Proceedings of the TMCE 2004, April 13-17, 2004, Lausanne, Switzerland, Edited by Horváth and Xirouchakis

SUPPORTING EFFECTIVE AND EFFICIENT THREE-DIMENSIONAL SHAPE RETRIEVAL

Kuiyang Lou
Purdue Research and Education Center of Information Systems in Engineering (PRECISE)
School of Mechanical Engineering
Purdue University, West Lafayette
U.S.A.
kuilou@purdue.edu

Natraj Iyer, Subramaniam Jayanti, Yagnanarayanan Kalyanaraman, Karthik Ramani
Purdue Research and Education Center of Information Systems in Engineering (PRECISE)
School of Mechanical Engineering
Purdue University, West Lafayette
U.S.A.
{iyern, jayantis, yagna, ramani}@purdue.edu

Sunil Prabhakar
Department of Computer Science
Purdue University, West Lafayette
U.S.A.
sunil@cs.purdue.edu

ABSTRACT

In this paper, we discuss the search effectiveness and efficiency of content-based 3D shape search. The search effectiveness is characterized by the precision and recall of results; and the search efficiency is evaluated by the ratio of the number of the visited nodes in a search operation to the number of the nodes of an index. A real database of 113 3D shapes and synthetic databases created by random number generator are employed for the investigation. Four feature vectors are extracted to represent 3D shapes. The descending order of search effectiveness of using feature vectors are: principal moments, moment invariants, geometric parameters, and eigenvalues. We evaluated the performance of an R-tree based index to improve the search efficiency. The results show that R-tree index significantly improves the efficiency of both real and synthetic databases. The search efficiency decreases when the dimensionality of data record and the capacity of database node increase. Based on the experiments of synthetic database, the efficiency is stable when the size of a database is relative large. This means that the efficiency is scalable to the size of database. A multi-step refinement is proposed to improve search effectiveness. Based on the results of twenty six experiments with real database, the effectiveness of using multi-step refinement is 51% higher than that of one-shot search using principal moment as feature vector, although the difference becomes smaller when the number of retrieved shapes becomes larger.

KEYWORDS

Three-dimensional shape search, Effectiveness, Efficiency, Content-based system, Computer-aided design

1. INTRODUCTION

The benefits of computer-aided design systems have resulted in the proliferation of 3D CAD models. Searching and reusing CAD data has been considered critical for competitive engineering of product development (Lu, et al. 1997, Ramesh, et al. 2001, Cicirello, et al. 2001, Elinson, et al. 1995, Iyer et al. 2003). However, because of the complexity of 3D CAD data, current information technology, which mainly focuses on text or 2D image information, is not capable retrieve reusable CAD data. We thus focus our research in this paper the effectiveness and efficiency of 3D engineering shape search. Besides the potential in mechanical engineering, 3D shape search has also attracted substantial research interest in other important areas, such as proteomics and medical image processing (Kelley L. A. et al. 2000, Kastenmuller G. et al. 1998, Thiran J. –P. et al. 2000).

3D shapes are complex data that are impractical to search directly due to time or space complexity. Content-based search (CBS) systems are commonly used for 3D shape search instead of direct searching 3D shapes. In a content-based 3D shape search system, 3D models are represented by feature vectors, which serve as the signatures of shape content. Feature vector, a term commonly used in CBS systems, is a set of numbers that capture one or more characteristics of 3D shapes. Thus searching similar shapes is translated to searching similar feature vectors.

Based on the representation of 3D shapes, prior search systems can be classified into those based on moment invariants, Fourier coefficients, shape histogram, shape distribution, spherical harmonics, aspect graph, and graph. The use of moments for shape description was initiated by Hu, M. K. (1962) for 2D visual pattern reorganization. Sadjadi et al. (1980) extended the use of moment in shape search to 3D applications. Vranic et al. (2001) used the Fourier coefficient of discrete Fourier transformation to represent as a 3D shape description. Ankerst et al. (1999) used shape histograms that are based on complete and disjointed partitioning of the space in which the objects reside into cells. Osada et al. (2001) proposed shape distribution consisting of the probability distribution of geometric properties of the randomly sampled surface points. Kazhdan et al. (2002) investigated decomposing 3D models into a collection of functions defined on concentric spheres and using spherical harmonics as shape signature. Cyr et al. (2001) used aspect graph to match 3D objects by 2D views. El-Mehalawi et al. (2003) studied to the use of graphs extracted from the B-rep representation of 3D models for shape search. Iyer et al. (2003) has a detailed review about the 3D shape representation and search systems.

Although different feature vectors have been proposed in 3D shape search, there is no systematic work to evaluate the effectiveness of using different feature vectors in searching 3D shapes. The tests of different research groups were conducted on different data sets. Thus the results are not comparable. Search effectiveness is usually illustrated with a few examples, which do not objectively reflect the overall performance of the search system. Search efficiency, which is critical for the scalability of a search system, has rarely been explored. Furthermore, a one-shot search approach is commonly employed for shape search. However, because of the difficulties of representing 3D shape, the subjectivity of similarity, and the difference between the similarity degree computed by the system and that perceived by a user, it is difficult for one-shot approach to achieve satisfactory search effectiveness.

In this paper, we develop a benchmark database to compare the search effectiveness based on different feature vectors. We test the search efficiency of R-tree index of the system using both real and synthetic databases. In addition, we propose a multi-step search strategy and have tested it in order to improve search effectiveness.

The rest of the paper is organized as follows. Section 2 describes the feature vectors employed in the paper. Section 3 explains the similarity measure. Section 4 is devoted to the search effectiveness. Section 5 presents the work on search efficiency. Finally, Section 6 concludes the paper and proposes future work.

2. FEATURE VECTORS (FV)

Feature vectors are extracted to representing 3D models by the process termed as feature transformation. Through feature transformation, a 3D model is represented by a vector of real number, which can be further considered as the coordinates of the point in feature space. The spatial relationship between two points is employed to quantify the degree of similarity between two models in the paper. Figure 1 shows the feature transformation. A 2D coordinate system is used to illustrate the point representation of models in the feature space.

