

3D shape Retrieval Using Bag-of-Feature Method Basing on Local Codebooks^{*}

El Wardani Dadi¹, El Mostafa Daoudi¹, and Claude Tadonki²

¹University of Mohammed First, Faculty of Sciences, LaRi Laboratory, Oujda, Morocco
wrd.dadi@gmail.com, m.daoudi@fso.ump.ma

²Mines ParisTech, Laboratory of Research in Computer, Math & System,
Fontainebleau, France
claude.tadonki@mines-paristech.fr

Abstract. Recent investigations illustrate that view-based methods, with pose normalization pre-processing get better performances in retrieving rigid models than other approaches and still the most popular and practical methods in the field of 3D shape retrieval [9,10,11,12]. In this paper we present an improvement of the BF-SIFT method proposed by Ohbuchi et al [1]. This method is based on bag-of-features to integrate a set of features extracted from 2D views of the 3D objects using the SIFT (Scale Invariant Feature Transform [2]) algorithm into a histogram using vector quantization which is based on a global visual codebook. In order to improve the retrieval performances, we propose to associate to each 3D object its local visual codebook instead of a unique global codebook. The experimental results obtained on the Princeton Shape Benchmark database [3] show that the proposed method performs better than the original method.

Keywords: 3D Content-based Shape Retrieval, bag-of-features, SIFT, vector quantization, codebook.

1 Introduction

Currently, there are an increasing number of 3D objects on the web, including large databases, thanks to recent digitizing and modeling technologies. The need of efficient methods for 3D shape-content based retrieval, in order to ease navigation into related large databases, and also to structure, organize and manage this new multimedia type of data, has become an active topic in various research communities such as *computer vision*, *computer graphics*, *mechanical CAD*, and *pattern recognition*.

One major challenge in 3D objects indexation is to design an efficient canonical characterization of the objects. In the literature, this characterization is referred to as a

^{*} E.W. DADI is supported by the "Excellence Grant of Moroccan Ministry of Higher Education. Grant No. G 08/004".

descriptor or a signature. Since the descriptor serves as a key in the search process, it is a critical kernel with a strong influence on the searching performances (i.e. computational efficiency and relevance of the results).

Various 3D shape description methods have been proposed in the literature. The reader may refer to a very good survey in [4] and a comparative study of 3D retrieval algorithms [5, 6, 7, 8]). Those algorithms can be clustered into two main families: 2D/3D approaches and 3D/3D approaches. For 2D/3D approaches, the description model is obtained through different 2D projections of the 3D shape, whereas for the 3D/3D approaches, the description model is obtained from the 3D information directly extracted from the 3D shape. Recent investigations illustrate that view-based methods with pose normalization pre-processing get better performance in retrieving rigid models than other approaches and still the most popular and practical methods in the field of 3D shape retrieval [9, 10, 11, 12].

Our work presented in this paper is inspired by the BF-SIFT method (Ohbuchi et al [13]), which is based on a global codebook (visual dictionary) used to describe each 3D objects in the database. We propose an improvement of the method by using local codebooks, since we think that using a unique global codebook badly influences the retrieval performance. On the Princeton Shape Benchmark (PSB) [3], that contains various shapes with more geometric details, experimental results show that our variant performs better and provides more accurate results.

The rest of the paper is organized as follows. Section 2 presents a description of the BF-SIFT algorithm. In section 3, we present our variant of the BF-SIFT method. Experimental results are provided and analyzed in Section 4. Section 5 opens some perspectives and concludes the paper.

