Non-rigid and Partial 3D Model Retrieval Using Hybrid Shape Descriptor and Meta Similarity

Bo Li[1], Afzal Godil[1], and Henry Johan[2]

[1] National Institute of Standards and Technology, Gaithersburg, USA
{bo.li,afzal.godil}@nist.gov
[2] School of Computer Engineering, Nanyang Technological University, Singapore
henryjohan@ntu.edu.sg

Abstract. Non-rigid and partial 3D model retrieval are two significant and challenging research directions in the field of 3D model retrieval. Little work has been done in proposing a hybrid shape descriptor that works for both retrieval scenarios, let alone the integration of the component features of the hybrid shape descriptor in an adaptive way. In this paper, we propose a hybrid shape descriptor that integrates both geodesic distance-based global features and curvature-based local features. We also develop an adaptive algorithm to generate meta similarity resulting from different component features of the hybrid shape descriptor based on Particle Swarm Optimization. Experimental results demonstrate the effectiveness and advantages of our framework. It is general and can be applied to similar approaches that integrate more features for the development of a single algorithm for both non-rigid and partial 3D model retrieval.

1 Introduction

Non-rigid 3D model retrieval is a challenging research direction for the community of 3D model retrieval. Compared to generic 3D model retrieval, partial similarity 3D model retrieval is also more difficult and much less studied. Geodesic distance-based global features have intrinsic advantages in characterizing non-rigid 3D models and also have shown their superiority in recognizing deformable models, which has been demonstrated by Smeets et al. [1] [2]. On the other hand, employing local features and Bag-of-Words [3] framework has demonstrated its apparent advantages in dealing with partial similarity retrieval, such as [4] [5]. Curvature is an important local feature and it is the basis of several other important local features, such as Shape Index [6] and Curvedness [6]. Motivated by this, our target is to utilize both geodesic distance-based global features and curvature-based local features together with the Bag-of-Words framework to develop a 3D shape retrieval algorithm that can be used for both non-rigid and partial similarity retrieval. Geodesic distance-based and curvature-based features show different properties and retrieval performances in recognizing non-rigid or partial 3D model retrieval. To adaptively combine these two features, a meta similarity based on Particle Swarm Optimization (PSO) [7] has been proposed to fuse their distance matrices. This framework is general and can be extended to integrate different or more features to develop other similar unified retrieval algorithms for both non-rigid and partial 3D model retrieval.

G. Bebis et al. (Eds.): ISVC 2012, Part I, LNCS 7431, pp. 199–209, 2012.
© Springer-Verlag Berlin Heidelberg 2012

The paper is organized as follows. We briefly discuss the related work in Section 2. Section 3 introduces the hybrid 3D shape descriptor. Section 4 presents our 3D model retrieval algorithm, together with the method of weight assignment for the meta similarity based on Particle Swarm Optimization. We give in detail the experiments in Section 5 and conclude the paper and list the future work in Section 6.

2 Related Work

During the past few years, geodesic distance-based, and local feature together with Bag-of-Words framework based approaches have received much attention, especially in dealing with the non-rigid and partial 3D model retrieval. Combining and integrating heterogeneous features is also an important issue if we employ a hybrid shape descriptor comprising several features. We give a brief review for these four topics as follows.

Geodesic Distance-Based Descriptors. Geodesic distance is an inelastic deformation invariant distance metric, thus popular for the analysis and recognition of non-rigid objects. Typically, the extracted geodesic distance-based feature for 3D is a geodesic distance matrix (GDM) measuring the distances among a set of points sampled on the surface of a 3D object. To deal with deformable 3D model retrieval, Smeets et al. [1] proposed a modal representation method based on the Singular Value Decomposition (SVD) of the GDM of a 3D model. They utilized several largest eigenvalues of a GDM as the shape descriptor. In SHREC 2011 Non-rigid watertight shape retrieval track [2], Smeets et al. further proposed a method by combing GDM and another method called Scale Invariant Feature Transform (SIFT) for meshes (meshSIFT) and they achieved the best retrieval performance among the nine participants.