Since the similarity between two shapes is mapped to that of feature vectors, feature vectors are critical to the effectiveness of search systems. In this paper, we extract four types of feature vectors and compare the search effectiveness based on them, respectively. The four feature vectors are moment invariants, geometric parameters, principal moments, and eigenvalues, respectively. Our experiments and methods to evaluate the search effectiveness of feature vectors are introduced in Section 4.

3D shapes are normalized, voxelized (Cybenko, 1997), and skeletonized (Palagyi, 1998) in order to obtain the feature vectors. Normalization is the process of transformation of a 3D model into a standardized or canonical form that retains relevant geometrical information of the original model. Skeletonization is the process to convert 3D shape representation into a set of volume elements. Skeletonization is the process to extract the skeleton of the voxel model (Cybenko, G. et al. 1997, Iyer, N. et al. 2003). Based on the skeletal model, we can construct skeletal graph, which is small in size and insensitive to the trivial changes of a 3D model. The graph structure constructed from B-rep representation of 3D models has been proposed for shape search (El-Mehalawi et al. 2003). However, this type of graph is much big and sensitive. A small change of the 3D model could significant changes the graph structure.

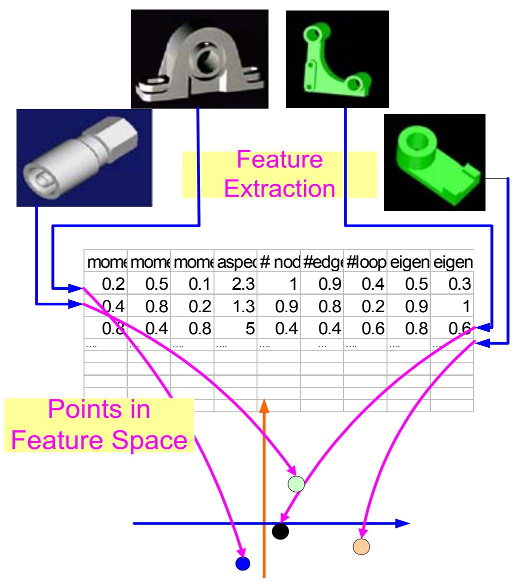

Figure 1 Shape representations along transformations

2.1. Moment Invariants

Moment invariants are derived from second order moments and are invariant to translation, scaling, and orientation (Cohen D. K. 1996). The invariant property is realized through the following steps: (1) Aligning the centroid of the shape to the origin of a coordinate system; compute the central moment (μ_{lmn}) that is invariant to translation; (2) Computing the volume of the shape, eliminate scaling impact on the central moments by dividing them by a volume term, the expression $I_{lmn} = \mu_{lmn} / \mu_{000}{}^{5/3}$ is scaling invariant (μ_{000} is the volume of the model), and (3) Constructing a matrix of I_{lmn} (Equation 2.1), the coefficients of the characteristic equation (a, b, and c in equation 2.1) of the matrix is orientation invariant.

$$P(\lambda) = \lambda^3 + a\lambda^2 + b\lambda + c = \begin{vmatrix} I_{200} - \lambda & I_{110} & I_{101} \\ I_{110} & I_{020} - \lambda & I_{011} \\ I_{101} & I_{011} & I_{002} - \lambda \end{vmatrix} \tag{2.1}$$

2.2. Geometric Parameters

Geometric parameters are retrieved from original 3D shapes and their normalized models. These parameters include two aspect ratios, the ratio of overall surface area to volume, the scaling factor used to normalize the model, and overall volume.

2.3. Principal Moments

Moments represent the distribution of the models in a coordinate system. The combination of all order moments accurately describes the density function of CAD models and vice versa. In (Elad, M. et al. 2001), it is reported that 4th - 7th order moments have been used. However, higher order moments are sensitive to noise. In this paper, principal moments that are second order are used as feature vector. They are invariant to translation and orientation but dependent on scaling. In order to make them comparable, the principal moments of normalized models are employed in this paper.

The feature elements of principal moments and moment invariants have different advantages. The moment invariants are calculated from the moments of the model, of which the centroid is aligned with the origin. It does not require scaling normalization and rotation normalization. This avoids the errors resulting from the scaling process. These errors can cause a serious problem by introducing large noise to the CAD models (Cohen D. K. 1996). The elements of moment invariants, however, have undergone complicated computation. It is difficult to relate them with geometrical characteristics of the models.

The principal moments directly reflect the distribution of the density function of the model. Furthermore, these three features are of the same order. This makes the relevance feedback more meaningful, especially when linear combinations of similarity based on different feature vectors are used as the overall similarity.

2.4. Eigenvalues

We use the eigenvalues of the adjacency matrix of the skeletal graph of a 3D shape as of shape feature vector. Skeletal graphs are built based on the skeletons of 3D shapes. However, it is difficult to directly search graphs efficiently, since graph matching is an NP complete problem. We use the eigenvalues to represent skeletal graphs. The eigenvalues are indexed for searching efficiently.

Each element in the matrix represents the relationship between two nodes. The value of the element is determined based on the type of relationship, for example, loop-to-loop connection and loop-to-line connection have different values. More information with regard to this topic is available in (McWherter, A., Regli, W.C., 2001).

3. SIMILARITY MEASURE

A similarity measure is a function used to quantify the similarity between two shapes. It takes the feature vectors of the query model and those of a shape in the database and outputs a real number that reflects the degree of similarity between the two models. It is preferable that similarity measures satisfy the metric axioms (Santini, S. et al. 1999, McWherter, D. et al. 2001) as follows.

Let S be a set of objects; a metric on S is a function d: $S \times S \to \Re$, which satisfies three conditions, i.e., Equations. (1) - (3), for all $x, y, z \in S$.

$$d(x,x) = 0 \quad (3.1)$$

$$d(x,y) = 0 \Rightarrow x = y \quad (3.2)$$

$$d(x,y) + d(x,z) \geq d(y,z) \quad (3.3)$$

The first axiom implies that a model is most similar to itself. The second means that if using similarity measure we can not differentiate two models, they are identical. This is a critical constraint on the similarity measure. Based on these metric, two more properties can be derived as in Equations 3.4 and 3.5. They show the non-negativity and symmetry of similarity measure.

$$d(x,y) \geq 0 \quad (3.4)$$

$$d(x,y) = d(y,x) \quad (3.5)$$

In our search system, the similarity measure is defined based on a weighted Euclidean distance, which is shown by equation 3.6. Equation 3.8 computes the similarity between two 3D shapes. Through adjusting the weights of the weighted Euclidean distance, we can reconfigure the similarity measure to reduce the influence of the subjectivity of similarity of 3D models (Lou et al. 2003).

$$d = \sqrt{\sum_{i=0}^{\dim - 1} w_i \bullet (q_i - x_i)^2} \quad (3.6)$$

where: d is the weighted Euclidean distance;

q is feature vector of query shape;

x is the feature vector of a shape in DB; and

dim is the dimensionality of feature vector.

