A Bayesian 3D Search Engine using Adaptive Views Clustering

Tarik Filali Ansary, Mohamed Daoudi, Jean-Phillipe Vandeborre
FOX-MIIRE Research Group (LIFL UMR USTL-CNRS 8022)
(GET/INT - ENIC Telecom Lille 1)
email: {filali, daoudi, vandeborre}@enic.fr
http://www-rech.enic.fr/miire

***Abstract*— In this paper, we propose a method for 3D model indexing based on 2D views, named AVC (Adaptive Views Clustering). The goal of this method is to provide an "optimal" selection of 2D views from a 3D model, and a probabilistic Bayesian method for 3D model retrieval from these views. The characteristic views selection algorithm is based on an adaptive clustering algorithm and use statistical model distribution scores to select the optimal number of views. Starting from the fact that all views do not have equal importance, we also introduce a novel Bayesian approach to improve the retrieval. We finally present our results and compare our method to some state of the art 3D retrieval descriptors on the *Princeton 3D Shape Benchmark* database and a 3D CAD models database supplied by the car manufacturer *Renault*.**

I. Introduction

In recent years, many systems have been proposed for efficient information retrieval from digital collections of images and videos. However, the solutions proposed so far to support retrieval of such data are not always effective in application contexts where the information is intrinsically three-dimensional.

Antini et al [1] present an approach based on curvature correlograms. The main advantage of correlograms relates to their ability to encode not only the distribution of features but also their arrangement on the object surface.

In 3D retrieval using 2D views, the main idea is that two 3D models are similar, if they look similar from all viewing angles. Funkhouser et al. [2] apply view based similarity to implement a 2D sketch query interface. In the preprocessing stage, a descriptor of 3D model is obtained by 13 thumbnail images of boundary contour as seen from 13 view directions.

Filali et al. [3] propose an adaptive nearest neighbor like framework to choose the characteristic views of a 3D model. The framework gives good results but was experimented on a small database.

Chen et al. [4] defend the intuitive idea that two 3D models are similar if they also look similar from different angles. Therefore they use 100 orthogonal projections of an object and encode them by Zernike moments and Fourier descriptors. They also point out that they obtain better results than other well-known descriptors.

At last, for a further read on 3D retrieval state of the art, Tangelder and Veltkamp [5] present a complete survey on 3D shape retrieval.

In this paper, we propose a method for 3D model indexing based on 2D views, named AVC (Adaptive Views Clustering). This method aims at providing an optimal selection of 2D views from a 3D model, and a probabilistic Bayesian method for 3D models indexing from these views. This paper is organized in the following way. In section 2, we present the main principles of our method for characteristic views selection. In section 3, our probabilistic 3D models indexing is presented. Finally, the results obtained from two databases of 3D models are presented showing the performances of our method. We compare our method to some state of the art 3D retrieval descriptors on the *Princeton 3D Shape Benchmark* database and the SEMANTIC-3D database.

II. Selection of characteristic views

Let $D_b = \{M_1, M_2, \ldots, M_N\}$ be a collection of N three-dimensional models. We wish to represent each 3D model M_i by a set of 2D views that best represent it. To achieve this goal, we first generate an initial set of views from the 3D model, then we reduce it to the only views that best characterize the 3D model. This idea comes from the fact that all the views of 3D model do not have equal importance: there are views that contain more information than others.

A. Generating the initial set of views

To generate the initial set of views for a model M_i of the collection, we create 2D views (projections) from multiple viewpoints. These viewpoints are equally spaced on the unit sphere. In our current implementation, we use 320 initial views. To represent each of these 2D views, we use 49 coefficients of Zernike moment descriptor [6] [7]. Consequently to the use of Zernike moments, the approach is robust against translation, rotation and scaling.

B. Characteristic views selection

As every 2D view is represented by 49 Zernike moment coefficients, choosing a set of characteristic views that best caraterise the 3D models (320 views), is equivalent to choose a subset of points that represent a set of 320 points in 49 dimensions space. The problem of choosing X points that represent best a set of $N = 320$ point,is well known as *clustering* problem.