Local Shape Descriptors. Paul et al. [8] presented a comparative evaluation of several local shape descriptors. Koenderink and Doorn [6] proposed a curvature-based local feature named Shape Index which measures the local topological/convexity geometry, such as ridge, saddle, cup and cap and another local feature called Curvedness which measures the amount of curvature. 3D shape spectrum [9] based on Shape Index distribution was also proposed as the MPEG 3D shape feature standard.

Bag-of-Words Framework. Recently, the Bag-of-Words (BoW) framework has been successfully applied into 3D model retrieval. It has demonstrated successful applications in either view-based (e.g. [10]) or geometry-based (e.g. [4], [5]) 3D model retrieval and apparent advantages in partial similarity 3D model retrieval (e.g. [4], [5]), as well. To reduce the computational cost for distance computation, Ohbuchi et al. [10] encoded the SIFT features of a set of depth views of a 3D model into a histogram by utilizing the BoW approach. Toldo et al. [4] extended the BoW framework from 2D to 3D to represent 3D components. 3D subparts resulting from segmentation are clustered to define a 3D vocabulary comparable to the 2D codewords. Lavoué [5] applied the BoW framework to the Laplace-Beltrami spectrum features of a set of uniformly sampled points on the surface of a 3D model by projecting the geometry onto the eigenvectors of the Laplace-Beltrami operator and also achieved superior partial retrieval performance.

Meta Similarity. Employing several features together in 3D shape retrieval needs a solution of integrating them properly to make them compliment each other to achieve the optimal performance. In the field of 3D model retrieval, compared to new shape descriptors, this topic has received less attention and is also less studied, let alone the adaptive approaches of weight assignment to generate the meta similarity. We can merge several feature vectors directly or merge the distances resulting from different features, as well. Akbar et al. [11] combined features extracted from surface and volume by assigning the weights based on the properties of the two features and they tested on both merging schemes. Unfortunately, the retrieval performance improvement is not apparent. Daras et al. [12] investigated several factors that affect retrieval performance, such as feature selection, dissimilarity metric, feature combination and weight optimization and they suggested that more focus should be given to the efficient combination of low-level descriptors rather than the investigation of the optimal 3D shape descriptor.

3 Hybrid 3D Shape Descriptor

In this section, to represent a 3D model we propose a hybrid shape descriptor composed of a curvature-based local feature vector V_C proposed by us and a geodesic-based global feature vector V_G, described as follows.

3.1 Curvature-Based Local Feature Vector: V_C

Extracting local features are important for partial similarity 3D model retrieval. First, we propose a curvature-based combined local shape descriptor for each vertex of a 3D model and after that we apply the Bag-of-Words framework to generate the local shape descriptor distribution as our proposed local feature vector V_C. To extract the local shape descriptor, we need to define its two basic components: local support region and local features. We regard the adjacent vertices of a vertex as its local support region and consider the following first three curvature-based local features.

(1) Curvature Index Feature. Curvature is an important feature to characterize the local geometry. Based on curvature, Koenderink and Doorn [6] proposed Shape Index and Curvedness. Curvature Index [8] further maps Curvedness values into a reasonable range using a log function. For a vertex p, its Curvature Index CI is computed as follows,

$$CI = \frac{2}{\pi} log(\sqrt{\frac{K_1^2 + K_2^2}{2}}) \tag{1}$$

where K_1 and K_2 are the two principal curvatures in the x and y directions respectively at the point of vertex p.

(2) Curvature Index Deviation Feature. To measure the tendency of the Curvature Index change in a local support region of a vertex, we compute the standard deviation Curvature Index difference of the adjacent vertices of the vertex p,

$$\delta CI = \sqrt{\frac{\sum_{i=1}^{n} (CI_i - \widetilde{CI})}{n}} \tag{2}$$

where CI_1, CI_2,...,CI_n are the Curvature Index values of the adjacent vertices of p and $\widetilde{CI}$ is the mean Curvature Index of all the adjacent vertices.

