

Non-rigid 3D shape classification using Bag-of-Feature techniques

Hedi Tabia, Olivier Colot, Mohamed Daoudi, Jean-Philippe Vandeborre

▶ **To cite this version:**

Hedi Tabia, Olivier Colot, Mohamed Daoudi, Jean-Philippe Vandeborre. Non-rigid 3D shape classification using Bag-of-Feature techniques. IEEE International Conference on Multimedia and Expo (ICME), Jul 2011, Barcelona, Spain. pp.475. hal-00666732

HAL Id: hal-00666732
https://hal.science/hal-00666732

Submitted on 6 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NON-RIGID 3D SHAPE CLASSIFICATION USING BAG-OF-FEATURE TECHNIQUES

Hedi Tabia, Olivier Colot

University Lille 1, France
LAGIS (FRE CNRS 3303)

Mohamed Daoudi, Jean-Philippe Vandeborre

Institut TELECOM, TELECOM Lille 1, France
LIFL (UMR USTL/CNRS 8022)

ABSTRACT

In this paper, we present a new method for 3D-shape categorization using Bag-of-Feature techniques (BoF). This method is based on vector quantization of invariant descriptors of 3D-object patches. We analyze the performance of two well-known classifiers: the Naïve Bayes and the SVM. The results show the effectiveness of our approach and prove 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.

Index Terms— Bag-of-Feature, 3D-Shape, classification.

1. INTRODUCTION

The 3D-object classification 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. Some paper such as [1, 2] have addressed the categorization problem. Their methods distinguish the training step, in which the classes of the database are constructed, from the actual classification, that associates the object model to one class. These methods are based on a Bayesian classifier and they are limited to specific datasets. Biasotti et al. [3], proposed a comparison framework for 3D-object classification methods. The method compared in the framework are based on global descriptors. Such descriptors are not robust to the high variability that objects can undergo. Object classification must be invariant to different transformations and shape variations. This problem has an analogy with image domain, image categorization, which consists to group images depicting similar scenes or objects. 3D-objects as well as images, may undergo significant variability and the big challenge is to create matching techniques that would be insensitive to such changes. In the areas of computer vision and pattern recognition, bag-of-feature methods (BoF) have recently gained great popularity due to the introduction of the scale invariant feature transform [4]. These methods show good performance in many problems such as object recognition [5] and image categorization [6, 7]. The main advantage of feature-based approaches in image categorization problems is that they allow to think of images as a collection of visual "words", and hence use the well-developed methods from text categorization [8].

Our work addresses the categorization problem with a BoF approach. We developed a new feature extraction algorithm. We propose and compare two alternative implementations using different classifiers: Naïve Bayes and SVM. The main advantages of the method are that it is simple, computationally efficient and intrinsically invariant. While working on this paper, we discovered that [9] has used the bags of features for 3D-object categorization. Toldo's categorization framework is based on semantic segmentation. In general, the problem of segmenting a 3D object into meaningful parts is not a trivial issue. Their framework is quite sensitive to the identification of the boundaries of the meaningful part. The focus of this paper is to give a meta-algorithm where we combine feature point sampling, geodesic histograms and BoF classifiers.

The remainder of the paper is organized as follows. In Section 2, the method is detailed. Then, in Section 3 the experiments are presented. Conclusions and future developments end the paper.

2. METHOD OVERVIEW

Our method encompasses four main steps. 1) The first step is the detection and description of 3D-object patches, 2) The second step consists to assign patch descriptors to a set of predetermined clusters (an alphabet set) with a vector quantization algorithm, 3) The third one is the construction of a bag of keyshapes, which takes account of the number of patches assigned to each cluster and finally applying a multi-class classifier, treating the bag of keyshapes as the feature vector, and thus determine which class or categories to assign to the 3D-object to be classified.

In order to increase classification accuracy and decrease the computational effort, the descriptors extracted in the first step should be invariant to variations that are irrelevant to the categorization task (deformable shapes) but sufficiently rich to discriminate among the various classes at the category level. The alphabet used in the second step should be large

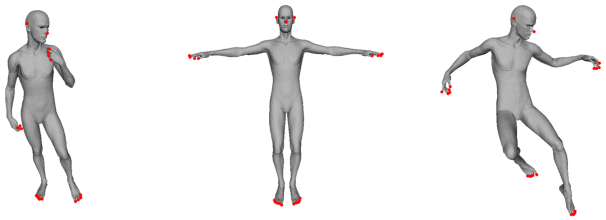


Fig. 1. Feature points extracted from different poses of 3D-model.