2 The BF-SIFT Method

The BF-SIFT (Bag-of-Features - Scale Invariant Feature Transform) method proposed by Ohbuchi et al [13] compares 3D shapes using thousands of local visual features per model. A 3D model is rendered into a set of depth images, and from each image, local visual features are extracted by using the Scale Invariant Feature Transform (SIFT) algorithm of Lowe [2]. To efficiently compare among a large set of local visual features, the algorithm uses bag-of-features (BoF) approach in order to integrate, for each model, the local features into a vector of features. The BoF approach vector quantifies (or encodes) the SIFT features into a representative vector (or “visual word”), using a global codebook. The global codebook is generated with thousands of features extracted from a set of models in the retrieval database. In the following, we present an overview of the BF-SIFT algorithm:

- *Pose normalization (position and scale)*: The BF-SIFT performs pose normalization only for position and scale, so that the model is rendered with an appropriate size in each of the multiple-view images. Pose normalization is not performed for rotation.
- *Multi-view rendering*: in this step, a set of depth-buffer views of a 3D object are captured uniformly in all directions in order to catch up all symmetries.

- *SIFT feature extraction*: a 3D object can be approximately represented by a set of depth-buffers from which salient SIFT descriptors are extracted using the SIFT algorithm [2].
- *Vector quantization*: each 3D model is associated with thousands of local features. Each SIFT feature extracted from 3D models is quantified as a *vector* or *visual word* by using a global visual codebook. The vector quantification is to find frequencies of visual words (local features) generated from a model in the visual codebook which is learned, by using a clustering algorithm type (e.g. *k-means*, *kd-tree*, *ERC-tree*, and *Locality sensitive hashing*).
- *Histogram generation*: quantified local features or “visual words” are accumulated into a histogram with N_v bins (N_v is considered as the size of the codebook). The histogram becomes the feature vector of the corresponding 3D model.
- *Distance computation*: Dissimilarity among pairs of feature vectors (the histograms) is computed by using Kullback-Leibler Divergence (KLD).

$$D(x, y) = \sum_{j=1}^{N_v} (y_j - x_j) \ln \frac{y_j}{x_j}$$

Where $x = (x_i)$ and $y = (y_i)$ are the features vectors and N_v the dimension of the vectors.

3 Proposed Improvements Based on Local Codebooks

The construction of the visual codebook is one of the sensitive stages. Indeed, the descriptor of each object in the database will be calculated using the visual words in the codebook. For that, it is important to generate codebooks as representative as possible.

We propose to construct a visual codebook for each 3D object in the database. In this case, the vector quantification is based on local codebooks instead of a unique global codebook as the BF-SIFT method. We associate each 3D object with its codebook, which is learned from the features extracted from the 3D model. This is the fundamental difference between the two methods, but all other steps are similar.

After the construction of the visual codebook of each 3D object in the 3D models database, we calculate the corresponding descriptors using the codebook of each object its nearest neighbor in the codebook. This consists in finding the visual words appearance frequencies in the codebook for each 3D object. Thus, each object is associated with a histogram bins. By searching for the nearest neighbor in the codebook, a local descriptor is assigned to each 3D object.

After the vector quantification, the code vectors frequencies, also called “visual words”, are counted to create a histogram for each 3D model, whose number of bins equals the size of the codebook. The histogram is the features vector of the 3D model.

Finally, to compare two 3D objects, using our method, we calculate the descriptor of the 3D object query following the same steps described above and illustrated in Figure 1. We use the codebook of the 3D model with the query well be compared, to vector quantize, the feature vector of 3D object-query.

Using the descriptor, the 3D object query is compared with features vector of a 3D model in the database. The feature vector of the query is calculated every time an occurrence is found in order to be compared with a 3D model in the database.

