

A 2D Sketch-Based User Interface for 3D CAD Model Retrieval

Jiantao Pu, Kuiyang Lou and Karthik Ramani

Purdue University, {pjiantao, kuilou, ramani}@purdue.edu

ABSTRACT

This paper describes a sketch user interface enhanced by feedback for 3D CAD model retrieval. Users can express their intent by sketching 2D shape in the way as engineers draw three views of 3D models. It not only supports users' free form sketches, but also accepts users' further editing operations and feedbacks. The interaction paradigm proposed in this paper is supported by a 3D shape matching method, in which three problems are solved: determination of projecting planes and directions, 2D view generation, and similarity measuring between views. In addition, experiments are conducted to evaluate the performance of this sketch user interface by 3D model retrieval.

Keywords: information retrieval, CAD models, sketch user interface, similarity measurement

1. INTRODUCTION

Until now, many methods have been proposed to retrieve desired 3D models quickly from a database. These methods can be classified into four categories: (1) feature-vector based method [1-5], (2) statistics-based method [6-8], (3) topology-based method [9-12], and (4) image-based method [13-14]. However, few of them can be directly applied to the user interface of 3D model retrieval because the shape features can not be expressed to computer intuitively. The existing interaction ways for 3D model retrieval can be summarized as four paradigms:

- **Keyword query.** As the most popular way to retrieve information, keywords are also used in the 3D model retrieval. They discriminate 3D shapes by some semantics information, such as name, color, texture, function, file size or ID number. However, this paradigm depends on many factors, such as culture, habit, language, etc. Such example systems include CADlib [15], Mesh-Nose [16] and the National Design Repository [17]. Despite its simplicity, it is not sufficient and easy to describe 3D shape correctly using text.
- **Hierarchy browser.** To reduce the search space, most 3D model repositories provide a browsing interface, in which users are required to traverse a hierarchy by selecting keywords. The models in the library are classified according to the keywords. This paradigm can be implemented easily although intensive manual efforts are needed to organize the models. Generally, this data organization is done under the guidance of the semantics of 3D shapes. Users need to own some knowledge about the semantics of shapes.
- **Example query.** An interaction way adopted by most shape matching methods is to query by an existing example. Users are generally allowed to upload a 3D model or select a result from a previous research as an input. If a query model is not available, users have to create it from the scratch. If the 3D model is complex, this process will be time consuming.
- **Sketch query.** To overcome the above limitations, Thomas et al. [14] introduce a web-based search engine that supports queries based on 3D sketches, 2D sketches, 3D models, and/or text keywords. The 3D-sketch user interface is supported by a 3D sketch tool named Teddy, which is not easy for novices and some special training is needed. By the 2D sketch user interface, users can draw 2D shapes with a pixel paint program. The sketches are matched to several 2D projections of each 3D model in the database using an image matching method. The obvious advantage of this method is easy to use. However, 13 thumbnail images with the boundary contours of each 3D object from 13 orthographic view directions are needed to be prepared during a preprocess phase.

In this paper, we present a new 2D sketch user interface. The query process is similar to the process that engineers express their shape concepts on a piece of paper. No training is required for users. In contrast with the method presented by Thomas et al. [14], our proposed method only need three or six views which are determined automatically and sufficient to represent the 3D shape accurately. Users can not only emphasize some shape by

specifying weights for views, but also do some editing operation on the views obtained from the retrieved models. The whole retrieval process forms a loop by which users can refine the search results step by step.

The content of this paper is organized as follows. In Section 2, the framework of our proposed sketch-based user interface is outlined and some key components are described briefly. Section 3, 4 and 5 explain the involved algorithms that support this user interface paradigm in detail, which is followed by the introduction of the prototype system implementation in Section 4. Section 5 presents some experimental results and discussion. Finally, the conclusion and future work are described.