enough to distinguish relevant changes in 3D-object patches, but not very large as to distinguish irrelevant variations such as noise. By analogy with "keywords" in text categorization, we refer to the quantized feature vectors (cluster centers) as "keyshape". Nevertheless, "words", in our case, do not necessarily have a well-formed meaning such as "legs", or "head", nor is there an obvious best choice of alphabet. Rather, our goal is to utilize a alphabet that allows good categorization performance on a given training dataset. Therefore, the steps involved in training the system allow consideration of multiple possible sets of alphabets: 1) Detection and description of 3D-object patches for a set of labeled training objects. 2) Constructing a set of alphabets: each is a set of cluster centers, with respect to which descriptors are vector quantized. 3) Extracting bags of keyshape for these alphabets. 4) Training multi-class classifiers using the bags of keyshapes as feature vectors. 5) Selecting the alphabet and classifier giving the best overall classification accuracy. We now discuss the choices made for each step in more detail.

2.1. 3D feature extraction

2.1.1. 3D patches detection

In this section, 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 [10, 11]. However it is difficult to find a formal definition that characterizes this concept. In Katz et al. [11] 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. [12], 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 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 [13, 14]). 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 robust 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)) have also 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

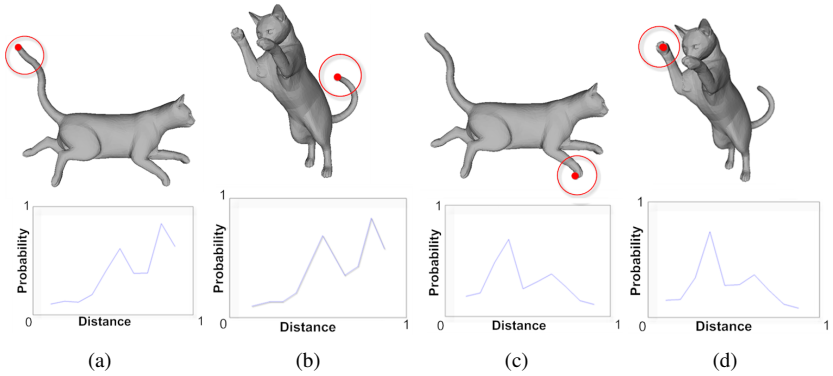


Fig. 2. Distance-distributions on four different feature points.

vector:

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

where $p_r = \int_{(r-1)/R}^{r/R} f(d)\delta d$. The descriptor $P(F_i)$ 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 alphabet

In BoF approach, the alphabet is obtained by quantification of the set of descriptors extracted in the training stage. The alphabet is used to construct discriminant representatives, with which any 3D-object can be described. The most common method to build the shape alphabet is to arrange descriptors encountered in the training stage into a finite number of clusters using a clustering algorithm. The number of clusters is alphabet size. For this end, we chose to use the k-means algorithm [15]. It proceeds by iterated assignments of points to their closest cluster centers and re-computation of the cluster centers. We 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 [16].

2.3. 3D-object categorization

After assigning each descriptor to its closest cluster, the problem of categorization is scaled down to that of multi-class supervised learning. In order to make a decision about an object to be classified the categorizer performs two steps: training and testing. The purpose of the training is to achieve correct categorization of future 3D-objects. Based on knowledge learned on labeled data, the categorizer make a decision rule for distinguishing categories. By applying this decision rule on the object to be classified, the categorizer predicts the class of that object. In this paper, we analyze the behavior of two well-known classifier: The Naïve Bayes and the Support Vector Machine.

2.3.1. Naïve Bayes classifier

The Naive Bayes Classifier [17] is a probabilistic classifier based on the Bayesian theorem. To demonstrate the concept of shape categorization using Naïve Bayes classifier, let us assume we have a set of labeled 3D-object $O = \{O_i\}$ and an alphabet $V = \{v_t\}$ of representative keyshapes. Each descriptor extracted from a 3D-object is labeled with the keyshape to which it lies closest in feature space. We count the number $N(t, i)$ of times keyshape v_t occurs in object O_i. To categorize a new 3D-object, we apply Bayes's rule and take the largest a posteriori score as the prediction:

$$P(C_j/O_i) \propto P(O_i/C_j)P(C_j) = P(C_j)\prod_{t=1}^{|V|} P(v_t/C_j)^{N(t,i)}. \quad (1)$$