In the above equation, the weighted Euclidean distance (d) is a measure of dissimilarity between two shapes. Since we normalize each element of our feature vectors, a point (P) in the feature space is represented as *(P_1, P_2, P_3, ... P_n)* and $0 < P_i < 1$ for $i = 1, 2, \ldots n$. Theoretically, the maximum distance in this normalized feature space is between the two points, of which coordinates are *(0,0, ...,0)* and *(1, 1, ... 1)*. Therefore, in an n-dimensional space, the maximum weighted Euclidean distance is:

$$d_{\max} = \sqrt{\sum_{i=0}^{\dim - 1} w_i \bullet (1-0)^2} = \sqrt{\sum_{i=0}^{\dim - 1} w_i^2} \quad (3.7)$$

The similarity measure is defined in Equation 3.8. The weighted Euclidean distance is normalized by the maximum distance of points in the feature space, so that the maximum and minimum values of $\frac{d(q,x)}{\sqrt{d_{\max}}}$ are 1.0 and 0, respectively. The normalization ensures the degree of similarity between two shapes locates in the interval between 0 and 1.

$$s = 1 - \frac{d(q,x)}{\sqrt{d_{\max}}} \quad (3.8)$$

where: s is the similarity measure, and

q, x, d and d_{max} are as above.

4. EFFECTIVENESS

4.1. Procedures

Figure 2 shows a flow chart illustrates how a user interacts with the system and how our prototype system processes a query. It has an important post-processing mechanism – relevance feedback and multi-step refinement. A user submits a shape model, either created using modeling tools or picked from the interface presenting shapes sampled from the database, as a query example, specify the features to represent shape content; the system then goes through the feature extraction process and uses the feature vector as a key to search the multi-dimensional index; the results are shown in the interface; at this point, the user can choose whether to do Relevance feedback or multi-step refinement. For the experiments in this section, the relevance feedback feature is turned off.

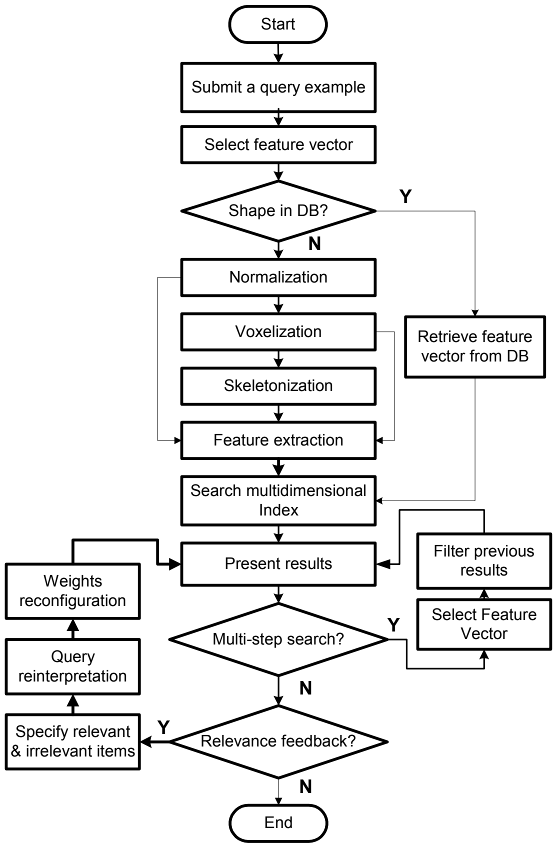

Figure 2 Flow chart of query processing

There are two ways that a user can control the number of the shapes to be presented. One method uses similarity threshold (SIT), which can be any value from 0 to 1.0. All the shapes, of which the similarity measures are larger than SIT, are considered as similar shapes and present to a user. The other method uses a natural number (NAN) that has to be smaller than the size of the database. The system presents NAN most similar shape to users, irrespective of the degree of similarity. In the following sections, we refer to these two methods SIT and NAN, respectively.

4.2. Methods

In this paper, the search effectiveness is characterized by two parameters – precision and recall, which are commonly used in information retrieval (Santini, S. 2001). The database employed for effectiveness is a real database of 113 models. These 113 models are manually classified into 26 groups as well as 27 "noisy shapes" that do not belong to any groups. The groups formed by manual classification are term in the paper "cluster map", which serves as the "ground truth" to test our system. When a model in group X is chosen as a query, the search results of the system are compared with the cluster map to compute the precision and recall, which are defined as follows.

Suppose that the true set of shapes in the database similar to the query is A and the retrieved shape set is R. Precision (Pr) is the ratio of the number of the relevant records retrieved (A∩R) to the total number of shapes retrieved (R). Recall (Re) is the ratio of the number of retrieved relevant shapes (A∩R) to the number of all relevant shapes (A) in the database. Figure 3 illustrates the relationship of A, R and A∩R. The mathematic definition of precision and recall are as Equation 4.1 and 4.2.

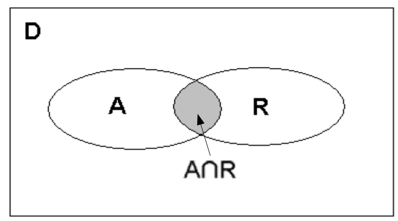

Figure 3 Illustration of precision and recall definition

$$\mathrm{Pr} = \frac{|A \cap R|}{|R|} \qquad (4.1)$$

$$\mathrm{Re} = \frac{|A \cap R|}{|A|} \qquad (4.2)$$

where $|A \cap R|$, $|A|$, and $|R|$ are the cardinalities of the corresponding set, respectively.