2. Framework

Fig.1. shows the framework of our proposed sketch user interface for 3D model retrieval. The sketch editor is used to hand draw the sketches and execute some editing operations (e.g., erase, modify and copy). It can also display the views generated from 3D models so that users can edit the retrieval results and begin a new round retrieval for refinement. The “View Generation” module aims to generate 2D views automatically from 3D shapes. The views consisted of a series of line segments to represent the silhouette of models. The “Similarity Measurement” module is to compute the similarity between different views. Although the process of ortho-view generation for 3D models in a database can be executed at run-time, this step is usually finished offline to accelerate the retrieval efficiency. It is worth to point out that the feedback process is related to all these modules. In the next Sections, these components will be explained in detail.

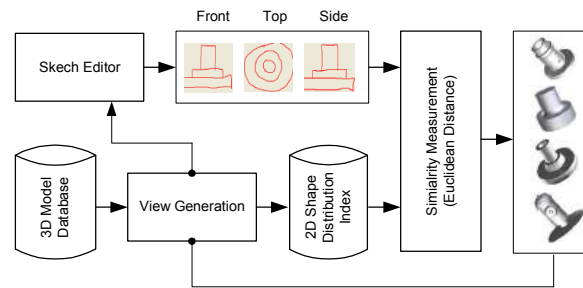


Fig. 1. Framework of the 2D sketch user interface.

3. SKETCH EDITOR DESIGN

The responsibility of the sketch editor is to record users’ search intent and translate it correctly to a computer. Users can employ pen or mouse to draw sketches. In our design, we use a series of line segments to represent the sketches. Fig. 2. shows the visual appearance and the architecture of the sketch editor. In the query process, it is unavoidable for users to make some mistakes. Therefore, some editing operations are provided for users. On the top of Figure 2(a), some editing functions are provided to modify users’ sketches.

The sketch editor consists of four basic function modules. The first one monitors the action of the mouse or pen. Once the mouse cursor is moving and the left button is pressed down, it can be concluded that users begin to draw sketches and the moving path of the cursor will be recorded at real time. Because the path is represented by a series of points, the sketches can be recorded as a series of line segments formed by two adjacent points.

The second module enables users to edit the generated sketches. Some basic operations, such as erase, trim, move, rotate, zoom, and view copy, are included. Users can not only erase and trim unnecessary sketches but also execute affine transformations. In practice, some 3D models may have same views along different directions. Therefore, in order to improve the hand-draw efficiency, “view copy” is provided for users to copy one view to another viewing region. More operations can be added into this system, although only a few basic operations are provided in this prototypical system.

The third module is the weight configurator, through which users can specify weight values to emphasize certain views. The larger the weight, the more important the view is to a user. The influence of the weight values can be reflected by the similarity measurement as Eqn.(1) which will be explained in later Section:

$$Similarity = \sum_{i=0}^n w_i (s_i - k_i)^2 \quad \sum_{i=0}^n w_i = 1 \quad (1)$$

where w_i is the weight of view i .

The fourth module displays the views generated from the retrieved models. This function makes it possible to allow users to reuse the retrieved result and edit it for further retrieval. In so doing, a user does not need to start from the scratch to create query views.