It is evident in this formula that Naïve Bayes requires estimates of the class conditional probabilities of keyshape v_t given category C_j. In order to avoid probabilities of zero, these estimates are computed with Laplace smoothing:

$$P(v_t/C_j) = \frac{1 + \sum_{O_i \in C_j} N(t, i)}{|V| + \sum_{s=1}^{|V|} \sum_{O_i \in C_j} N(s, i)}. \quad (2)$$

2.3.2. SVM classifier

The SVM classifier is a classification method that performs classification tasks by constructing hyperplanes in a multi-dimensional space that separates cases of different class labels with maximal margin [16]. In order to apply the SVM to multi-class problems we take the one-against-all approach. Given an m class problem, we train m SVM's, each distinguishes object from some category i from objects from all the other $m - 1$ categories j not equal to i. Given an object to be classified, we assign it to the class with the largest SVM output.

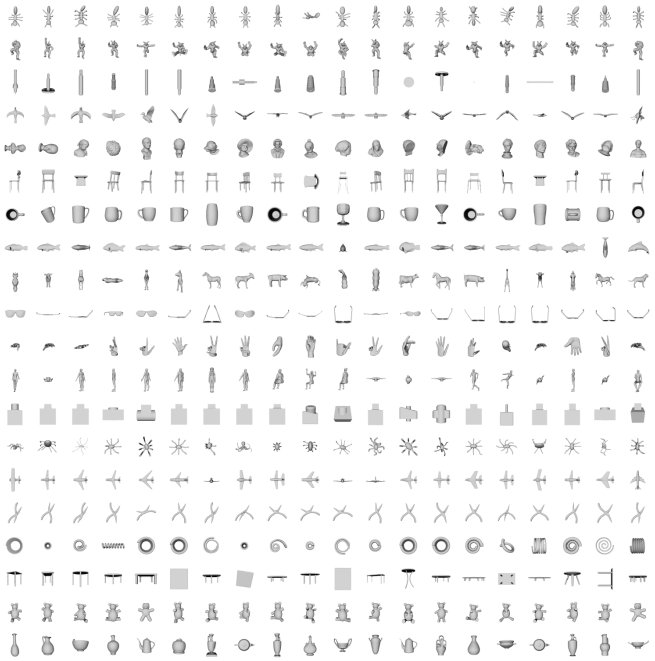


Fig. 3. Data-set snapshot.

3. EXPERIMENTAL RESULTS

In this section, we give results from three experiments. In the first experiment, we test the effect of the number of clusters on classifier accuracy and analyze the effectiveness of the Naïve Bayes classifier. In the second experiment, we analyze the performance of the SVM on the same problem. These experiments were conducted on a 20 class dataset. In the last experiment we present results on the fourteen class dataset employed in [3]. Our database contains 400 3D-objects in 20 classes. Figure 3 shows some examples from this dataset.

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.

We used three performance measures to evaluate our multi-class classifiers.

- The confusion matrix:

$$M_{ij} = |\{O_k \in C_j : h(O_k) = i\}| \quad (3)$$

  $i, j \in \{1, ..., N_C\}$, C_j is the set of test objects from category j and $h(O_k)$ is the category which obtained the highest classifier output for object O_k.

- The overall error rate:

$$R = 1 - \frac{\sum_{j=1}^{N_C} M_{jj}}{\sum_{j=1}^{N_C} |C_j|} \quad (4)$$

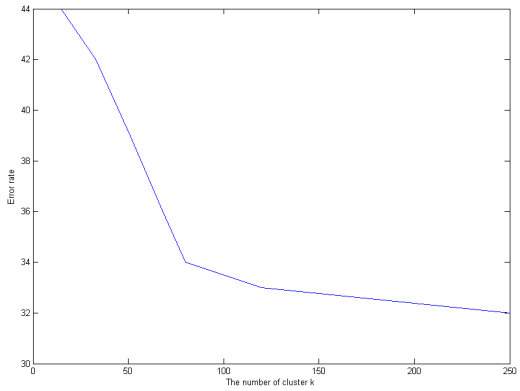


Fig. 4. The overall error rate for different choices of k.

- The mean ranks: these are the mean position of the correct labels when labels output by the multi-class classifier are sorted by the classifier score.

Each performance metric was evaluated with 10-fold cross validation.

3.1. Results from Naïve Bayes classifier

In Figure 4 we present the overall error rates using Naïve Bayes as a function of the number of clusters k. Each point in Figure 4 is the best of 10 random trials of k-means. The error rate only improves slightly as we move from k = 100 to k = 250. We therefore assert that k = 100 presents a good trade-off between accuracy and speed[1].