(3) Shape Index Feature. Shape Index [6] is a feature that has been applied into generic 3D shape retrieval. Here, we utilize it within the Bag-of-Words framework for non-rigid and partial 3D model retrieval. Its definition is as follows,

$$SI = \frac{2}{\pi}\arctan(\frac{K_1 + K_2}{|K_1 - K_2|}) \tag{3}$$

where K_1 and K_2 are the two principal curvatures in the x and y directions respectively at the point of vertex p. $SI \in$[-1,1].

(4) Combined Local Shape Descriptor. The three local features described above depict the local properties in different aspects. To more comprehensively measure the local information, a combined local shape descriptor F comprising the above three features is devised,

$$F = (CI, \delta CI, SI) \tag{4}$$

(5) Local Feature Vector Generation: Bag-of-Words. We regard the combined local shape descriptor distribution of all the vertices of a 3D model, with respect to a set of centers, as its local feature vector V_C. Based on the Bag-of-Words framework, the local feature vector generation process includes the following two steps: 1) **Codebook generation**. We cluster the combined local shape descriptors of the vertices of all the 3D models in a 3D dataset into a set of class centers (codewords) $O_1, O_2, \ldots, O_{N_C}$ based on K-means algorithm, where N_C is the number of codewords. In our experiments, L2 distance metric, N_C=500 cluster centers and 100 maximum clustering iteration number are experimentally determined. 2) **Local feature vector formulation**. Based on the generated codebook (cluster centers), we count the distribution V_C of the local shape descriptors of all the vertices of a 3D model with respect to the codewords in terms of maximum similarity,

$$V_C = (h_1, h_2, \cdots, h_{N_C}), \tag{5}$$

where h_i is the percentage of the local shape descriptors whose closest codeword is O_i. To find the closest codeword, Canberra distance metric [13] is utilized to measure the difference between two combined local shape descriptors F_i and F_j: $d_F = \frac{1}{n}\sum_{l=1}^{n}\frac{|F_i(l)-F_j(l)|}{|F_i(l)+F_j(l)|}$, where n is the dimension of F_i and F_j, $d_F \in [0,1]$.

3.2 Geodesic Distance-Based Global Feature Vector: V_G

For non-rigid 3D model retrieval, by utilizing the eigenvalues of global geodesic distance matrix (GDM), Smeets et al. [1] [2] have achieved outstanding retrieval performance. Global GDM considers the geodesic distances among all the sample points on the surface of a 3D model to form a 2D square distance matrix. The eigenvalues of the GDM is comparable to the spectrum of a 3D shape, which shows superior performance when dealing with non-rigid 3D model retrieval. Hybrid approaches by combining global and local features like [14] have been verified to be an effective way to develop a more comprehensive shape descriptor to further improve the retrieval performance. Considering this, we also compute a global geodesic distance matrix-based feature for a 3D model, especially for non-rigid 3D model retrieval.

(1) 3D Model Simplification. To reduce computational cost for geodesic distance-based feature extraction, we simplify each model by adopting the mesh simplification method proposed by Garland and Heckbert [15]. It iteratively contracts vertices pairs under the control of quadric surface error. It is efficient and preserves the most important features. In experiments, we simplify the models to make they contain the same number (e.g. 1000 in our experiments) of vertices.

(2) Geodesic-Based Global Feature Vector Generation. We first compute the geodesic distances based on the method in [16] among all the vertices of a simplified model to form a geodesic distance matrix GDM. Then we decompose the GDM based on Singular Value Decomposition (SVD) and keep the first largest k (e.g. 50 in our experiments) eigenvalues as the global feature vector V_G,

$$V_G = (e_1, e_2, \cdots, e_k) \tag{6}$$

where k is the threshold number of eigenvalues that we are interested in. Similarly, Canberra distance (Section 3.1) is used to measure the distance between two V_G.

4 Non-rigid and Partial 3D Model Retrieval Algorithm Based on a Hybrid Shape Descriptor and Meta Similarity

4.1 Retrieval Algorithm

Given a query 3D model and a target 3D model database, we retrieve relevant models from the target database. Our 3D model algorithm is based on the hybrid shape descriptor presented in Section 3. The complete retrieval algorithm is as follows.