Obviously, the set R represents the models that the system considers similar ones; and the set A represents the models that the user considers similar.

An ideal search system retrieves the dataset exactly the same as all the similar shapes in the database, i.e. R=A, which make Pr = Re = 1.0. However, because of the complexity of presenting shape content in low level features, developing an ideal search system has been a challenge for years. It is impossible to ensure Pr = Re = 1.0 for all queries. In reality, the precision and recall have an inverse relationship. Usually one of the two parameters can grow at the expense of the other. Thus, precision and recall are usually used simultaneously to characterize the search performance. Neither of them can independently characterize the performance of the system. For example, if we set the threshold for similarity to a small value (close to 0), then the system could retrieve all the shapes from the database. In this case, the recall is 1, however the precision is low. If we enforce a strict measure on similarity (threshold close to 1), the system probably retrieves a very limited number of shapes. In this case the precision will be high. However the recall is low.

In this paper, we designed two types of experiments to evaluate the effectiveness of using different feature vectors. In one type of the experiments, we compared the precision and recall curve (PRC) plotted with average precision and recall values of different query models, which were chosen from the cluster map. In the other type of experiments, we compared the average precision (APR) and recall (ARE) of twenty-six quires.

4.3. Results and discussion

4.3.1 Precision and Recall Curve (PRC)

In order to characterize the search effectiveness of using different feature vectors, we chose five models from the twenty-six similarity groups and no two models are from same group. Figure 4 shows the models that were employed for this test. They are numbered sequentially from 1 to 5. In this section, we use SIT method to control the number of parts to be presented. Different pairs of precision and recall are collected by changing the SIT values, which are set as 0.7, 0.75, 0.8, 0.85, 0.9, 0.92, and 0.94. The shapes that have a larger similarity measure than the threshold are regarded as similar by the system and presented to the user. The precision and recall for this query are then evaluated using the known cluster map. If different precisions are received at same recall, we use the average precision to pair up with the recall.

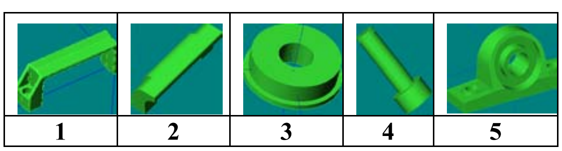

Figure 4 Sample shapes tested for Precision-recall curves

Figure 5 illustrates the result of a query. The top-left part is the query shape, which belongs to a group of five similar shapes. The feature vector used for this query is moment invariants. The similarity threshold is 0.85. The precision for this query is 0.22; and the recall is 0.50. When we compute the precision and recall, we do not count the query shape itself, because it is guaranteed to be retrieved.

Figure 5 An example of shape search (SIT= 0.85)

Figure 6 shows the precision-recall curve based on the results of the five queries. It can be seen from the figure that the PRC of moment invariants, geometric parameters, and principal moments show inverse relationship in general. However, the PRC of eigenvalues has almost fixed recall values for different precision values, which implies that no more similar shapes can be retrieved even though we increase the number of shapes to be retrieved.

Thus, the eigenvalues of the adjacency matrix of skeletal graphs are not suitable for generic shape descriptors for shape search. One of the reasons could be that the size of the skeletal graph is small, thus the eigenvalues can not differentiate different shapes. This will become worse when the database becomes larger. Therefore, it is necessary to use other local geometric information to improve selectiveness of the eigenvalues of the adjacency matrix of skeletal graph.

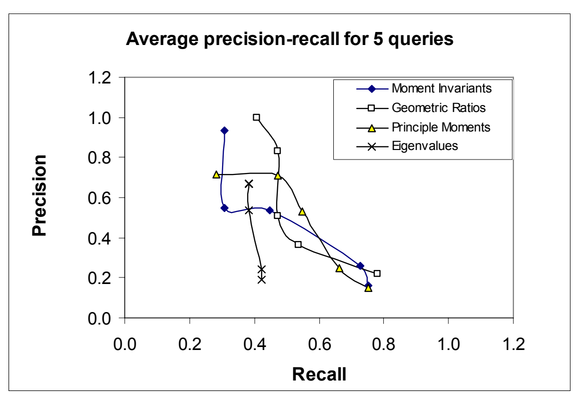

Figure 6 Precision-recall curve based on five queries

4.3.2 Average effectiveness (AE)

For the experiments in this section, we used NAN method to control the number of shapes that are presented to a user. We did two sets of experiments. In one set of experiments, we specified NAN values to the size of the group (SoG) in which the query shape is. Based on the definition of in Equations 4.1 and 4.2, the precision and recall are the same, since the size of retrieved data sets (R) is the same as that of the relevant data set (A). In the other set of experiments, the NAN value is set to 10, which is relatively larger than SoG. For both sets of experiments, the effectiveness is characterized by the average precision (APR) and average recall (ARE) of twenty six queries. We chose one and only one shapes as query example from each of the twenty six groups.

Figure 7 shows the average effectiveness (AE) of the first set of experiments. Since we retrieved number is SoG for Figure 7, the cardinalities of retrieved set R and that of relevant set A are equal, although the elements of these two sets are not necessary the same. Thus the precision and recall have the same value. Figure 8 shows average effectives of the second batch of experiments. It can be seen from both figures that descending order of AE of using different feature vectors are: principal moments, moment invariants, geometric parameters, and eigenvalues. With larger NAN value, we received higher recall and lower precision. In addition, the differences among the average effectiveness of principal moments, moment invariants, and geometric parameters become smaller, when the NAN value becomes larger.

