An Efficient Mechanism for 3D Model Retrieval*

Liang Ji[1,2], Gangshan Wu[1,2], Wenjun Dai[1,2]
[1]*State Key Laboratory for Novel Software Technology, Nanjing China 210093*
[2]*Department of Computer Science & Technology, Nanjing University, Nanjing China 210093*
jiliang@graphics.nju.edu.cn

Abstract

Shape matching is one of the crucial problems in 3D model retrieval system. A usual approach is to exhaustively search through the database comparing each database model with the query model. This approach is inefficient especially for a large database. In this paper, an efficient mechanism for 3D model retrieval is proposed. In the preprocessing stage, a set of reference models are selected from the database using cluster analysis, and distances between database models and reference models are computed and stored. Then in the query stage, for a certain query model, searching is accelerated by reducing the large amount of model comparisons using triangle inequality based on the reference distances computed before. The proposed retrieval mechanism is implemented and experiment result shows that retrieval efficiency is greatly improved than the usual approach without any precision loss.

1. Introduction

With recent developments in 3D data acquisition techniques and modeling methods, construction of 3D models becomes much easier. This has led to an increasing accumulation of 3D models, both on the Internet and the other. Therefore, the need for the ability to retrieve models from large databases has gained prominence. Nowadays, many experimental 3D model retrieval systems have been designed and implemented. Usually, they exhaustively search through their database comparing each database model with the query model. Obviously, this commonly used approach is inefficient especially for a large model database because of a huge amount of model comparisons. M Ankerst, G Kastenmuller, H Kriegel, et al.[1] implement a retrieval system for the Brookhaven Protein Data Bank on HP C160 workstations, and they cost about 1.42 seconds to search a single model. J Pu, Y Liu, Y Gu, et al.[2] implement a retrieval system containing about 2700 models, and they cost 1.7~6.22 seconds to search a singe model. D Chen, X Tian, Y Shen, et al.[3] implement a retrieval system based on visual similarity in a server with Pentium IV 2.4GHz CPU, and they cost about 2 seconds to search a single model. M Hilaga, Y Shinagawa, T Kohmura, T Kunii[4] implement a retrieval system containing 230 models on a PC with Pentium II 400MHz CPU, and their search time reaches amazing 12 seconds.

Usually, a practical 3D model retrieval system should be suitable for interactive querying. So it is always a challenge that how to reduce the amount of model comparisons and accelerate the retrieval. In this paper, an efficient mechanism for 3D model retrieval is proposed. In the preprocessing stage, a set of reference models are selected from the database using cluster analysis, and distances between database models and reference models are computed and stored. Then in the query stage, for a certain query model, searching is accelerated by reducing the large amount of model comparisons using triangle inequality based on the reference distances computed before.

We introduce the framework of our 3D model retrieval system in section 2. In section 3, we explain the application of triangle inequality to reduce the large amount of shape comparisons. In section 4, detailed algorithm to accelerate 3D models retrieval is proposed. The selection of reference models is a key problem, which is discussed in section 5. Experiment result and conclusion are presented in section 6.

2. Framework of our 3D retrieval system

The framework of our 3D model retrieval system consists of a preprocessing stage offline and a query engine online, as illustrated in Figure 1. In the

* Supported by the National Natural Science Foundation of China under Grant No.60533080.

Proceedings of the 16th International Conference on
Artificial Reality and Telexistence--Workshops (ICAT'06)
0-7695-2754-X/06 $20.00 © 2006 IEEE

preprocessing stage offline, each 3D model first has to be identified with a shape descriptor, providing a compact overall description of the shape. Our retrieval system uses the shape descriptor of model's volume distribution proposed by Dai Wenjun, Wu Gangshan, Zhang Fuyan[5]. Then a set of reference models are selected from the database using cluster analysis, and reference distances between database models and reference models are computed and stored. In the query stage online, the query engine computes the query model's descriptor, then use the shape matching module to retrieve models similar to the query. Shape matching is a key module, in which we use the triangle inequality to reduce the large amount of model comparisons based on the reference distances computed before. Therefore, the searching process is accelerated and the retrieval efficiency is greatly improved.