(1) Curvature-based local feature vector V_C and local feature distance matrix M_C computation. For each query and target 3D model, we extract its curvature-based local feature vector V_C as described in Section 3.1. It is very efficient, so we consider all the available vertices and use the original models directly. After that, we compute the Canberra distance (Section 3.1) between the local feature vectors of a query model and a target model to form the curvature-based local feature distance matrix M_C.

(2) Geodesic distance-based global feature vector V_G and global feature distance matrix M_G computation. Based on the algorithms presented in Section 3.2, we simplify each query or target model to make it has 1000 vertices and keep the largest 50 eigenvalues as its global feature vector V_G. Similarly, the Canberra distance between a query and a target model's global feature vectors V_G is computed to form the geodesic distance-based global feature distance matrix M_G.

(3) Meta distance matrix generation and ranking. We adaptively find the weights w_C and w_G for the distance matrices M_C and M_G respectively to generate a meta distance matrix M based on the approach in Section 4.2.

$$M = w_C * M_C + w_G * M_G \tag{7}$$

where w_C and w_G are in the region of [0,1]. Finally, we sort all the models in the database in ascending order based on their distances and output the retrieval lists accordingly. The two weights w_C and w_G are needed to be computed only once for each target database which is always available in order to perform a retrieval. If the query database is available, we use it directly as queries to compute the weight values, otherwise, we use the target models as queries. In our experiments, we use the target models directly in Section 5.1 and use the query database in Section 5.2.

4.2 Meta Similarity by Particle Swarm Optimization

The simplest method to find the optimal weights for different features is by performing a brute-force search. We can uniformly sample the values by adopting a fixed step. The drawback of the brute-force search is the high computational cost. For example, in order to find a result with an accuracy of $\Delta\delta$ (e.g. 0.01) for N (e.g. 3) weights, we have to sample at least at a step of $\Delta\delta$ (e.g. 0.01), which means $(\frac{1}{\Delta\delta})^{N-1}$ (e.g. 10000) combinations. As such, the brute-force search is not the ideal method for finding the optimal weights.

To efficiently find the optimal weights, we develop a weight assignment method based on Particle Swarm Optimization (PSO) [7] which is a swarm intelligence optimization technique by imitating the behavior of a flock of birds searching for a piece of food in a region. Each bird learns from its neighboring birds and update itself based on the position of the bird nearest to the food. Our PSO-based weight assignment for the meta distance matrix generation is as follows.

(1) PSO Initialization. We initialize the number N_P and positions of a set of search particles $\{x = (w_C, w_G)\}$ and then compute the private best for each particle and current global best based on all the private bests. In experiments, we uniformly distribute the search particles within its search region of $\{[0,1],[0,1]\}$. We regard the $\lfloor N_P/3 \rfloor$ nearest neighbors of a search particle as its neighborhood, based on which we compute its private best. Finally, we also set the maximum search iterations N_t.

(2) Update Particles. We update the position of each particle by adopting a similar strategy as [17],

$$x(i+1) = x(i) + s \cdot v(i), \tag{8}$$

$$v(i+1) = \omega * v(i) + c_1 \cdot r_1 \cdot (x_p(i) - x(i)) + c_2 \cdot r_2 \cdot (x_g(i) - x(i)). \tag{9}$$

$x(i)$ and $v(i)$ are the position and velocity of a particle; the velocity update step s is inversely proportional to the current iteration number i: $s = \frac{N_t - i}{N_t} + c$, where c is a constant variable and in experiments we choose c to be 0.5. r_1 and r_2 are random variables between 0 and 1; x_p and x_g are the particle positions of private and global bests. c_1 and c_2 are non-negative constants, typically c_1=c_2=2 [7]. The inertia-weight ω is a tradeoff between the global and local search abilities. Bigger ω indicates more powerful global search ability and less dependency on the initial locations of the search particles, while smaller ω means finer search within a local area. Similar as [17], we linearly decrease ω from 1.4 to 0 according to the iteration number i: $\omega = \frac{\omega_{min}-\omega_{max}}{N_t} \cdot i + \omega_{max}$, where ω_{max}=1.4 and ω_{min}=0. The new position $x(i+1)$ may be out of the search area, as such we clamp it by subtracting (if larger than 1) or adding (if smaller than 0) 1 .