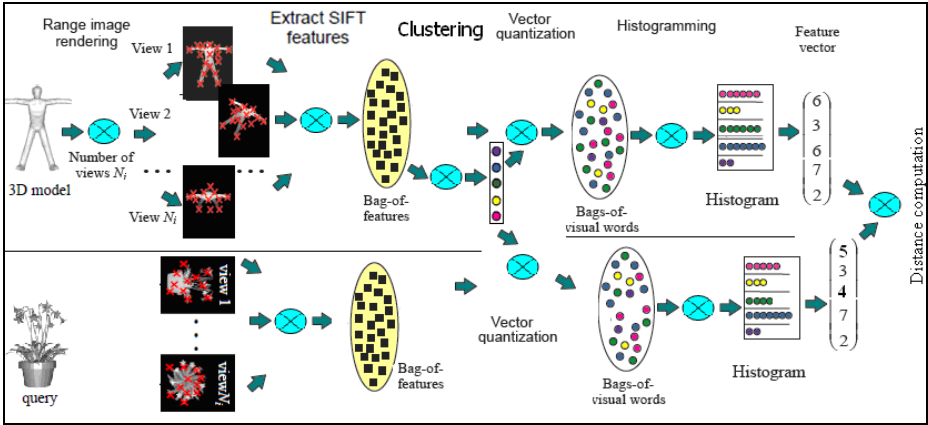


Fig. 1. Processing of comparison between a 3D object-query and a 3D model in database basing on local codebook of the 3D model

4 Experiments and Results

Our tests are made on the Princeton 3D Shape Benchmark database [3] with a set of different and rigid shapes. To fairly compare the retrieval effectiveness and performance of our improvement of the BF-SIFT method, we show the top 10 objects matched between the query models and the retrieved models, both for the original BF-SIFT original and our variant.

For both implementation (original BF-SIFT and our variant), we proceed as follows:

- To extract local feature from a depth-buffer view, the SIFT is implemented with the VLFeat MATLAB source of Veldaldi [19].
- To learn the codebook, we use the k-means MATLAB implementation, also in the VLFeat MATLAB source of Veldaldi [19], in order to cluster the set of local features by setting N_V to the size of vocabulary.
- For vector quantification, we use the MATLAB implementation of the linear k-nearest neighbor (KNN) search.

On figure 2, we report the top 6 objects. Experimental results show that as our method performs better than the BF-SIFT basing on global codebook but at the expense of more computational cost.

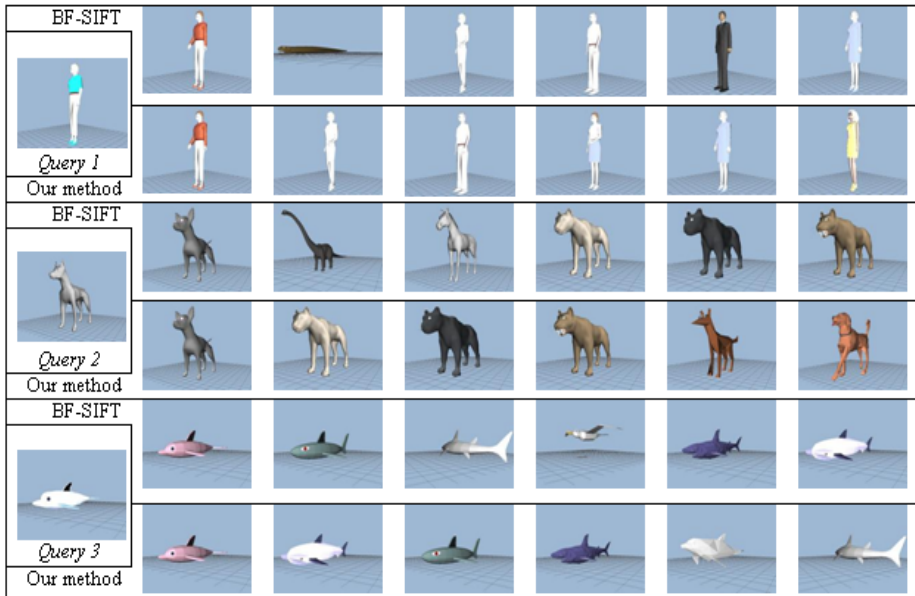


Fig. 2. The 6 top 3D objects retrieved from different “Class” query, using the BF-SIFT and our method

5 Conclusion and Perspectives

In this paper the principal object was to improving the retrieval performance of BF-SIFT. We presented an improvement of the 3D shape retrieval method using Bag-of-Features and SIFT. The key idea is to use local codebooks to vector quantization of salient local features, extracted from a given 3D object, basing on its associated codebook. We are compared the performance of the original BF-SIFT method with our improvement basing on the top-k result of the retrieval in Shape Benchmark database. Experimental results show that our method performs better than BF-SIFT.