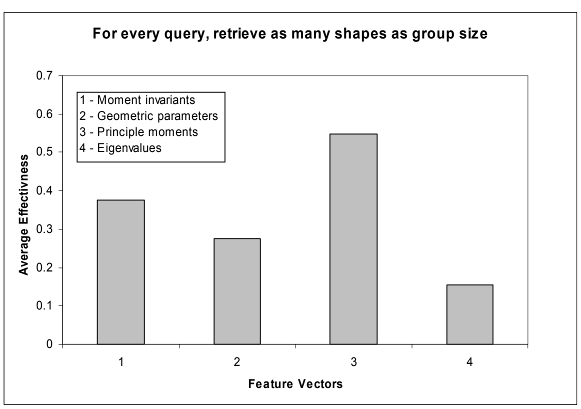

Figure 7 AE of four feature vectors (NAN = SoG)

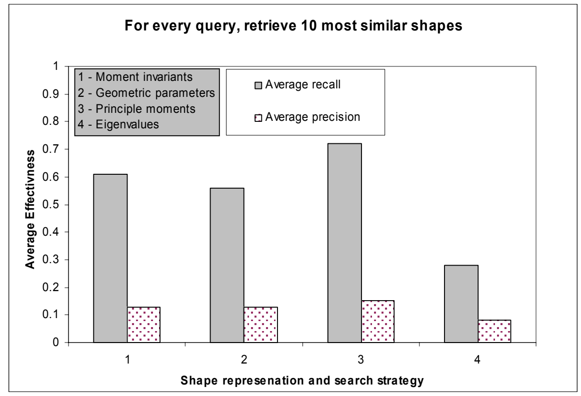

Figure 8 AE of four feature vectors (NAN = 10)

4.3.3 Improve Effectiveness

The tests to compare the precision and recall in the previous section were conducted with a one-shot searching approach. However, the semantic gap (Santini 2001, Lou, et al. 2003) between the low-level feature vectors and high-level shapes significantly affects the effectiveness of content-based shape search systems. In our previous publication (Lou et al, 2003), we have applied relevance feedback to bridge the semantic gap. Two approaches, query reconstruction and weight reconfiguration, have been investigated to improve the interaction between users and the system, so as to adjust the system's similarity view to a user's. The core part of the relevance feedback is the pattern recognition from the relevant and irrelevant dataset.

In this paper, we propose and implement a different approach - multi-step search - to improve the search effectiveness. Different from the relevance feedback, this approach gives a user more flexibility to employ her expertise to determine how to filter search results. After the search system presents the search results to the user, she observes the results and determines if she would like to further filter the previous results with other feature vectors. In our experiments, we observed many cases that the system retrieved more similar shapes using multi-step search than using individual or combined feature vectors. Figures 9 and 10 are examples using one-shot and multi-step, respectively.

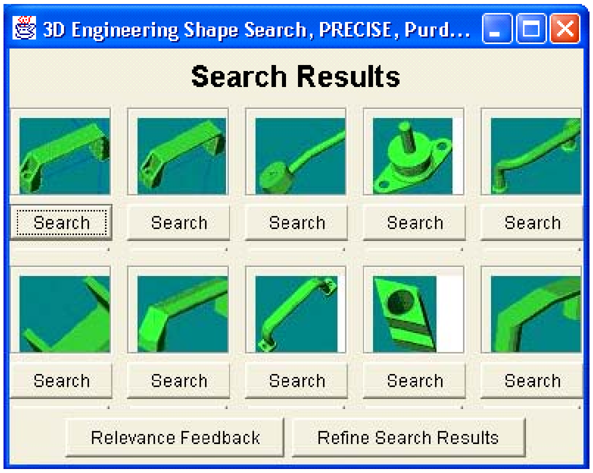

Figure 9 Search results using principal moments

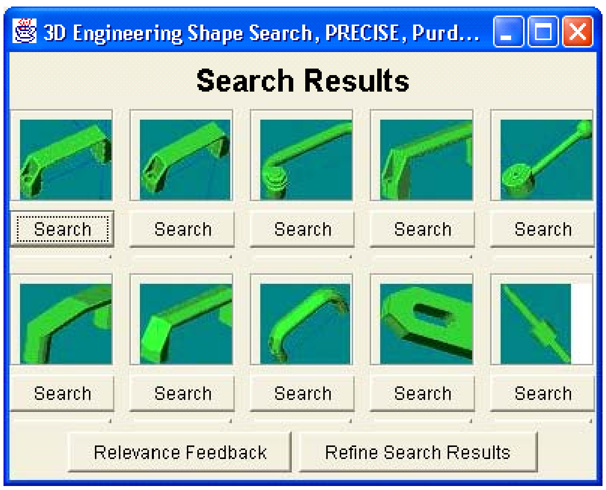

Figure 10 Search results using multi-step refinement

Figure 9 shows the best results using individual feature vector. Figure 10 shows the results using moment invariants first and geometric parameters second. In either of these cases, the system retrieves thirty shapes and presents the ten most similar shapes to users. Thus, for the multi-step search, the system first retrieves thirty shapes based on moment invariants, uses the geometric parameters to reorder these thirty shapes and then presents ten most similar shapes. The precision and recall of Figure 9 is 0.33 and 0.43, respectively, while for Figure 10, they are 0.55 and 0.71.

In our experiments, we observed that, in general, multi-step strategy outperformed the one-shot search. Nevertheless, there were also queries that multi-step search strategy did not outperform the one-shot strategy. In order to get an overall evaluation of multi-step search strategies, we conducted following experiments: from each of the twenty six groups, choose one shape as query model; submit the search and let the system present only a fixed number of shapes; compute the precision of the search results. The experiments were done with the number of the retrieved shape fixed to ten and the size of the group (SoG) to which the query shape belongs, respectively.

Figure 11 illustrates the average precisions of twenty six queries with different feature vectors when NAN is set to SoG. Figure 12 is the comparison when NAN is set to ten. It can be seen that the multi-step search strategy has better performance than any one-shot search with individual feature vector. The ARE and APR of multi-step search are both 51% higher than the best individual feature vector – principal moments, when NAN is set to SoG. When the size of retrieved shapes is increase, the ARE of multi-step search strategy still 15% higher than that of using principal moments. However, there is not much difference between the APRs.