(3) Search Evaluation. Based on the new position of each particle, we assign the corresponding weights w_C and w_G and compute the meta distance matrix based on Equation (7) and thus the corresponding retrieval performance metrics, such as First Tier (FT) and mean Normalized Discounted Cumulative Gain (NDCG) [18], and regard them as PSO fitness value to evaluate the weight assignment result. After that, we update its private best as well as the global best based on all the private bests.

(4) Result Verification. If the maximum iteration number N_t has been reached, we stop and output the position of the current global best as the optimal weight assignment result and also output the corresponding optimal meta distance matrix M and retrieval performance metrics; otherwise, go to step (2) to continue the search. The complexity of our POS-based weight assignment algorithm is $O(N_P + N_P \cdot N_t)$.

5 Experiments

To investigate the performance of our algorithm in terms of non-rigid and partial 3D model retrieval, we choose to use the following two benchmarks.

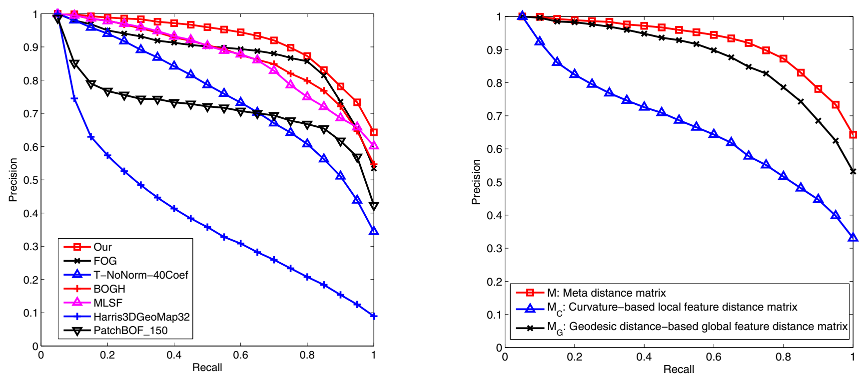
(1) **SHREC'11-Non-rigid:** the benchmark for the SHREC 2011 non-rigid 3D watertight models retrieval track [2]. It contains 600 watertight and deformable models, classified into 30 classes, each with 20 models.

(2) **SHREC'07-Partial:** the benchmark used in the SHREC 2007 partial matching track [19]. The target dataset has 400 watertight models, divided into 20 classes, each with 20 models. The query dataset comprises 30 models by combining the parts of two or more models of the target database.

To comprehensively evaluate the non-rigid 3D model retrieval results, we employ six metrics [18] including Precision-Recall (PR), Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), Discounted Cumulative Gain (DCG) and Average Precision (AP). We use the Normalized Discounted Cumulative Gain (NDCG) [18] metric to evaluate the performance of partial retrieval results.

5.1 Non-rigid 3D Model Retrieval

Geodesic distance is invariant to model deformation, which makes it has advantages in non-rigid 3D model retrieval. This means that we should increase its weight during the retrieval. While, adding curvature-based features will probably further improve the retrieval performance. However, it is non-trivial to find an optimal weight assignment



(a) Our retrieval algorithm and other methods (b) Hybrid shape descriptor and its components

Fig. 1. Precision-Recall performance comparison on the SHREC'11-Non-rigid benchmark

for these two features, let alone for more features. Thus, PSO-based algorithm is utilized to train the weights. We set N_P=10, N_t=20, and select First Tier as the PSO fitness value to evaluate search results. Based on the algorithm in Section 4.2, we find the optimal weights values: w_C=0.349036, w_G=0.650965. Optimal First Tier value is 0.864999.