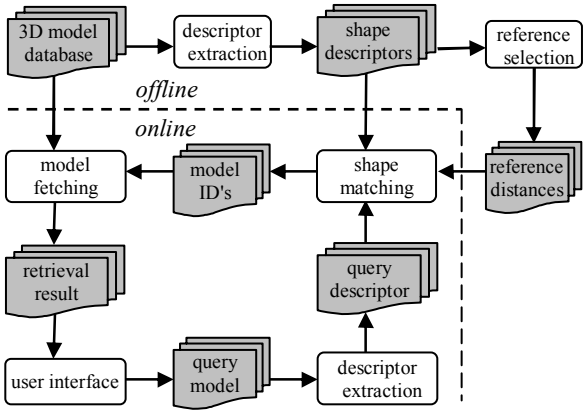

Figure 1. Framework of our retrieval system

3. Principle to reduce shape comparisons

Shape descriptors are compared with each other to decide how similar two models are. Usually, a dissimilarity measure, but not a similarity measure, is used to compute the distance between pair of descriptors indicating the degree of their resemblance. J. W. Tangelder and R. C. Veltkamp[6] has discussed that an excellent dissimilarity measure should obey several properties. In [6], a dissimilarity measure on a set S is formalized to a non-negative valued function $D: S \times S \to R^+ \bigcup \{0\}$. And if function D obeys the property:

$$\forall x, y, z \in S, D(x,z) \le D(x,y) + D(y,z)$$

the corresponding dissimilarity measure obeys the triangle inequality. Experiment in section 6 uses simple L_1 distance to measure the dissimilarity of model's volume distribution descriptors[5]. And it is easy to prove that L_1 distance obeys the triangle inequality.

When a dissimilarity measure obeying triangle inequality is used to compute the distance between pair of shape descriptors, the large amount of computation can be greatly reduced, and the searching can be accelerated. Suppose that we want to find the best match for a query model q from a database X containing n models. We then define the reference model $r \in X$. If a model x in database X satisfies the following inequality:

$$|D(x,r) - D(q,r)| \ge \omega \qquad (1)$$

we are sure that model x is not the best match for query model q and can be safely eliminated from further consideration[7]. That is, $D(x,q)$ needn't to be computed. In inequality (1), $D(x,r)$ is the distance between each shape descriptor in the database and descriptor of the reference model r, which can be computed in the preprocessing stage beforehand. Now we suppose that a query model q is submitted in the query stage. Well then, only with the computing of $D(r,q)$ (the distance between q and the reference model r), a large number of models in database X dissimilar to the query model q can be excluded from further consideration beforehand according to the inequality (1). Therefore, the amount of computing for $D(x,q)$ is greatly reduced, and the searching is accelerated. For further proving, we can refer to Figure 2, and consider for simplification a dissimilarity measure using Euclidean Distance based on 2D shape descriptors. The positions of the reference model r and the query model q are illustrated in Figure 2, and $D(q,x_1) = D(q,x_2) = \omega$. According to inequality (1), $D(q,r) - D(x,r) \ge \omega$ excludes database models x lying inside the inner circle; while $D(x,r) - D(q,r) \ge \omega$ excludes database models x lying outside the outer circle. Therefore, none but models lying between the inner circle and the outer circle remain to be compared with the query factually.

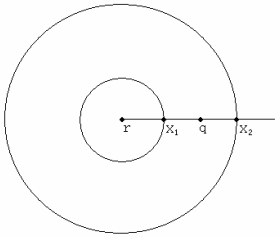

Figure 2. Use triangle inequality to exclude some models

Proceedings of the 16th International Conference on
Artificial Reality and Telexistence--Workshops (ICAT'06)
0-7695-2754-X/06 $20.00 © 2006 IEEE

4. Algorithm to accelerate 3D retrieval

In section 3, we discuss how to reduce descriptor comparisons using only one reference model. As a matter of fact, further reduction of descriptor comparisons can be achieved by using multiple reference models[7]. The problem of how to select reasonable reference models from the database is discussed in section 5. Detailing below, the algorithm for only one reference model is improved to be suitable for a fine selection of multiple reference models.

In a 3D model retrieval system, the retrieval result for a certain query is usually a set of database models, but not only one model. So the algorithm to find one best match for a query should be improved. In detail, we maintain a result queue γ with the length of m, and ensure that models in queue γ are always the first m best match models which are well sorted by their similarity to the given query.

We propose an efficient 3D models retrieval algorithm using multiple reference models based on triangle inequality. Our algorithm is divided into two stages: preprocessing and query. The preprocessing stage is done once before any query is processed; while the steps of the query stage are done for each query. So the final retrieval time does not contain the time of preprocessing. Detailed algorithm is as follows:

INPUT: model database $X = \{x_1 x_2, \cdots, x_n\}$, and a query model q

OUTPUT: result queue γ consists of the first m best match models (well sorted by similarity to q)

ALGORITHM:

Preprocessing:

1. Select s reference models to build a set $R = \{r_1, r_2, \cdots, r_s\}$ using algorithm 2 in section 5.
2. Compute distances between each shape descriptor in the database and each descriptor of reference models: $\forall x \in X, \forall r \in R, computeD(r, x)$.