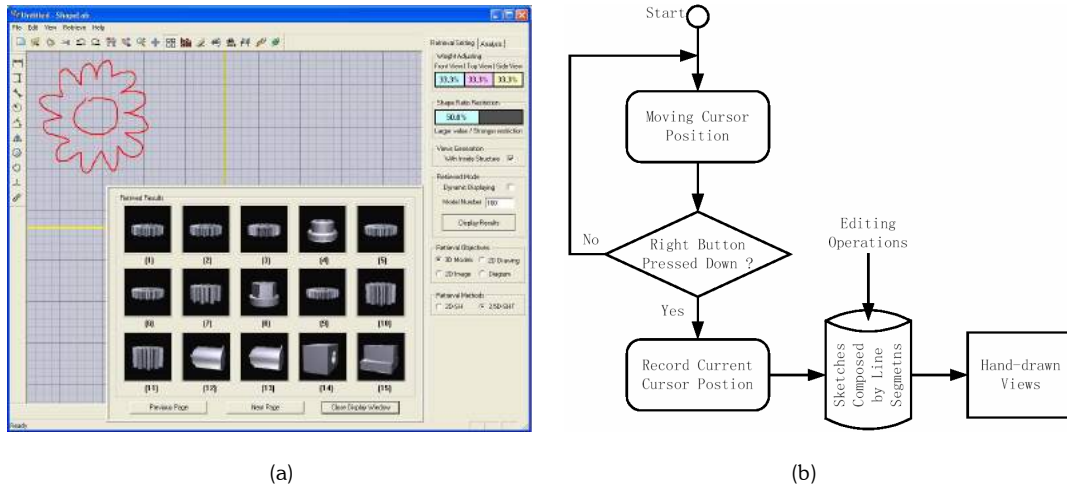


Fig. 2. The sketch editor: (a) the visual appearance; and (b) the logic architecture.

4. 2D VIEW GENERATION

The purpose of 2D view generation is to obtain some typical views that match to the 2D sketch. This process should be in accordance with human visual perception so that most users can express the concept of 3D shape intuitively. With this method, three or six stable views can be obtained by projecting 3D model onto the planes of a bounding box. 2D view generation consists of two function modules: bounding box determination and silhouette calculation. The first one determines the projection plane and direction, while the second one generates the views similar to the practical engineering drawing.

4.1 Bounding Box Determination

To obtain a robust and stable bounding box, we have proposed the maximum normal distribution method (MND). It can find out a bounding box that has better accordance with human perception mechanism than the commonly used inertial principle axis (IPA) method. As for its detailed explanation, refer to [18] in which there is special explanation. We will describe basic steps that MND method includes as follows.

First, the normal direction \mathbf{N}^k of each triangle $\square^k \mathbf{p}^k \mathbf{q}^k \mathbf{r}^k$ is computed and normalized. It is the cross product of any two edges:

$$\mathbf{N}^k = \frac{\mathbf{p}^k \mathbf{q}^k \times \mathbf{q}^k \mathbf{r}^k}{\|\mathbf{p}^k \mathbf{q}^k \times \mathbf{q}^k \mathbf{r}^k\|} \quad (2)$$

Second, the area of each triangle a^k is calculated and the sum of the areas of all triangles with the same or opposite normals are computed. We treat the normals with positive and negative directions belong to the same distribution. From the normal distributions, the normal with maximum areas is selected as the first principal axis \mathbf{b}^u . By traversing the remaining distribution, we can determine the next principal axis \mathbf{b}^v that satisfies two conditions: (1) with maximum areas, and (2) orthogonal to the first normal. The third axis can be obtained by the cross product between \mathbf{b}^u and \mathbf{b}^v :

$$\mathbf{b}^w = \mathbf{b}^u \times \mathbf{b}^v.$$

Third, the center and the half-length of the bounding box are calculated by projecting the points of the convex hull onto the direction vector and finding the minum and maximum along each direction.

With the information generated by the above processes, we can determine the bounding box of a 3D model. Fig.3. shows some examples obtained by this method. If you move the line connected the cash machin (as Fig.3(a)), the

direction of the bounding box will not change accordingly although the size might be changed. Therefore, this method can derive stable views since the projection direction and plane are fixed.

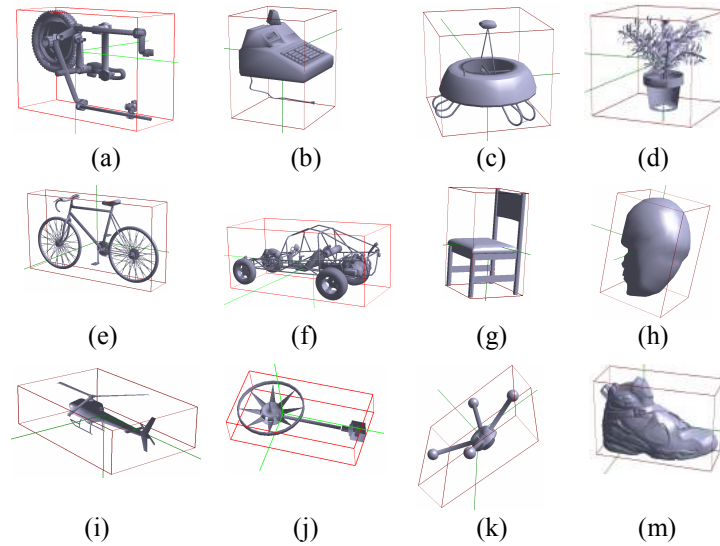


Fig. 3. Some pose determination examples.