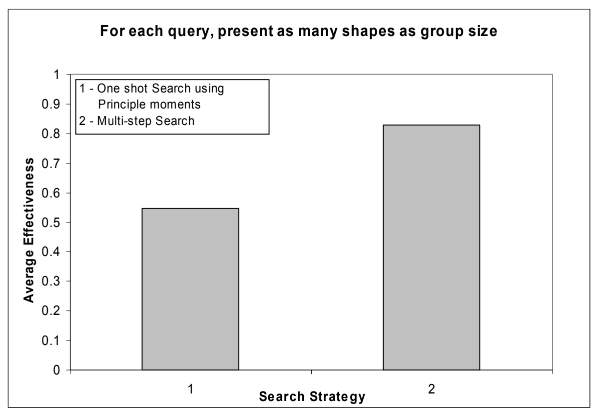

Figure 11 Comparison of the AE (NAN = SoG)

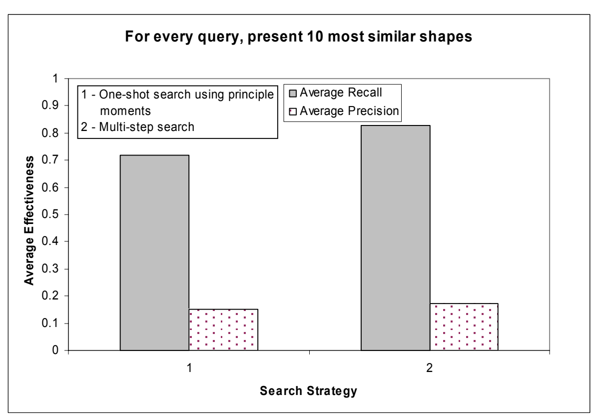

Figure 12 Comparison of AE (NAN = 10)

5. EFFECIENCY

5.1. Indexing Shape Similarity

Using index structures to speed up searching is a critical issue for 3D shape search systems. The logarithmic query time of B+ tree is in part responsible for the success of relational databases. However, the B+ tree index is basically a one-dimensional index structure, which orders the data records directly by attributes. This one-dimensional index structure is not sufficient for 3D shape search systems (Hellerstein, et al. 1997). The fundamental problem is that the feature vectors representing the 3D models are complex data types. Searching is usually based on overall similarity (similarity query) rather than matching individual attributes (attribute query). We have to use all the attributes simultaneously to determine the similarity. In other words, we cannot discard a model from a candidate list only because some attributes do not match the query model.

Thus, in order to search similar 3D models efficiently, we need an index structure with properties such that: (1) it is a multidimensional index. The multidimensional index arranges the models using all the feature elements, and (2) it groups similar models and puts them in the same node or contiguous index nodes. This grouping can reduce the disk I/O time during search operations.

R-tree based multidimensional indexes (Guttman, A. 1984) have been extensively studied for content-based image retrieval. In such an index, points in feature space are clustered in groups and a group is represented by a bounding rectangle/hyper-rectangle containing the points. The bounding hyper-rectangle is a tight bounding box that is represented by the coordinates of its vertices. To answer a query, the query point is compared with the bounding box in order to prune the sub-tree rooted at this hyper-rectangle. Therefore, the R-tree index satisfies the requirements of the similarity index listed above. In R-tree index, the leaf nodes of an R-tree contain pointers to 3D models. The search starts at the root and is directed by internal node to the leaf node. The tree structure is similar to a B+ tree; however, the criterion to arrange the records is totally different from that of B+ trees. In this section, we investigate the efficiency of the R-tree index using our content-based 3D search system.

5.2. Methods

In order to test the efficiency of R-tree based index in searching similar 3D shapes, we used both real and synthetic databases. The real database accurately represents the distribution of feature vectors in feature space. However, at this moment, a real database, which is large enough to test the scalability of indexing techniques, is not available. Although our real database is being expanded, the current database, consisting of 113 3D shapes, is still relatively small as compared to typical design repositories. Thus, we use synthetic databases to investigate the scalability of indexing techniques. The synthetic database was built using a random number generator. The sizes of synthetic datasets are flexible. We chose the range from 50 to 1,000,000 in the experiments.

The search efficiency is characterized by the ratio (Ra) of the number of the visited node (NV) in a search operation to the total number of nodes (NN) in the index.

$$Ra = \frac{NV}{NN} \tag{5.1}$$

The visited nodes are loaded into computer memory (RAM) from the database during search operations. The query model is compared with the models in the retrieved node to determine the search direction or to compute similarity. Since disk I/O is the bottleneck for current computer systems, the ratio of visited nodes reflects overall performance of search systems.

The efficiency of the R-tree index is determined by its pruning ability. The search basically proceeds through all the nodes at a level and prunes the nodes based on pruning criterion. In (Roussopoulos, et al. 1995, Jagadish, 1991), there is a detailed description about pruning the nodes using MinDist and MinMax of the node to a query model. The pruning ability is, in turn, determined by the distribution of similarity models. If similar models are grouped in the same node or similar nodes, the R-tree has better pruning ability.

In this section, the dependences of efficiency on three parameters - dimensionality of data records (DDR), volume of database nodes (VDN), and size of databases (SoD) - are investigated with both real and synthetic databases. The tests were conducted on a DELL Pentium 2.66 GHz PC with 1.0 GB RAM. All the tests in this section use NAN methods described in Section 4.1. The NAN value is set to one. This means the system retrieves only one most similar shape from the database.

5.3. Results and discussion

5.1.1 Real Database

As described in the previous section, there are four types of feature vectors that are extracted from 3D shapes. The sizes of these feature vectors correspond to the dimensionality of data records (DDR). DDRs of the four feature vectors vary from 3 to 10.

Besides using these single feature vectors, we randomly combined a few groups (from 2 to 4) of feature vectors to form a new feature vector. This is done by a simple database join operation. The DDR of all single and combined feature vectors ranges from 3 to 18. An R-tree index was created for each of the feature vectors to test the efficiency of the index. Figure 13 shows efficiency of R-tree indexing real databases. It can be seen that the value of Ra can be as low as approximately 0.1, which means that there is only one that is retrieved for checking similarity for every ten database nodes. In addition, it can be seen that the Ra received with DDR less than 10 is substantially smaller than that when DDR is larger than 10. This shows that the efficiency decreases when DDR increases, and the efficiency significantly deteriorates when DDR reaches 10. In (Beyer, K. et al. 1999), it is observed that a linear scan might outperform a multi-dimensional index when the DDR reaches 7 or 8, due to the "curse of dimensionality".