Data clustering is a well known problem for the mathematical and computer science communities, the literature in this domain is huge. One of the widely used method is K-means[8]. Its attractiveness lies in its simplicity and in its local-minimum convergence properties. It has, however, one main shortcomming. The number of clusters K has to be supplied by the user.

As we want from our method to adapt the number of characteristic views to the geometrical complexity of the 3D model, using K-means is not suited. To avoid this problem, we use a derivative from X-means[9]. Instead of a fixed number of clusters, we propose to use a range in which we will choose the best number of clusters. In our case the range will be $[1, \ldots, 40]$. During all this paper we assume that the maximum number of characteristic views is 40. It is a good compromise between speed, descriptor size and representation.

Algorithm 1 gives an overview of the characteristic views selection algorithm. Ror more details prefere to [**?**]

Algorithm 1 characteristic views selection algorithm.

Number of characteristic views = 1
while Number of characteristic views $<$ Maximum number characteristic views **do**
  Make global K-means on all the views (The start centers are the characteristic views).
  Save the characteristic views set and it's BIC Score[10].
  for all cluster of views **do**
    Make K-means (with K=2) on the cluster.
    Choose the representation with the higher BIC score. The original characteristic view or the two new characteristic views
    Update the number of characteristic views.
  end for
end while
Select the K and the characteristic view set with the higher BIC score.

III. Probabilistic approach for 3D indexing

Each model of the collection D_b is represented by a set of characteristic views $V = \{V_1, V_2, \ldots, V_C\}$, with C the number of characteristic views. To each characteristic view corresponds a set of represented views called V_r. Considering a 3D request model Q, we wish to find the model $M_i \in D_b$ which is the closest to the request model Q. This model is the one that has the highest probability $P(M_i/Q)$. Knowing that each model is represented by its characteristic views, $P(M_i/Q)$ can be written:

$$P(M_i|Q) = \sum\nolimits_{k=1}^{K} P(M_i|V_Q^k)P(V_Q^k|Q)$$

With K the number of characteristic views of the model Q. Let H be the set of all the possible hypotheses of correspondence between the request view V_Q^k and a model M_i, $H = \{h_1^k \vee h_2^k \vee \ldots \vee h_N^k\}$. A hypothesis h_p^k means that the view p of the model is the view request Vc_Q^k. The sign $\vee$ represents *logic or operator*. Let us note that if an hypothesis h_p^k is true, all the other hypotheses are false. $P(M_i|Vc_Q^k)$ can be expressed by $P(M_i|H^k)$. We have:

$$P(M_i|H^k) = \sum\nolimits_{j=1}^{N} P(M_i, V_{M_i}^j|h_j^k)$$

The sum $\sum_{j=1}^{N} P(M_i, V_{M_i}^j|h_j^k)$ can be reduced to the only true hypothesis $P(M_i, Vc_{M_i}^j|H_j^k)$. In fact, a characteristic view from the request model Q can match only one characteristic view from the model M_i . We choose the characteristic view with the maximum probability.

$$P(M_i|Q) = \sum\nolimits_{k=1}^{K} Max_j(P(M_i, V_{M_i}^j|h_j^k))P(V_Q^k|Q)$$

Using the Bayes theorem we obtain :

$$P(M_i|Q) = \sum\nolimits_{k=1}^{K}$$

$$Max_j\left(\frac{P(h_j^k|V_{M_i}^j, M_i)P(V_{M_i}^j|M_i)P(M_i)}{\sum_{i=1}^{N}\sum_{k=1}^{K} P(h_j^k|V_{M_i}^j, M_i)P(V_{M_i}^j|M_i)P(M_i)}\right)P(V_Q^k|Q)$$

With $P(M)$ the probability to observe the model M. $P(M_i) = \alpha e^{(-\alpha.|M_i|)/\sum_{i=N}^{i=1}|M_i|)}$. Where $|M_i|$ is the number of characteristic views of the model M_i. α is a parameter to hold the effect of the probability $P(M_i)$. The algorithm conception makes that, the complex is the geometry of the 3D model, the greater is the number of its characteristic views. Indeed, simple object (e.g. a cube) are more frequent and got more probability of appearance then complex ones. They can be at the root of more complex objects.