4.2 Silhouette Calculation

For a polygon mesh (as Fig. 4(b) shows), the direct parallel-projection upon a plane will result in an unordered triangle grid. To diminish this effect and obtain a clear silhouette to describe the view shape, we need to cull the occluded polygons and the connection between edges efficiently. Three steps are included for this purpose: (1) backface culling; (2) inside-edge culling; and (3) projection along viewing direction.



Fig. 4. The necessity of silhouette calculation.

The first step is to cull backface. To determine if a polygon is facing the camera, we need only calculate the dot product of the normal vector of that polygon, with a vector from the camera to one of the polygon's vertices. If the dot product is less than zero, the polygon is facing the camera. If the value is greater than zero, it is facing away from the camera. However, for the case when the dot product is equal to zero, we regard it as visible because the side polygon will contribute to the final view. An example after backface culling is shown in Fig.5(b).

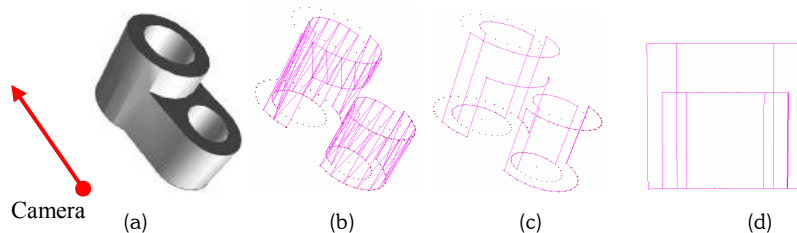


Fig. 5. Three steps for silhouette calculation: (a) backface culling; (b) inside-edge culling; and (c) projection along viewing direction

As Fig. 5(b) shows, the culled result is still represented by polygons. To obtain the contour lines of 3D models, the inside edges have to be discarded. For a triangle in a polygon mesh, if an edge is shared by two faces, then the edge is termed “inside edge”. With these definitions, we can cull the inside edges completely by traversing all the triangles. Fig.5(c) shows the results after inside-edge culling. After this step, we can project the result upon a projection plane. Fig.5(d) is the projection result. It is also the common representation in engineering practice. However, when the inside structure of the 3D model is complex, this kind of view will affect the performance of 2D view similarity measurement. Such an example is shown in Fig.6(b).

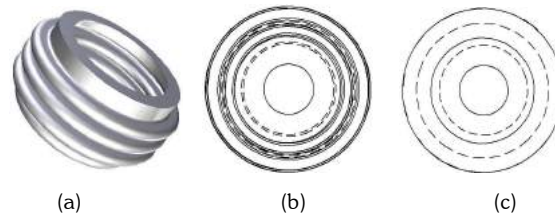


Fig. 6. The projected view with complex inside structure.

Therefore, we use a further step to cull the occluded polygons. In this paper, we use a depth-detection method to minimize the intersection operations. First, the 3D models are transformed to make sure that the camera direction is in accordance with positive z-axis. Second, given a polygon, we check its depth with other polygons. If its depth is greater than the depth of some other polygons, then it is possible that the given polygon will be occluded. Finally, by calculating the ray-polygon intersection between these potential occluded polygons, we can discard the occluded polygons, as Fig.6(c) shows. This step eliminates the influence of the inside structure. In the experiments, we will compare the influence of the inside structure on the performance of 3D model retrieval. Fig.7 is an example for the view generation.