Table 1. Other performance metrics comparison on the SHREC'11-Non-rigid benchmark

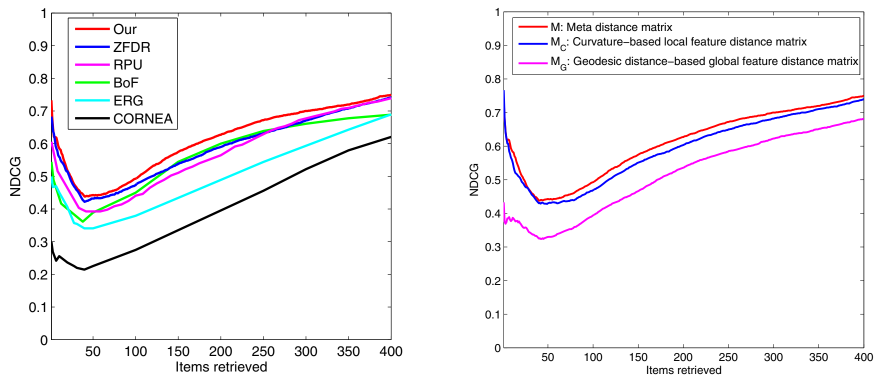
Methods	NN	FT	ST	DCG	AP
Our	**99.7**	**86.5**	**93.1**	**97.1**	**93.4**
FOG	96.8	81.7	90.3	94.4	89.5
T-NoNorm-40Coef	95.5	67.2	80.3	89.7	78.1
BOGH	99.3	81.1	88.4	94.9	89.1
MLSF	98.7	80.9	87.9	94.8	88.2
Harris3DGeoMap32	56.2	32.5	46.6	65.4	43.2
PatchBOF_150	74.8	64.2	83.3	83.7	74.1
M_C	83.7	58.8	77.2	83.7	69.7
M_G	99.3	81.4	88.1	95.3	89.4

We compare with the approaches in the SHREC 2011 Non-rigid track which mainly extract geodesic distance-based features, such as FOG, BOGH and Harris3DGeoMap32; or adopt the Bag-of-Words framework and some other geometric features, like T-NoNorm-40Coef, MLSF and PatchBOF. We also compare with the performances of the two component features of our hybrid shape descriptor, that is, comparing the performances of meta distance matrix M, curvature-based local feature distance matrix M_C, and geodesic distance-based global feature distance matrix M_G. Figure 1 compares their Precision-Recall performances while Table 1 lists their other performance metrics.

As can be seen from Figure 1 (a) and Table 1, our hybrid shape descriptor and meta similarity-based retrieval algorithm outperforms all the six participating approaches which use the features and the Bag-of-Words framework that fall in the same category as our approach. Based on the results shown in Figure 1 (b) and Table 1, we also find that our approach apparently improves the retrieval performances, in terms of all the six metrics, for non-rigid 3D model retrieval.

5.2 Partial Similarity 3D Model Retrieval

Unlike non-rigid model retrieval, in this case curvature-based local features will contribute more for partial 3D model retrieval. Similarly, we optimize their weights based on PSO after computing the curvature-based feature distance matrix M_C and geodesic distance-based feature distance matrix M_G. We set N_P=10 and N_t=20. Since NDCG is used to evaluate the partial retrieval performance, we use the mean NDCG over all the 400 models to evaluate search results. The optimal weights values are as follows: w_C=0.397384, w_G=0.602614, while the optimal mean NDCG is 0.613296. Similar as Section 5.1, the NDCG performance comparisons with the participants in the SHREC 2007 partial matching track [19] as well as other approaches mentioned in [18], and the hybrid shape descriptor's components are shown in Figure 2 (a) and (b), respectively. Based on the comparison results in Figure 2, we can draw a similar conclusion as the non-rigid retrieval experiments in Section 5.1 for the partial similarity retrieval.



(a) Our retrieval algorithm and other methods (b) Hybrid shape descriptor and its components

Fig. 2. NDCG performance comparison on the SHREC'07-Partial benchmark

6 Conclusions and Future Work

Non-rigid and partial 3D model retrieval are two important and challenging research directions in the field of 3D model retrieval. While different approaches based on either geodesic distance or some local features have been proposed to deal with either of the above two retrieval problems, little work has been done in developing a hybrid shape descriptor that works for both cases, especially in an adaptive way. We have found that geodesic distance-based global features and curvature-based local features have advantages in non-rigid and partial 3D model retrieval, respectively. To utilize both features and make them compliment each other, we develop a hybrid shape descriptor comprising these two types of features and adaptively combine their feature distance matrices to form a meta distance matrix based on Particle Swarm Optimization.