Table 1 shows the performance as a function of category obtained with k=100. Due to the size of the confusion matrix, we display in this table only the values corresponding to the diagonal of that matrix M_{jj}. The diagonal elements are the counts of the correct predictions.

3.2. Results from SVM classifier

Results from applying the SVM are presented in Table 2.

As awaited the SVM performance surpass the performance of Naïve Bayes classifier, reducing the overall error rate from 33 to 24%.

In training this SVM we used the same best alphabet with k=100 as for Naïve Bayes. We compared linear, quadratic and cubic SVM's and found that linear method gave the best performance. The penalty parameter Cst of the SVM classifier was determined for each SVM and values of around $Cst = 0.006$ typically gave the best results.

From a qualitative point of view, Figure 5 shows the different behavior of the two classifiers on a set of twenty ob-

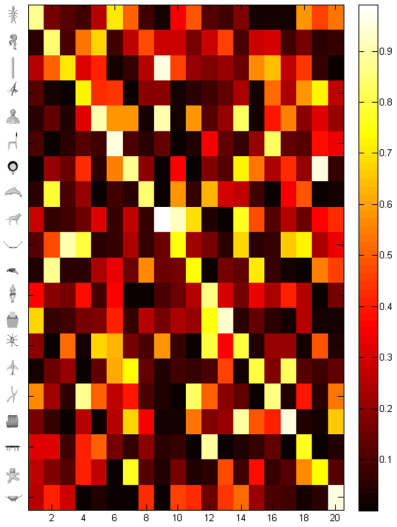
[1] It takes less than 10 seconds to get predicted labels on the whole database using the Naive Bayes classifier with k = 100 using a PC with a 3 Ghz Core 2 Duo processor with 3 GB memory.

True classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M_{jj}	**12**	**13**	**15**	**15**	**11**	**12**	**13**	**14**	**12**	**13**	**14**	**14**	**14**	**13**	**15**	**13**	**14**	**15**	**14**	**12**
Mean ranks	1.59	1.87	1.34	1.32	1.76	1.25	1.77	1.33	1.34	1.37	1.34	1.38	1.37	1.25	1.54	1.98	1.58	1.87	1.54	1.47

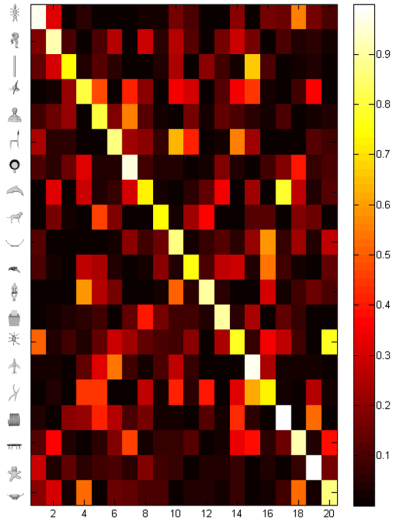
Table 1. Naïve Bayes classification Results

True classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M_{jj}	**15**	**16**	**16**	**16**	**15**	**16**	**14**	**15**	**16**	**14**	**17**	**16**	**15**	**14**	**16**	**13**	**15**	**16**	**16**	**13**
Mean ranks	1.10	1.24	1.67	1.35	1.54	1.38	1.58	1.57	1.13	1.58	1.12	1.35	1.29	1.22	1.17	1.24	1.28	1.34	1.11	1.25

Table 2. SVM classification result.



(a)



(b)

Fig. 5. Example of some 3D-objects classification. (a) Naïve Bayes classifier results. (b) SVM classifier results

	Methods				
	SH	LF	MRG	ERG	Our approach (SVM)
Rate	89%	88 %	88 %	83 %	91%
Mean rank	2%	2 %	2 %	2 %	1.62%

Table 3. Classification result comparison with related work.

jects selected from the database. Rows in Figure 5 correspond to 3D-objects (the objects correspond to the last column in Figure 3), and columns correspond to the different categories shown in the rows of Figure 3 (in the same order). In this visualization, the lightness of each element $(i; j)$ is proportional to the magnitude of the probability of the belong of the object i to the category j. Lighter elements represent better matches, while hot elements indicate worse matches. The lightness of the diagonal squares of the matrix proves the effectiveness of the classifier.

One can notice in this visualization that the SVM classifier (Figure 5(b)) shows more robust results compared with the Naïve Bayes classifier (Figure 5(a)).