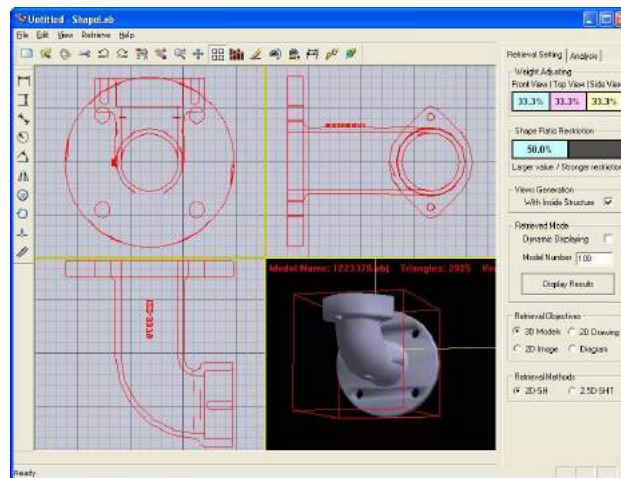


Fig. 7. The projected view with complex inside structure.

5. SIMILARITY MEASUREMENT

Since the 3D models are represented by 2D views, the similarity problem of two 3D models is converted to compute the similarity between 2D views (as Fig.8 shows). We have proposed a 2D shape distribution method. For its detailed explanation, refer to [18]. We describe the basic steps as follows:

- (1) The first step samples a large amount of random points on the edges. In our prototype system, we sample 100,000 points along edges uniformly.
- (2) The second step calculates the Euclidean distance distribution between two random points. The histogram can be represented as Fig.9 shows. The two histograms are the 2D shape distribution of the two views in Fig.8.

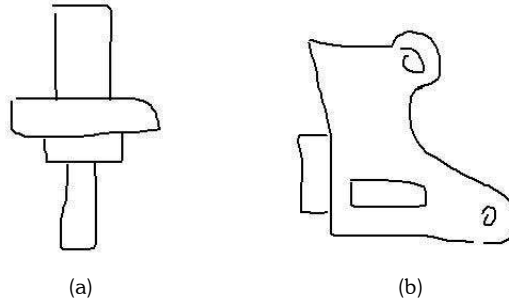


Fig. 8. Similarity measurement between (a) and (b).

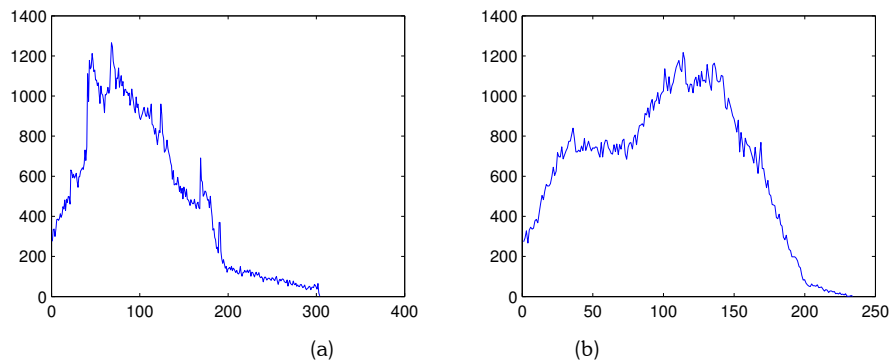


Fig. 9. 2D shape distributions of the views shown in Figure 8.

(3) The final step is to compute the difference between the shape distributions. We divide the distribution along distance-axis into 500 blocks and compute the average values. For any two histograms, we can accumulate the differences between these blocks. This process can be expressed by a formula as Eqn.(1), where s and k are the histogram blocks of two distributions.

6. FEEDBACK INTERACTION FOR REFINING RETRIEVAL

ShapeLab provides several feedback ways for users to interactively refine retrieval results:

- **Weight Value Adjustment for Orthogonal Views:** In this paper, a 3D model is described by three orthogonal views. Different views reflect certain shape characteristics from different perspectives. To find similar shapes with certain features, users are allowed to emphasize certain views by adjusting their weights. This interaction is illustrated by Fig.10. In this process, after some coarse results are retrieved, users can refine the results by assigning a higher weight to the views that play a more important role in the desired shape. The larger the weight value is, the more important the view will be. Users can repeat this process to refine the searched results.