On the other hand $P(V_{M_i}^j|M_i) = 1 - \beta e^{(-\beta.N(Vr_{M_i}^j)/320)}$ Where $N(Vr_{M_i}^j)$ is the number of views represented by the characteristic view j of the model M. The greater is the number of represented views $N(Vr_{M_i}^j)$, the more the characteristic view $Vc_{M_i}^j$ is important and the best it represents the three-dimensional model. The β coefficient is introduced to reduce the effect of the view probability. We use the values $\alpha = \beta = 1/100$ which give the best results during our experiments.

The value $P(h_j^k|V_{M_i}^j, M_i)$ is the probability that, knowing that we observe the characteristic view j of the model M_i, this view is the k view of the 3D query model Q: $P(h_j^k|V_{M_i}^j, M_i) = 1 - D_{(Q^k, h_{V_{M_i}^j})}$. With $D_{h_q, h_{V_{M_i}^j}}$ the Euclidean distance between the 2D Zernike descriptors of Q and of the $Vc_{M_i}^j$ characteristic view of the three-dimensional model M_i.

IV. Experiments and results

A. Princeton Shape Benchmark

In our experiment, we computed the distances between all pairs of models in the *Princeton 3D Shape Benchmark* and analyze them with the *Princeton 3D Shape Benchmark* evaluation tools to quantify the matching performance with respect to the base classification.

As mentioned before, we use several different performance measures to objectively evaluate our method: the First Tier (FT), Second Tier (ST), Nearest Neighbor (NN), E-Measure, Discounted Cumulative Gain (DCG) and Normalized Discounted Cumulative Gain (N-DCG) match percentages, as well as the recall-precision plot [11].

Table I shows micro averages storage requirement (for our method, we used 23 views that is the average number of characteristic views for all the database models) and

Methods	Discrimination						
	Storage size	NN	FT	ST	E-Measure	DCG	N-DCG
LFD	4,700	65.7%	38.0%	48.7%	28.0%	64.3%	21.3%
AVC(probability)	**1,113**	**60.6%**	**33.2%**	**44.3%**	**25.5%**	**60.2%**	**13.48%**
REXT	17,416	60.2%	32.7%	43.2%	25.4%	60.1%	13.3%
GEDT	32,776	60.3%	31.3%	40.7%	23.7%	58.4%	10.2%
AVC(simple distance)	**1,113**	**58.2%**	**31.1%**	**42.7%**	**25.1%**	**59.9%**	**11,8%**
2-GR	512	55.5%	28.7%	39.1%	23.0%	56.3%	—%
EXT	552	54.9%	28.6%	37.9%	21.9%	56.2%	6.0%
SECSHEL	32,776	54.6%	26.7%	35.0%	20.9%	54.5%	2.8%
VOXEL	32,776	54.0%	26.7%	35.3%	20.7%	54.3%	2.4%
SECTORS	552	50.4%	24.9%	33.4%	19.8%	52.9%	-0.3%
CEGI	2,056	42.0%	21.1%	28.7%	17.0%	47.9%	-9.6%
EGI	1,032	37.7%	19.7%	27.7%	16.5%	47.2%	-10.9%
D2	136	31.1%	15.8%	23.5%	13.9%	43.4%	-18.2%
SHELLS	136	22.7%	11.1%	17.3%	10.2%	38.6%	-27.3%

TABLE I

RETRIEVAL PERFORMANCES FOR PRINCETON SHAPE BENCHMARK

retrieval statistics for each algorithm. Storage size is given in bytes. We found that micro and macro-average results gave consistent results, and we decided to present micro-averaged statistics.

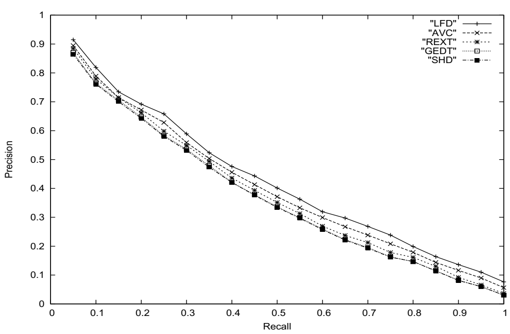


Fig. 1. Recall Precision on *Princeton 3D Shape Benchmark* database.