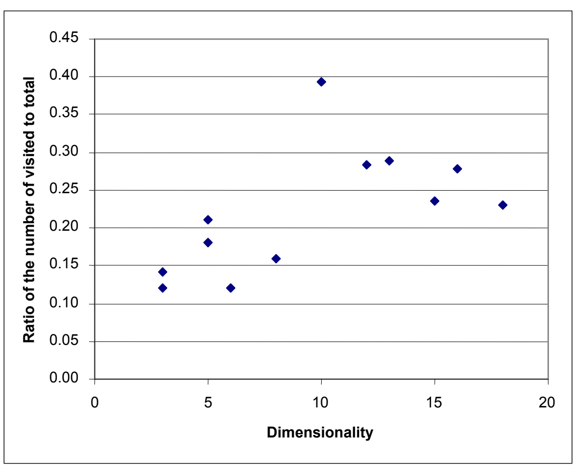

Figure 13 Efficiency of R-Tree index (real database)

5.1.2 Synthetic Database

In this section, synthetic database is used to test the dependence of efficiency on the volume of database node (VDN), the dimensionality of data records (DDR), and the size of database (SoD). VDN, defining the maximum data records that can be stored in a database node, is the capacity of a node.

Figures 14 shows the ratios (Ra) changing with different SoD and DDR, respectively. The VDN of these experiments is set to 5. Figure 15 illustrates the Ra at different VDN and SoD when DDR is set to 5.

It is seen from these figures that the multidimensional index prunes a large number of nodes, especially at lower DDR and VDN. In the case where the DDR is 3 and the VDN is 5, the ratio of visited is about 0.00125. In other words, for every 800 nodes, there is only one loaded in temporary memory for similarity computation. It can be seen from the figures that the Ra increases when the SoD, DDR, and VDN increase. Furthermore, it is clear that the efficiency of the R-tree becomes stable when with the increase of the size of databases (SoD). This means that the R-tree is scalable to database size.

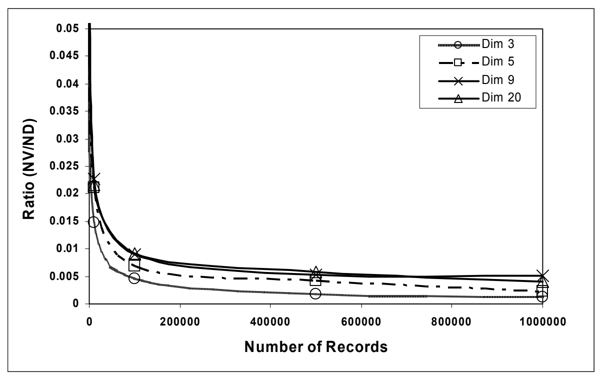

Figure 14 Ratios at different dimensionality and database sizes

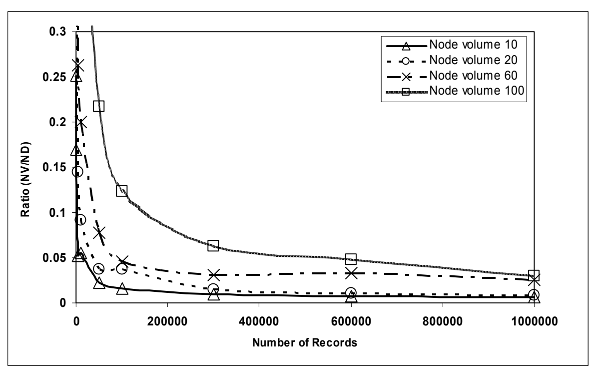

Figure 15 Ratio with different database size and node volumes

6. CONCLUSION AND FUTURE WORK

In this paper, we describe our approaches to improve search effectiveness and efficiency of content-based 3D shape search systems. A systematic method is developed to compare the effectiveness and efficiency in shape retrieval using real and synthetic databases.

The search effectiveness is characterized by the precision and recall of queries. The descending order of search effectiveness of using feature vectors are: principal moment, moment invariant, geometric parameters, and eigenvalues.

An R-tree based index is employed to improve the search efficiency. The results show that the index structure significantly improves search efficiency with both real and synthetic databases. The search efficiency decreases when the dimensionality of data record and the capacity of database node increase. It becomes stable with the increase of database size.

Multi-step refinement is proposed as a strategy to improve search effectiveness. The performance of the multi-step refinement is compared with one-shot search scheme. The precision and recall of using multi-step refinement is 51% higher than that of one-shot search using principal moment as feature vector, although the difference becomes smaller when the number of retrieved shapes become larger.

ACKNOWLEDGEMENTS

Initial funding for this project came from the 21st Century Research and Technology Fund award from the state of Indiana. We acknowledge the support from the Innovation Realization Lab at Purdue University and the University Faculty Scholar Award from Purdue University. We thank Prof. Christoph M. Hoffmann in Computer Science and Prof. Zygmunt Pizlo in Psychological Science, Purdue University, for their valuable suggestions and helps.

REFERENCES

Connor, D., Gutknecht, M., (1998), “Helping Clients Harness Knowledge to Drive Innovation,” Proc. of the Second International Conference, Basel, Switzerland, October 29-30.

Lu, W. F., Liu, X. F., Chang, H., Dong, L., (1997), “Feature-based Representation and Indexing of Axisymmetric Parts in Machining Process Planning using Case-Based Reasoning,” 9th International Conference on Tools with Artificial Intelligence (ICTAI '97), Nov.3-8, 1997, Newport Beach, CA

Ramesh, M. M., Yip-Hoi, D., Dutta, D., (2001), “Feature Based Shape Similarity Measurement for Retrieval of Mechanical Parts,” JCISE, Vol. 1, No. 3, pp. 205-280,

Cicirello, V., Regli, W. C., (2001), "Machining Feature-Based Comparisons of Mechanical Parts," Int'l Conf. on Shape Modeling and Applications, sponsored by ACM SIGGRAPH, the Computer Graphics Society and EUROGRAPHICS, IEEE Computer Society Press, Genova, Italy, 07-11 May 2001, pp.176-185.

Elinson, A., Nau, D. S., and Regli, W. C, (1995) “Featurebased similarity assessment of solid models”, In Christoph Hoffman and Wim Bronsvoort, editors, Fourth Symposium on Solid Modeling and Applications, pages 297--310, New York, NY, USA, May 14-16 1995. ACM, ACM Press. Atlanta, GA.