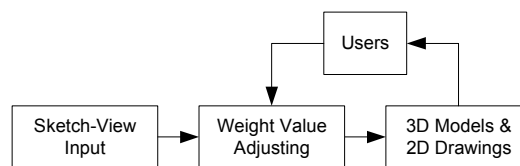


Fig. 10. Retrieval refinement by adjusting weights of different views.

- **Initial Sketch Input Editing:** In some cases, just adjusting weight value is not enough because the initial sketch is not sufficient to represent the desired models and sometimes users make mistakes. Enabling a user to edit the initial sketch input is an efficient way to refine the search results, especially when the searched results can

be displayed dynamically along the modification of the sketch. This interaction can be illustrated in Fig.11. Users can modify the input sketches according to the retrieved results.

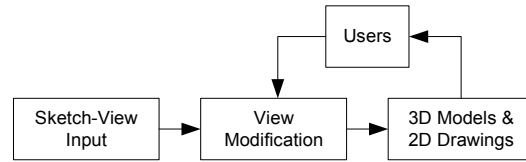


Fig. 11. Feedback process by initial sketch input editing.

- Retrieved Result Editing: In the two interactions described previously, retrieved results are used to help users to check and modify the improper input. We also propose a third approach to edit the views of the retrieved 3D models or 2D drawings to improve and refine the retrieval results. As described in Section 4, the 2D views of a 3D model can be generated automatically. Therefore, users can input a simple coarse sketch first and find some similar 3D models as another round sketch input. Based on the generated views, users can edit it according to their shape concepts. Because the generated views are more regular than the hand-drawn ones, this kind of feedback interaction is expected to achieve better performance. The feedback process can be illustrated by Fig.12. In contrast with the previous feedback mechanism, this feedback mechanism allows users to edit the views of the retrieved models.

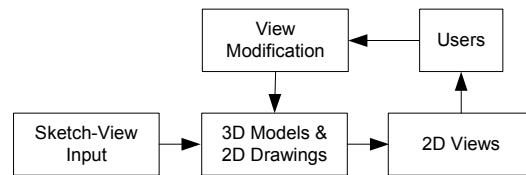


Fig. 12. Feedback process by retrieved result editing.

- Dynamic Retrieved: With the above feedback mechanisms, users can retrieve 2D drawings or 3D models in a coarse-to-fine way. Generally, it is not easy for users to find proper weight values or modify the sketches just within one time. Users usually have to repeat this process for several times. To accelerate this process, we provide a dynamic retrieval mechanism, as Fig.13 shows. A user's behaviors are monitored by ShapeLab all the time. During the feedback-based interactions, once an operation is finished, the retrieval results will be dynamically refreshed so that a user can experience the impact of his operation upon the retrieval results. In this way, users can get instant feedback and adjust their operations efficiently.

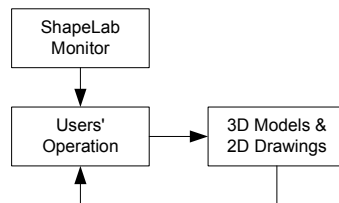


Fig. 13. Dynamic retrieval.

7. EXPERIMENTS AND DISCUSSION

The proposed sketch user interface is implemented as a prototype system on Windows platform. In order to test the performance, we have built a benchmark, namely Purdue 3D Shape Benchmark, which includes 801 CAD models from industrial fields. All the data is available publicly from our website <http://engineering.purdue.edu/precise/PBenchMark.html>. The 3D model benchmark consists of 42 clusters of similar parts. Following, we will present some experimental evaluation results.

The Precision-Recall (PR) curve is the most common way to evaluate the performance of a retrieval system. Recall measures the ability of a system to retrieve the desired objects, while precision measures the ability of a system to weed out what users do not want. Definitions of precision and recall are illustrated in Figure 25. PR curves describe the relationship between precision and recall for an information retrieval method. We used standard techniques of

constructing PR curves from the NIST TREC standards [19]. A perfect retrieval retrieves all relevant models consistently at each recall level, producing a horizontal line at precision = 1.0. However in practice, precision decreases with increasing recall. The closer a PR curve tends to the horizontal line at precision = 1.0, the better the information retrieval method. If the PR curves for two information retrieval methods cross each other or are very close to each other, it is difficult to make judgments about the relative effectiveness of each method.