Figure 1 shows the recall precision plots for our method AVC and some other shape descriptors. We presented the curves of the most relevant algorithm only, to keep the figure clear.

We find that the shape descriptors based on 2D views (LFD and AVC) provides the best retrieval precision in this experiment. We might expect shape descriptors that capture 3D geometric relationships would be more discriminating than the ones based solely on 2D projections, the opposite is true.

We can notice that our method provides more accurate results with the use of Bayesian probabilistic indexing. The experiment shows that AVC gives better performances then 3D harmonics, Radialized Spherical Extent Function and Gaussian Euclidean Distance Transform on the *Princeton 3D Shape Benchmark* database. Light Field Descriptor gives better results than our method but uses 100 views, does not adapt the number of views to the geometrical complexity and uses two descriptors for each view (Zernike moments and Fourier descriptor), which make it slower and more memory consuming descriptor compared to the method we presented.

Overall, we can conclude that AVC gives a good compromise between quality (relevance) / cost (memory and on-line comparison time) between the shape descriptors we compared to using the *Princeton 3D Shape Benchmark*.

B. *SEMANTIC-3D Database*

The experiments are made on a database that contains 5000 3D models. To objectively evaluate the performance of our method on this database, a classification was made (ground truth). 758 models are classified on 75 classes. A ”special” class called others contain all the 3D models that was not classified. Figure 2 shows the recall precision plots for our method AVC on the SEMANTIC-3D database. We can explain the good results of our method by the fact that the variance intra-class are very small. As the database contains real professional 3D CAD Models, the 3D models from the same class represent real mechanical parts used in a car. The 3D models from the same class represent different versions of the same models with small changes.

Methods	Discrimination				
	NN	FT	ST	E-Measure	DCG
probability	**99.2%**	**88.3%**	**96.7%**	**52%**	**96.3%**
simple distance	**98.1%**	**86.8%**	**95.1%**	**51.3%**	**93.9%**

TABLE II

RETRIEVAL PERFORMANCES FOR SEMANTIC-3D DATABASE

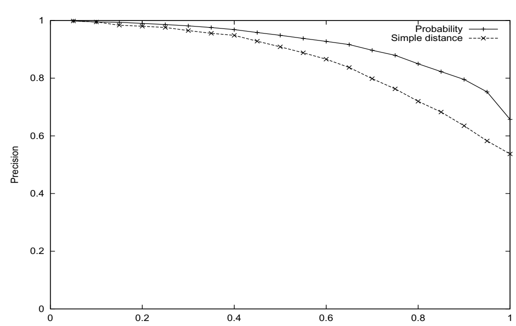


Fig. 2. Recall Precision for Semantic-3D database.

In the other hand, 3D CAD models in the database contains holes so that they can be fixed to other mechanical parts. The positions and the dimension of the holes can differentiate between two different models from the same class. As we represent each view of the 3D model by Zernike moments, the holes and the global shape are well taken into account. We can also notice the result enhancement when we use the probabilistic approach for retrieval.

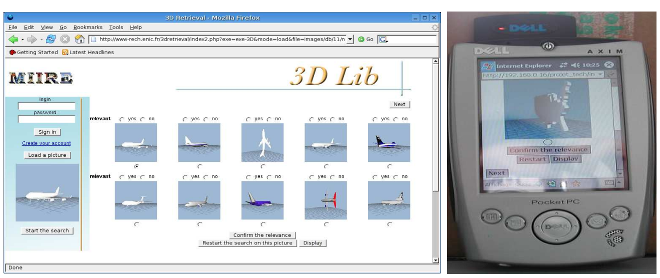
C. On-line Search Engine

The SEMANTIC-3D project focuses on the development of tools and methods required to implement new operational services for retrieving 3D content through the Web and communicating objects. Information and communication system must be available for remote access and assistance, interconnecting originators (mechanical part designers), nomadic users (automotive industry technicians) and a central 3D data server.

To experiment our algorithms and to asset the results presented in the previous sections, we developed an on-line 3D search engine. Our search engine can be reached from any device having compatible web browser (PC, PDA, SmartPhone, etc.) [12].