Experimental results based on a latest non-rigid 3D model retrieval benchmark and a partial 3D model retrieval dataset as well, demonstrate the effectiveness and advantages of our framework. It applies to two different, important and difficult retrieval scenarios and improves the retrieval performances based on an adaptive integration strategy. The idea is general and it can be applied to integrate three or more features for developing a single algorithm for both non-rigid and partial 3D model retrieval, which is also among our future work. Another interesting work is to test the performances of concatenating our global and local feature vectors directly to form a hybrid feature vector by assigning appropriate weights based on our Particle Swarm Optimization algorithm and then perform a comparative evaluation with the retrieval algorithm proposed in the paper.

References

1. Smeets, D., Fabry, T., Hermans, J., Vandermeulen, D., Suetens, P.: Isometric Deformation Modelling for Object Recognition. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 757–765. Springer, Heidelberg (2009)
2. Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoué, G., Nguyen, H.V., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Reuter, F.P.M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H., Vandermeulen, D.: SHREC 2011 track: Shape retrieval on non-rigid 3D watertight meshes. In: 3DOR, pp. 79–88 (2011)
3. Li, F.F., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: CVPR, vol. (2), pp. 524–531 (2005)
4. Toldo, R., Castellani, U., Fusiello, A.: Visual vocabulary signature for 3D object retrieval and partial matching. In: 3DOR, pp. 21–28 (2009)
5. Lavoué, G.: Bag of words and local spectral descriptor for 3D partial shape retrieval. In: 3DOR, pp. 41–48 (2011)
6. Koenderink, J.J., van Doorn, A.J.: Surface shape and curvature scales. Image Vision Comput. 10(8), 557–564 (1992)
7. Eberhart, R.C., Hu, X.: Human tremor analysis using particle swarm optimization. In: Proc. of the Congress on Evolutionary Computation, pp. 1927–1930 (1999)
8. Heider, P., Pierre-Pierre, A., Li, R., Grimm, C.: Local shape descriptors, a survey and evaluation. In: 3DOR, pp. 49–56 (2011)
9. Bober, M.: MPEG-7 visual shape descriptors. IEEE Trans. Circuits Syst. Video Techn. 11(6), 716–719 (2001)
10. Ohbuchi, R., Osada, K., Furuya, T., Banno, T.: Salient local visual features for shape-based 3D model retrieval. In: Shape Modeling International, pp. 93–102 (2008)
11. Akbar, S., Kueng, J., Wagner, R.: Multi-feature based 3D model similarity retrieval. In: 2006 International Conference on Computing Informatics, ICOCI 2006, pp. 1–6 (2006)
12. Daras, P., Axenopoulos, A., Litos, G.: Investigating the effects of multiple factors towards more accurate 3-D object retrieval. IEEE Transactions on Multimedia 14(2), 374–388 (2012)
13. Laga, H., Nakajima, M.: Supervised Learning of Similarity Measures for Content-Based 3D Model Retrieval. In: Tokunaga, T., Ortega, A. (eds.) LKR 2008. LNCS (LNAI), vol. 4938, pp. 210–225. Springer, Heidelberg (2008)

14. Wu, H.-Y., Zha, H., Luo, T., Wang, X., Ma, S.: Global and local isometry-invariant descriptor for 3D shape comparison and partial matching. In: CVPR, pp. 438–445 (2010)
15. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: SIGGRAPH, pp. 209–216 (1997)
16. Kalogerakis, E., Hertzmann, A., Singh, K.: Learning 3D mesh segmentation and labeling. ACM Trans. Graph. 29(4), 102:1–102:12 (2010)
17. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proc. of IEEE International Conference on Evolutionary Computation, ICEC, pp. 69–73 (1998)
18. Li, B., Johan, H.: 3D model retrieval using hybrid features and class information. Multimedia Tools and Applications, 1–26 (2011) (Online First version)
19. Veltkamp, R.C., ter Haar, F.B.: SHREC 2007 3D Retrieval Contest. Technical Report UU-CS-2007-015, Department of Information and Computing Sciences, Utrecht University (2007)