For 3D model retrieval, to compare the performance between our proposed method (namely Orthogonal Main View) and other methods, we implemented six other available 3D shape descriptors: (1) Moment Invariants (MI) [20]; (2) Spherical Harmonics (SH) [21]; (3) Crinkliness and Compactness (CC) [22]; (4) 3D Shape Distribution (3DS) [8]; (5) Convex Hull Histogram (CHH) [23]; and (6) Solid Angle Histogram (SAH) [24]. The final PR curves are shown in Fig.14. From the precise-recall curves, it has been found that our proposed method has the best discrimination ability.

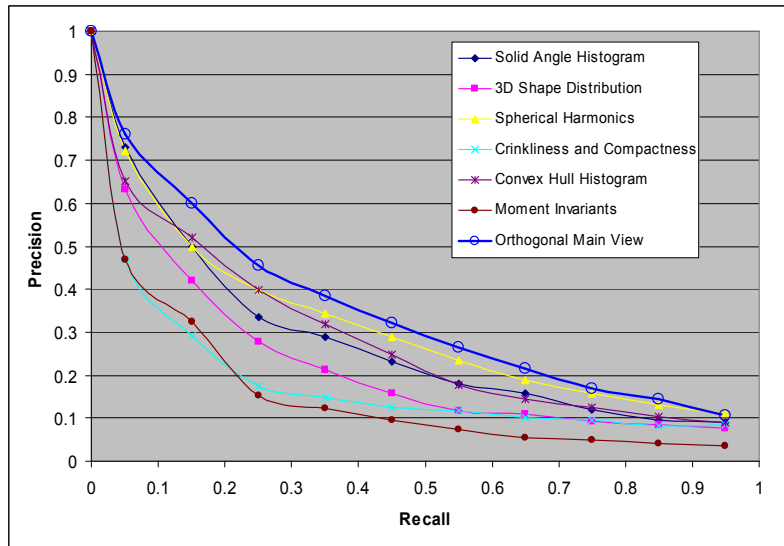




















Fig. 14. Recall-Precision curves for 3D model retrieval.

To explain the effects of the feedback interaction upon retrieval performance, we also list some experimental data. It shows that users' active interaction can improve the overall performance obviously. Tab.1. shows an example to explain the effect of weight adjustment.

7. CONCLUSION

This paper describes a 2D sketch user interface for 3D model retrieval, in which several feedback ways are provided to users to refine the retrieval performance interactively. This kind of interaction paradigm can provide higher retrieval accuracy and less time than query by example, which is a commonly used approach. Preliminary experiments show that interactive feedbacks can improve the performance of 3D model retrieval further. In the future, we will parameterize the sketches and conduct extensive experiments to verify the feedback interaction and parameterization.

Front view: 	$w_y = 0.33$ $w_z = 0.33$ $w_x = 0.33$					
Top view: 	$w_y = 0.61$ $w_z = 0.26$ $w_x = 0.13$					
Side view: 	$w_y = 0.25$ $w_z = 0.15$ $w_x = 0.60$					