Query:

1. Compute distances between each descriptor of reference models and the descriptor of query model: $\forall r \in R, computeD(r, q)$.
2. Maintain a result queue γ with the length of m, and ensure that models in queue γ are well sorted by their distances to query model q. New coming models will be inserted into queue γ at a suitable position according the sorting rule and if queue γ becomes full the model with maximum distance will be eliminated.
3. Insert the first m database models into queue γ, and compute distances between descriptors of them and the descriptor of query model q, then the maximum value of the distances is sent to ω.
4. Go through all the database models x, if $\exists r \in R, |D(x,r) - D(q,r)| \geq \omega$, the model x will be excluded from comparison with the query; otherwise, compute the distance between the model and the query $D(q,x)$, insert the model x into queue γ, and ω is updated to the maximum value of the distances in queue γ.

5. Reference Models Selection

Which models should be selected as reference? We can do something to make the selection more reasonable. The models selected as reference should be uniformly distributed in the descriptor's vector space. That is, we should avoid that two or more reference models lie too close to each other in the descriptor's vector space. The reason is that two reference models with little shape distance will exclude almost the same database models for a query, so one of the two references has not worked, and is "wasted". We select reference models from database using the approach of cluster analysis[8]. In the process all the database models are partitioned into s clusters. Models inside a single cluster are similar to each other, while those belong to different clusters are dissimilar to each other. Then we select the s models nearest to each cluster's center as reference. The factual amount of the clusters, i.e. the amount of reference models, is also very important. The more references, the more database models will be excluded from comparing with the query. However, the cost for excluding itself is increasing along with the reference models, so the retrieval time for a query will not decrease all the time. Therefore, the reasonable selection of reference models is not "the more, the better", but needs a compromise. In section 6, the amount of reference models s is decided by experiment.

We first consider the popular k-means algorithm for cluster analysis. In the beginning, k models are selected from the database randomly as the initial cluster centers. Then the algorithm is represented as an iterative process below: distances between each database model and each cluster center are computed, based on which all database models are partitioned into clusters nearest to them. Afterwards each cluster center is recomputed. The process above is iterative until each database model no longer flows between clusters and the system trends to be stable. Based on the shape

Proceedings of the 16th International Conference on
Artificial Reality and Telexistence--Workshops (ICAT'06)
0-7695-2754-X/06 $20.00 © 2006 IEEE

descriptor of model's volume distribution[5], we implement the k-means algorithm for a database containing 10911 models[9]. During hundreds of iterations in several days, the running program does not converge, and then is given up. Therefore, the k-means algorithm is not applicable to cluster analysis based on shape descriptor of model's volume distribution.

We then attempt another method of agglomerative hierarchical clustering, also called the *bottom-up* approach. It starts with each model forming a separate cluster and then merges these clusters close to each other iteratively, until a termination condition holds. To decide which two clusters should be merged, we need a measure for similarity between clusters. Generally speaking, four widely used methods to compute distances between clusters are as follows:[8]

1. *Single linkage method:*

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} \|x - y\|$$

2. *Complete linkage method:*

$$d(C_i, C_j) = \max_{x \in C_i, y \in C_j} \|x - y\|$$

3. *Average linkage method:*

$$d(C_i, C_j) = \frac{1}{n_i n_j} \sum_{x \in C_i} \sum_{y \in C_j} \|x - y\|$$

thereinto n_i and n_j are separately the amount of members in cluster C_i and C_j.

4. *Centroid linkage method:*

$$d(C_i, C_j) = \|m_i - m_j\|$$

thereinto m_i and m_j are separately the centroid of cluster C_i and C_j.

In the four methods above, (1) and (2) are easy to implement but seldom applicable to practice, due to lack of consideration of the clusters' constructional information. (3) takes into consideration the clusters' construction, and is practical to some extent. (4) is a compromise between (1) and (2). It works stably and robustly without sensibility to noise, and is selected to measure the similarity between clusters in our algorithm. Agglomerative hierarchical clustering algorithm based on centroid linkage is implemented for a database containing 10911 models[9]. The result shows that the amounts of models in different clusters vary from one to thousands non-uniformly. So the clustering result goes against the selection of reference models. Therefore, the algorithm needs to be improved.