Lou, K., Jayanti, S., Iyer, N., Kalyanaraman, Y., Prabhakar, S., Ramani, K., (2003) "A Reconfigurable, Intelligent 3D Engineering Shape Search System Part II: Database Indexing, Retrieval and Clustering", ASME DETC' 03, Sept2-6, 2003, Chicago.

Hartly, J. R., (1992) Concurrent Engineering: Shortening Lead Times, Raising Quality, and Lowering Costs, Productivity Press, 1992

Iyer, N., Kalyanaraman, Y., Lou, K., Jayanti, S., Ramani, K., (2003) "A Reconfigurable, Intelligent 3D Engineering Shape Search System Part I: Shape Representation", ASME DETC' 03, September 2-6, 2003, Chicago, Illinois.

Kelley L. A., MacCallum R. M., and Sternberg M. J. E., (2000), “Enhanced Genome Annotation using Structural Profiles in the Program 3D-PSSM”, J. Mol. Biol. 299(2), 501-522

Kastenmüller, G., Kriegel, H. -P., and Seidl, T., (1998), “Similarity Search in 3D Protein Databases,” Proc. German Conf. on Bioinformatics, Cologne, Germany.

Thiran, J. -P., and Butz, T., (2000), "Fast non-rigid resistration and model-based segmentation of 3D images using mutual information," SPIE; San Diego; USA, 3979, (2), 2000, pp. 1504-1515.

Hu, M. K., (1962), “Visual Pattern Recognition by Moment Invariants,” IRE Trans. on Information Theory, Vol. 8, pp. 179-182.

Sadjadi, F. A., and Hall, E. L., (1980), “Three-dimensional Moment Invariants,” IEEE Trans. Pattern Analysis Mach. Intell. PAMI-2, pp.127-137.

Ankerst, M., Kasternmuller, G., Kriegel, H. P., and Seidl, T., (1999), “3D Shape Histograms for Similarity Search and Classification in Spatial Databases,” Proc. Int’l Symposium on Spatial Databases (SSD’99), Hong Kong, China.

Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D., (2002), "Matching 3D Models with Shape Distributions," Shape Modeling International, Genova, Italy, May 2001.

Cyr, C. M., and Kimia, B. B., (2001) “3D Object Recognition Using Shape Similarity-Based Aspect

Graph," Proc. of IEEE International Conference on Computer Vision, Vol. 1, pp. 254-261.

Zloof, M., (1975), "Query by Example", Proc. of AFIPS Conference, 44, 431—432.

Hellerstein, J. M., Koutsoupias, E., and Papadimitriou, C. H., "On the Analysis of Indexing Schemes," SIGMOD-SIGART Symposim on Principles of Database Systems, pp 249-256, 1997.

Roussopoulos, N., S. Kelley, and F. Vincent, "Nearest neighbor Queries," SIGMOD, pp71-79, 1995.

Cybenko, G., Bhasin, A., Kurt, D., Cohen, K. D., (1997), "Pattern Recognition of 3D CAD Objects: Towards an Electronic Yellow Pages of Mechanical Parts," Vol.1, Smart Engineering Systems Design, pp.1-13.

Palagyi, K., and Kuba, A., (1998), "A 3D 6-subiteration Thinning Algorithm for Extracting Medial Lines," Pattern Recognition Letters, Vol. 19, pp. 613-627.

Cohen, D.K, (1996), "Feature Extraction and Pattern Analysis of Three-Dimensional Objects", Master Thesis, School of Engineering, Dartmouth College.

Elad, M., Tal, A., and Ar, S., (2001), "Content based Retrieval of VRML Objects - An iterative and Interactive Approach," The 6th Eurographics workshop in Multimedia, Manchester UK, Sept. 2001.

McWherter, D., Peabody, M., Shokoufandeh, A., and Regli, W. C., (2001), "Database Techniques for Archival of Solid Models," 6th ACM/SIGGRAPH Symp. on Solid Modeling and Applications, Ann Arbor, MI, 06 Jun 2001, pp. 78-87.

Guttman, A., (1984), "R-trees: a dynamic index structure for spatial searching, Proc. of the SIGMOD Conference, Boston, MA, June 1984, pp. 47-57.

Santini, S., (2001), Exploratory Image Databases: Content-Based Retrieval, Academic Presses.

Santini, S., and Jain, R., (1998), "Beyond query by example," Proc. of the Sixth ACM International Multimedia Conference, Bristol, England, pp. 345-350.

Vranic, D. V., and Saupe, D., (2001), "3D Shape Descriptor Based on 3D Fourier Transform," Proc. The EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services, Budapest, Hungary, 2001, pp. 271-274.

Kazhdan, M., and Funkhouser, T., (2002), "Harmonic 3D Shape Matching," to appear as a SIGGRAPH 2002 Technical Sketch.

El-Mehalawi, M., and Miller, R.A., (2003), "A Database System of Mechanical Components Based on Geometric and Topological Similarity. Part I: Representation," Computer-Aided Design, Vol. 35, Issue 1, Jan, 2003, pp. 83-94.

Jagadish, H. V., (1991), "A Retrieval Technique for Similar Shapes," SIGMOD, 1991, pp. 208-217.

Santini, S., and Jain, R., (1999), "Similarity Measures," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.21, No.9, pp.871-880.

Beyer, K., Glodstein, J., Ramakrishnan, R., and Shaft U., (1999), "When is 'Nearest Neighbor' meaningful?" Proc., 7th International Conference, Jerusalem, Israel, pp. 217-235.

Cardone, A., Gupta, S. K., and Karnik M., (2003), "A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications", Journal of Computing and Information Science in Engineering, Vol. 3, pp. 109 -118.

McWherter, A., Regli, W.C., (2001), "An Approach to Indexing Databases of Solid Models", Technical Report DU-MCS-01-02, Department of Mathematics and Computer Science, Drexel University.

Iyer, N, Lou, K, Jayanti, S, Kalyanaraman,Y, Ramani, K, (2003) "Shape Based Searching For Product Lifecycle Management Systems", submitted to Journal of Computer-Aided Design.