Tab. 1. A weight adjustment example

8. REFERENCES

- [1] Elad, M., Tal, A. and Ar, S., Content based retrieval of VRML objects: an iterative and interactive approach, in *Proceedings of 6th Eurographics Workshop on Multimedia 2001* (Manchester, UK, 2001), pp 107-118.
- [2] Motofumi, T. S. A Web-based Retrieval System for 3D Polygonal Models, in *Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference* (Vancouver, 2001), pp 2271-2276.
- [3] Zhang, C. and Chen, T. Indexing and retrieval of 3D models aided by active learning, in *Proceedings of ACM Multimedia 2001* (Ottawa, Ontario, Canada, 2001), pp615-616.
- [4] Vranic, D.V. and Saupe, D. Description of 3D-shape using a complex function on the sphere, in *Proceedings of IEEE International Conference on Multimedia and Expo* (Lausanne, Switzerland, 2002), pp 177-180.
- [5] Kazhdan, M., Chazelle, B., Dobkin, D., Funkhouser, T. and Rusinkiewicz, S., A reflective symmetry descriptor for 3D models, *Algorithmica*, Vol.38, No.2, 2003, pp 201-225.
- [6] Mihael, A., Gabi, K., Hans-Peter, K. and Thomas, S. 3D shape histogram for similarity search and classification in spatial databases, in *Proceedings of 6th International Symposium on Spatial Databases* (HongKong, China, 1999), pp 207-228.
- [7] Paquet, E. and Rioux, M. Nefertiti: a tool for 3D shape databases management. *SAE Transactions on Aerospace*, Vol.108, 2000, pp 387-393.
- [8] Robert, O., Thomas, F., Bernard, C. and David, D. Shape distribution. *ACM Transactions on Graphics*, Vol.21, No.4, 2002, pp 807-832.
- [9] Hilaga, M., Shinaagagawa, Y., Kohmura, T. and Kunii, T.L. Topology matching for fully automatic similarity estimation of 3D shapes, in *Proceedings of SIGGRAPH 2001* (Los Angeles, USA, 2001), pp 203-212.
- [10] Su, L.S. and Lee, T.Y. 3D model retrieval using geodesic distance, in *Proceedings of Computer Graphics Workshop* (Taiwan, 2002), pp 16-20.
- [11] Siddiqi, K., Shokoufandeh, A., Dickinson, S. and Zucker, S. Shock graphs and shape matching. *Computer Vision*, Vol.35, No.1, 1999, pp 13-20.
- [12] Sundar, H., Silver, D., Gagvani and Dickinson, S. Skeleton based shape matching and retrieval, in *Proceedings of Shape Modeling International 2003* (Seoul, Korea, 2003), pp 130-142.
- [13] Christopher, M.C. and Benjamin, B.K. 3D object recognition using shape similarity-based aspect graph, in *Proceedings of 8th International Conference On Computer Vision* (Vancouver, Canada, 2001), pp 254-261.
- [14] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., and Jacobs, D., A search engine for 3D models. *ACM Transactions on Graphics*, Vol.22, No.1, 2003, pp 83-105.
- [15] CADlib Web Based CAD Parts Library, <http://www.cadlib.co.uk>.
- [16] MeshNose, the 3D objects search engine, <http://www.deepfx.com/meshnose>.
- [17] National Design Repository, <http://www.designrepository.org>.
- [18] Pu, J.T., Liu, Y., Xin, G.Y., Zha, H.B, Liu, W.B. and Yusuke U., 3D model retrieval based on 2D slice similarity measurements, in *Proceedings of 2th International Symposium on 3D Data Processing, Visualization & Transmission* (Thessaloniki, Greece, 2004), pp 75-83.
- [19] <http://trec.nist.gov/pubs/trec13/appendices/CE.MEASURES.pdf>, in *Proceedings of the Thirteenth Text Retrieval Conference*, TREC 2004.
- [20] Cybenko, G., Bhasin, A., and Cohen, K., Pattern recognition of 3D CAD objects. *Smart Engineering System Design*, Vol. 1, 1997, pp1-13.
- [21] Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S., Rotation invariant spherical harmonic representation of 3D shape descriptors, in *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (Aachen, Germany), 2003, pp 156-164.
- [22] Rea, H., Corney, J., Clark, D., Pritchard, J., Breaks, M., and MacLeod, R., Part sourcing in a global market, in *Proceedings of ICeCE' 01* (Beijing) China Machine Press, 2001, pp 192-201.
- [23] Barber, B.C., Dobkin, D., and Huhdanpaa, H., The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, Vol. 22, No. 4, 1996, pp 469-483.
- [24] Kriegel H.-P., Kröger P., Mashaël Z., Pfeifle M., Pötke M., Seidl S., Effective similarity search on voxelized CAD objects, in *Proceedings of 8th International Conference on Database Systems for Advanced Applications* (Kyoto, Japan), 2003, pp 27-36.