References

1. Axenopoulos, A., Daras, P., Dutagaci, H., Furuya, T., Godil, A., Ohbuchi, R.: SHREC 2009 - Shape Retrieval Contest of Partial 3D Models. In: Eurographics Workshop on 3D Object Retrieval (2009)
2. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *Int'l Journal of Computer Vision* 60(2) (November 2004)
3. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The Princeton Shape Benchmark. In: *Proc. SMI 2004*, pp. 167–178 (2004), <http://shape.cs.princeton.edu/search.html>
4. Tangelder, J.W.H., Velkamp, R.C.: A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications* 39(3), 441–471 (2008)

5. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The Princeton shape benchmark. In: Shape Modeling and Applications Conference, SMI 2004, Genova, Italy, pp. 167–178. IEEE (June 2004)
6. Zaharia, T., Prêteux, F.: 3D versus 2D/3D shape descriptors: A comparative study. In: SPIE Conf. on Image Processing: Algorithms and Systems III - IS &T/SPIE Symposium on Electronic Imaging, Science and Technology 2003, San Jose, CA, vol. 5298 (January 2004)
7. Bustos, B., Keim, D.A., Schreck, T., Vranic, D.: An experimental comparison of feature-based 3D retrieval methods. In: 2nd Int. Symp. on 3D Data Processing, Visualization, and Transmission (3DPVT 2004), Thessaloniki, Greece (September 2004)
8. Del Bimbo, A., Pala, P.: Content-based retrieval of 3D models. *ACM Trans. Multimedia Com.*
9. Daras, P., Axenopoulos, A.: A 3D shape retrieval framework supporting multimodal queries. *International Journal of Computer Vision* (2010)
10. Chaouch, M., Verroust-Blondet, A.: A new descriptor for 2D depth image indexing and 3D model retrieval. In: Proc. ICIP 2007, vol. 6, pp. 373–376 (2007)
11. Lian, Z., Rosin, P.L., Sun, X.: Rectilinearity of 3D meshes. *International Journal of Computer Vision* 89, 130–151 (2010)
12. Lian, Z., Godil, A., Sun, X.: Visual Similarity based 3D Shape Retrieval Using Bag-of-Features. In: IEEE International Conference on Shape Modeling and Applications, SMI (2010)
13. Ohbuchi, R., Osada, K., Furuya, T., Banno, T.: Salient local visual features for shape-based 3D model retrieval, 93-102. In: Proc. IEEE Shape Modeling International, SMI (2008)
14. Tangelder, J., Veltkamp, R.C.: A Survey of Content Based 3D Shape Retrieval Methods. In: Proc. SMI 2004, pp. 145–156 (2004)
15. Iyer, M., Jayanti, S., Lou, K., Kalyanaraman, Y., Ramani, K.: Three Dimensional Shape Searching: State-of-the-art Review and Future Trends. *Computer Aided Design* 5(15), 509–530 (2005)
16. Veltkamp, R.C., et al.: SHREC 2006 3D Shape Retrieval Contest, Utrecht University Dept. Information and Computing Sciences Technical Report UU-CS-2006-030 (ISSN: 0924-3275)
17. Veltkamp, R.C., ter Harr, F.B.: SHREC 2007 3D Shape Retrieval Contest, Dept. of Info. and Comp. Sci., Utrecht University, Technical Report UU-CS-2007-015
18. Chaouch, M., Verroust-Blondet, A.: Alignment of 3D models. *Graphical Models* 71, 63–76 (2009)
19. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms, <http://www.vlfeat.org/>