Figure 3(a) shows the interface of our system using a PC web browser. Figure 3(b) shows the interface of our system on PDA (Pocket PC under Windows Pocket 2003).

Depending on the web access device he/she is using, the user faces two different kind of web interfaces : a rich web interface for full-featured web browsers (figure 15(a)), and a simpler interface for PDA web browsers (figure 15(b)). In both cases, the results returned by the 3D search engine are the same. The only difference lies in the design of the results presentation. At last, the users should notice that, due to some copyright protection, the 3D CAD models from the SEMANTIC-3D project are not available for public on-line use. The 3D database available for tests of our 3D search engine is the **Princeton Shape Benchmark** Database [13].



(a) PC browser interface

(b) PDA browser interface

Fig. 3. The PC and PDA browser interface of our on-line search engine.

V. Discussion and Conclusion

In this paper, we propose a 3D model retrieval system based on characteristic views similarity called AVC (Adaptive Views Clustering). Starting from the fact that the more the 3D model is geometrically complex, the more its 2D views are different, we propose a characteristic views selection algorithm that corresponds the number of views to its geometrical complexity. Starting from 320 initial views, our algorithm select the ”optimal” characteristic views set that best represent the 3D model. The number of characteristic views varies from 1 to 40. We also propose a new probabilistic retrieval approach that takes into account that not all the views of 3D models have the same importance, and also the fact that geometrically simple models have more probability to be relevant than more complex ones. Based on some standard measures, we compared our method to twelve state of the art methods on *Princeton 3D Shape Benchmark* database. Our method gives the second best results. The AVC method we proposed gives a good quality/cost compromise compared to other well-known methods. The good results of our method on a large 3D CAD models database (5000 models) supplied by *Renault*, show that our method can also be suitable for 3D CAD models retrieval. Our method is robust against noise and model degeneracy. It can be suitable against topologically ill-defined 3D models. A practical 3D models retrieval system based on our approach is available on the web for on-line tests [12].

VI. Acknowledgments

This work is supported by the French Research Ministry and the RNRT (Réseau National de Recherche en Télécommunications) within the framework of the SEMANTIC-3D National Project (http://www.semantic-3d.net). We would like to thank Professor Dan Pelleg from Carnegie Mellon University for the help he provided on X-means.

References

[1] G. Antini, S. Berretti, A. D. Bimbo, and P. Pala, “Retrieval of 3d objects using curvature correlograms,” in *ICME*, july 2005.
[2] T. Funkhouser, P. Min, M. Kazhdan, A. Haderman, D. Dobkin, and D. Jacobs, “A search engine for 3D models,” *ACM Transactions on Graphics*, vol. 22, no. 1, pp. 83–105, 2003.
[3] T. F. Ansary, J. P. Vandeborre, S. Mahmoudi, and M. Daoudi, “A bayesian framework for 3D models retrieval based on characteristic views,” in *2nd IEEE International Symposium on 3D Data Processing, Visualization and Transmission*, September 2004.
[4] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung, “On visual similarity based 3D model retrieval,” in *Eurographics*, vol. 22, 2003, pp. 223–232.
[5] J. Tangelder and R. Veltkamp, “A survey of content based 3D shape retrieval methods,” in *Shape Modeling International*, 2004, pp. 145–156.
[6] A. Khotanzad and Y. Hong, “Invariant image recognition by Zernike moments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489 – 497, May 90.
[7] W. Kim and Y. Kim, “A region-based shape descriptor using Zernike moments,” *Signal Processing: Image Communication*, vol. 16, pp. 95–100, 2000.
[8] R. Duda and P. Hart, “Pattern classification and scene analysis,” *John Wiley and Sons*, 1973.
[9] 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, pp. 727–734.
[10] G. Schwarz, “Estimating the dimension of a model,” *The Annals of Statistics*, vol. 6, pp. 461–464, 1978.
[11] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, “The princeton shape benchmark,” in *Shape Modeling International*, 2004.
[12] “Miire - 3d search engine,” in *http://www-rech.enic.fr/3dretrieval*.
[13] “Princeton shape benchmark (2004),” in *http://shape.cs.princeton.edu/benchmark*.