with the TOSCA and the Sumner datasets, feature extraction turns out to be homogeneous within most classes. However, the geodesic metric is known to be very sensitive to topology which make the overall framework quite sensible to topology variations also.

4. CONCLUSION

In this paper, we have presented a novel method for retrieving 3D-objects based on the BoF techniques. We have proposed a new feature extraction algorithm. We have presented results for retrieving 3D-objects from 3 datasets the shrec07, the Tosca and the Sumner datasets. These results show the effectiveness of our approach and clearly demonstrate that the method is robust to non-rigid and deformable shapes.

REFERENCES

1. J. W.-H. Tangelder and R.-C. Veltkamp, “A survey of content based 3d shape retrieval methods,” *Multimedia Tools and Applications* **39**, pp. 441–471, September 2008.
2. G. Antini, S. Berretti, A. Del Bimbo, and P. Pala, “Retrieval of 3d objects using curvature correlograms,” in *IEEE International Conference on Multimedia & Expo*, July 2005.
3. T. Filali Ansary, M. Daoudi, and J.-P. Vandeborre, “A bayesian 3D search engine using adaptive views clustering,” *IEEE Transactions on Multimedia* **9**, pp. 78–88, January 2007.
4. R. Gal, A. Shamir, and D. Cohen-Or, “Pose oblivious shape signature,” *IEEE Transactions on Visualization and Computer Graphics* **13**, pp. 261–271, March-April 2007.
5. M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii, “Topology matching for fully automatic similarity estimation of 3D shapes,” in *SIGGRAPH*, pp. 203–212, 2001.
6. J. Tierny, J.-P. Vandeborre, and M. Daoudi, “Partial 3D shape retrieval by reeb pattern unfolding,” *Computer Graphics Forum - Eurographics Association* **28**, pp. 41–55, March 2009.
7. Y. Liu, H. Zha, and H. Qin, “Shape topics: A compact representation and new algorithms for 3d partial shape retrieval,” in *Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.
8. A. Bronstein, M. Bronstein, A. Bruckstein, and R. Kimmel, “Partial similarity of objects, or how to compare a centaur to a horse,” *International Journal of Computer Vision* , July 2008.
9. R. Ohbuchi, K. Osada, T. Furuya, and T. Banno, “Salient local visual features for shape-based 3d model retrieval,” in *International Conference on Shape Modeling and Applications (SMI)*, 2008.
10. M. Ovsjanikov, A.-M. Bronstein, M.-M. Bronstein, and L.-J. Guibas, “Shapegoogle: a computer vision approach for invariant shape retrieval,” in *Workshop on Nonrigid Shape Analysis and Deformable Image Alignment (NORDIA)*, 2009.
11. N. Cristianini, J. Shawe-Taylor, and H. Lodhi, “Latent semantic kernels,” *Journal of Intelligent Information Systems* **18**, pp. 127–152, 2002.
12. S. Katz, G. Leifman, and A. Tal, “Mesh segmentation using feature point and core extraction,” *The Visual Computer* **25**, pp. 865–875, 2005.
13. K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci, “Loops in reeb graphs of 2-manifolds,” in *ACM Symposium on Computational Geometry*, pp. 344–350, 2003.
14. E. Paquet and M. Rioux, “Nefertiti: a query by content system for three-dimensional model and image databases management,” *Image and Vision Computing* **17**, pp. 157–166, 1999.
15. R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, “Shape distributions.,” *ACM Transactions on Graphics (TOG)* **21(4)**, pp. 807–832, 2002.
16. O. Duda, P.-E. Hart, and D.-G. Stork, “Pattern classification,” in *John Wiley Sons*, 2000.
17. D. Pelleg and A. Moore, “X-means: Extending k-means with efficient estimation of the number of clusters,” in *International Conference on Machine Learning*, 2000.
18. V. Vapnik, “Statistical learning theory,” *Wiley* , 1998.
19. S. Marini, L. Paraboschi, and S. Biasotti, “Shape retrieval contest 2007: Partial matching track.,” *SHREC (in conjunction with IEEE Shape Modeling International)* , p. 1316, (2007).
20. A. Bronstein, M. Bronstein, and R. Kimmel, “Efficient computation of isometry-invariant distances between surfaces,” *IEEE Trans. VCG* **13/5**, pp. 902–913, 2007.
21. R. Sumner and J. Popovic, “Deformation transfer for triangle meshes,” in *International Conference on Computer Graphics and Interactive Techniques*, 2004.
22. D. Giorgi, S. Biasotti, and L. Paraboschi, “Shape retrieval contest 2007: Watertight models track,” 2007.
23. M. Chaouch and A. Verroust-Blondet, “3d model retrieval based on depth line descriptor,” in *IEEE International Conference on Multimedia and Expo*, 2007.