3.3. Comparison with related work

In this experiment we compare our method effectiveness to methods proposed in Biosotti et al. framework [3]. The framework adapts four popular shape retrieval methods for the classification problem. The spherical harmonics (SH) in [18] which is a volume-based descriptor, The light-field descriptor (LF) in [19] which is an image-based descriptor, and two topological matching methods, the Multi-resolution Reeb graph in (MRG) [20] and the Extended Reeb graph (ERG) in [21]. The results of the experiment are shown in Table 3. Each entry is related to the performance of a given shape descriptor (enumerated in the first row). The performance is evaluated in terms of classification rate (i.e. the percentage of object models which are correctly classified). Notice that these rates are computed when the position of the correct class with respect to the rank identified by the classifier is set to be 2 as in [3]. Our classifier show the highest classification rate 91%.

4. CONCLUSION

In this paper, we have presented a novel method for categorizing 3D-objects based on the BoF techniques. We have proposed a new feature extraction algorithm. We also have presented and compared two alternative implementations using different classifiers: Naïve Bayes and SVM. We have presented results for simultaneously classifying twenty semantic shape categories. 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. The classifiers have been compared with some related work and demonstrate effective results. Our system achieved a classification accuracy over 91%. As future works, we propose to integrate other descriptors and we will focus on combining the two classifiers.

5. REFERENCES

[1] D. Huber, A. Kapuria, R. Donamukkala, and M. Hebert, "Parts-based 3d object classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[2] R. Donamukkala, D. Huber, A. Kapuria, and M. Hebert, "Automatic class selection and prototyping for 3-d object classification," *3-D Digital Imaging and Modeling (3DIM)*, 2005.

[3] S. Biasotti, D. Giorgi, S. Marini, M. Spagnuolo, and B. Falcidieno, "A comparison framework for 3d object classification methods," in *Multimedia content representation, classification and security*, 2006, vol. 4105, pp. 314–321.

[4] D. Lowe, "Distinctive image features from scale-invariant keypoint," *IJCV*, 2004.

[5] S. Lazebnik, C. Schmid, and J. Ponce, "Semi-local affine parts for object recognition," in *Proceedings of the British Machine Vision Conference*, 2004.

[6] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," *ECCV'04 workshop on Statistical Learning in Computer Vision*, pp. 59 – 74, 2004.

[7] H. Jgou, M. Matthijs Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *International Journal of Computer Vision*, vol. 87, pp. 316–336, 2010.

[8] N. Cristianini, J. Shawe-Taylor, and H. Lodhi, "Latent semantic kernels," *Journal of Intelligent Information Systems*, vol. 18, pp. 127–152, 2002.

[9] R. Toldo, U. Castellani, and A. Fusiello, "A bag of words approach for 3d object categorization," in *Proceedings of the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques*, 2009.

[10] J. Tierny, J-P. Vandeborre, and M. Daoudi, "Partial 3D shape retrieval by reeb pattern unfolding," *Computer Graphics Forum - Eurographics Association*, vol. 28, pp. 41–55, March 2009.

[11] S. Katz, G. Leifman, and A. Tal, "Mesh segmentation using feature point and core extraction," *The Visual Computer*, vol. 25, pp. 865–875, 2005.

[12] 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*, 2003, pp. 344–350.

[13] E. Paquet and M. Rioux, "Nefertiti: a query by content system for three-dimensional model and image databases management," *Image and Vision Computing*, vol. 17, pp. 157–166, 1999.

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

[15] O. Duda, P-E. Hart, and D-G Stork, "Pattern classification," in *John Wiley Sons*, 2000.

[16] V. Vapnik, "Statistical learning theory," *Wiley*, 1998.

[17] D. Lewis, "Nave bayes at forty: The independence assumption in information retrieval," *ECML*, 1998.

[18] M. Kazhdan, T. Funkhouser, and S Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3d shape descriptors," *Geometry Processing, Aachen, Germany*, 2003.

[19] D-Y. Chen, X-P. Tian, Y-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Eurographics*, vol. 22, pp. 223–232, 2003.

[20] M. Hilaga, Y. Shinagawa, T. Kohmura, and T Kunii, "Topology matching for fully automatic similarity estimation of 3D shapes," in *SIGGRAPH*, 2001, pp. 203–212.

[21] S. Biasotti and S. Marini, "3d object comparison based on shape descriptors," *International Journal of Computer Applications in Technology*, vol. 23, pp. 57–69, 2005.