To uniformly partition the models into clusters, the algorithm for clustering has to facilitate the merging of clusters containing fewer models. So we consider increasing the distances between clusters containing more models artificially. Our solution is appending a coefficient $n_i \times n_j$ to the centroid linkage, while n_i and n_j are separately the amount of members in C_i and C_j. Then the centroid linkage is improved to:

$$d(C_i, C_j) = n_i \times n_j \|m_i - m_j\| \quad (2)$$

New result shows that improved centroid linkage promotes the performance of the agglomerative hierarchical clustering. Models are uniformly partitioned into clusters, while the property of "similar in clusters and dissimilar between clusters" is not destroyed. In conclusion, our algorithm for selection of reference models based on agglomerative hierarchical clustering is as follows:

INPUT: model database $X = \{x_1, x_2, \cdots, x_N\}$

OUTPUT: a set containing n reference models $R = \{r_1, r_2, \cdots, r_n\}$

ALGORITHM:

1. Let N models form N separate clusters: $C = \{C_1, C_2, \cdots, C_N\}$, $C_i = \{x_i\}$, $i \in I, I = \{i | 1, 2, \cdots, N\}$.
2. Find in the set $\{C_i | i \in I\}$ two clusters C_s and C_t satisfying $d(C_s, C_t) = \min_{\forall i, j \in I} d(C_i, C_j)$, thereinto $d(C_i, C_j)$ is the centroid linkage defined as equation (2).
3. Merge C_t and C_s to C_s, and then delete C_t.
4. Delete t from I, and if the cardinal number of I is more than n, turn to step 2.
5. Compute centroids of each cluster in the set $\{C_i | i \in I\}$, and then select the models $r_i | i \in I$ nearest to each centroid to form the reference set: $R = \{r_1, r_2, \cdots, r_n\}$

6. Experiment Result and Conclusion

The algorithms above are implemented based on shape descriptor of models' volume distribution[5] in a PC with Pentium IV 2.8GHz CPU, using a database of 10911 models[9]. In our experiment, the amount of reference models is given the values of 1、5、10、50、100、200、300 to decide that with which value the system performs best. For each value, we select the set of reference models using agglomerative hierarchical clustering. Then we use the 10911 database models as queries separately, and compute the average retrieval time for each query and the percentages (high, low,

Proceedings of the 16th International Conference on
Artificial Reality and Telexistence--Workshops (ICAT'06)
0-7695-2754-X/06 $20.00 © 2006 IEEE

average) of database models factually compared with each query. The experiment result is illustrated in table 1 below.

Table 1. Experiment result of our efficient retrieval system

amount of reference models	percentage of database models factually compared(%)			average retrieval time(ms)
	High	low	average	
1	99.99	5.31	85.95	291
5	99.83	4.07	56.21	193
10	99.17	3.78	47.18	163
50	91.00	2.71	31.52	118
100	88.18	3.98	28.13	112
200	84.18	3.82	24.86	114
300	80.79	3.82	22.62	115
500	76.29	3.78	19.98	136

Result shows that: the more references, the more database models are excluded but the average retrieval time does not decrease all the time because of increasing cost for excluding itself. Finally, we gain the conclusion that compromise of 100 reference models can reach an optimization with least average retrieval time. The retrieval efficiency of our system is 3.14 times improved than the usual approach without any precision loss.

Our retrieval system based on shape descriptor of model's volume distribution[5] is implemented and its performance is improved greatly using the mechanism proposed in this paper. The system is available on the Web for practical trial use in the site: *http://dmcu.nju.edu.cn:8080//3DR*.

Acknowledgments. Our research is funded by Digital Museum of Chinese University and China Digital Science Technology Museum.

References

[1] M Ankerst, G Kastenmuller, H Kriegel, et al. 3D Shape Histograms for Similarity Search and Classification in Spatial Databases. Proc. of 6th International Symposium on Advances in Spatial Databases, Hong Kong, China, 1999, 207-228.

[2] J Pu, Y Liu, Y Gu, et al. 3D Model Retrieval Based on 2D Slice Similarity Measurements, Proceedings of the Second International Symposium on 3D Data Processing, Visualization, and Transmission , Thessaloniki, Greece, 2004, 95-101.

[3] D Chen, X Tian, Y Shen, et al. On Visual Similarity Based 3D Model Retrieval, Computer Graphics Forum, 2003, 22(3): 223-232.

[4] M Hilaga,Y Shinagawa, T Kohmura , T Kunii. Topology Matching for Fully AutomaticSimilarity Estimation of 3D Shapes, ACM SIGGRAPH'2001, 2001, 203-212.

[5] Dai Wenjun, Wu Gangshan, Zhang Fuyan, 3D Model Retrieval Based on Volume Distribution, ICCMSE 2005 （International Conference of Computational Methods in Sciences and Engineering） Loutraki, Greece, 21-26 October,2005.

[6] J. W. Tangelder and R. C. Veltkamp. A survey of content based 3D shape retrieval methods. In Proc. Shape Modeling International, pages 145--156, Genoa, Italy, June 2004.

[7] J. Barros, J. French, W. Martin, P. Kelly, and M. Cannon. Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. In Proc. of SPIE, volume 267, pages 392–403, 1996.

[8] Jiawei Han ,Micheline Kamber .Data Mining, Higher Education Press.2001.

[9] 3D model retrieval system, http://3d.csie.ntu.edu.tw/.