Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Multi-level spherical moments based 3D model retrieval*

LIU Wei†, HE Yuan-jun

(*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*)

†E-mail: liu-wei@sjtu.edu.cn

Received May 20, 2006; revision accepted June 21, 2006

Abstract: In this paper a novel 3D model retrieval method that employs multi-level spherical moment analysis and relies on voxelization and spherical mapping of the 3D models is proposed. For a given polygon-soup 3D model, first a pose normalization step is done to align the model into a canonical coordinate frame so as to define the shape representation with respect to this orientation. Afterward we rasterize its exterior surface into cubical voxel grids, then a series of homocentric spheres with their center superposing the center of the voxel grids cut the voxel grids into several spherical images. Finally moments belonging to each sphere are computed and the moments of all spheres constitute the descriptor of the model. Experiments showed that Euclidean distance based on this kind of feature vector can distinguish different 3D models well and that the 3D model retrieval system based on this arithmetic yields satisfactory performance.

Key words: 3D model retrieval, Spherical moments, Feature extraction, Pose normalization
doi:10.1631/jzus.2006.A1500 **Document code:** A **CLC number:** TP391

INTRODUCTION

Recently, the development of 3D modelling and digitizing technologies has made the model generating process much easier. Also, through the Internet, users can download a large number of free 3D models from all over the world. All these lead to the necessity of a 3D model retrieval system. Content-based 3D shape retrieval for broad domains like World Wide Web has recently gained considerable attention in the Computer Graphics community (Zhang *et al*., 2001; Patrick, 2004). One of the main challenges in this context is the mapping of 3D object into compact canonical representations referred to as descriptor or feature vector, which serve as search keys during the retrieval process. The descriptor decisively influences the performance of the search engine in terms of computational efficiency and relevance of the results. In this paper, we propose an improved method to extract the feature vector of the 3D model based on multi-level spherical moments, with experiments showing that this method can distinguish well different models and yield comparatively perfect performance.

The outline of the rest of this paper is as follows. In the next section we review relevant previous work. In Section 3 we describe a method for descriptor extraction based on multi-level spherical moments. In Section 4 experiments and results are presented in detail. Finally we give the conclusions and describe future work in Section 5.

PREVIOUS WORK

Paquet *et al*.(1997; 1998) made the initial research and got remarkable results in the 1990s. From then on, many researches have been carried out and various approaches have been proposed. In general, they can be divided into four types: shape-based retrieval, topology-based retrieval, image-based retrieval and surface-attributes-based retrieval. Ankerst *et al*.(1999) directly syncopated 3D model with some mode, then calculated the proportions of points

* Project (No. 60573146) supported by the National Natural Science Foundation of China

number of each unit to that of the whole model, thus shape histograms were formed. In his paper, Anderst introduced three methods to syncopate 3D models: shell model, sector model and spider Web model. So we can retrieve model through comparison of shape histograms. Suzuki *et al*.(2000) put forward a different method called point density to syncopate 3D models. This method did not form feature vector simply by syncopated units but classified the syncopated units, so the feature vector dimension and computational quantity decreased greatly. Osada *et al*.(2002) investigated a 3D model retrieval system based on shape distributions. The main idea was to calculate and get large numbers of statistical data which could serve as shape distributions to describe the features of models. The key step was to define the functions which could describe the models. He defined five simple and easy-to-compute shape functions: *A*3, *D*1, *D*2, *D*3 and *D*4. Because this method was based on a large number of statistical data, it was robust to noise, resampling and predigestion. Vranic and Saupe (2001a)'s method first voxelized the 3D model, then applied 3D Fourier transform on these voxelizations to decompose the model into different frequencies, finally chose certain number of coefficients of frequency as this model's feature vector. Another approach investigated by Vranic and Saupe (2001b; 2002) was spherical harmonics analysis, also called 2D Fourier transform on unitary sphere. This method needed sampling and harmonics transform, so the process of feature extraction was slow. Chen and Ouhyoung (2002a) developed a Web-based 3D model retrieval system in which a topology method using Reeb graph (Hilaga *et al*., 2001) was introduced which can be constructed to different precisions, so that in this way multi-resolution retrieval was available. There are also many researches on image-based retrieval (Jobst, 2000; Min *et al*., 2002; Chen *et al*., 2003) of which Chen at Taiwan University made outstanding contribution. In Chen *et al*.(2003)'s system, several images taken from different views around, are used to represent a 3D model, which are then transformed into descriptors, after which the matching of any two 3D models is achieved by matching two groups of 2D images by those descriptors. One hundred images belonging to 10 light-fields for each model are required, with the calculation time being enormous.

OUR METHOD

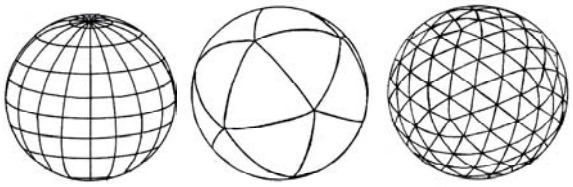
The main idea of our method is to sample at the 3D model surfaces and calculate the moments of the sampling points, which together server as the feature vector of the model. The more sampling points we take, the more accurately the points could approach the original model. A common method is that the 3D model is placed in a spherical coordinate frame with a sampling step following. When sampling, similar to the ray casting method, a collection of rays are projected in the directions per longitude and latitude: (θ_i,φ_j), $\theta_i=(i+0.5)\frac{\pi}{N}$, $\varphi_j=(j+0.5)\frac{2\pi}{N}$ (i, j=0, 1, …, N−1). And $\boldsymbol{u}_{ij}=(x_{ij}, y_{ij}, z_{ij})=(\cos\theta_i\cos\varphi_j, \cos\theta_i\sin\varphi_j, \sin\theta_i)$ is the corresponding point of intersection. Then we can define $M^{p,q,r}=\sum_{i,j=0}^{N-1} r(\boldsymbol{u}_{ij})x_{ij}^p y_{ij}^q z_{ij}^r$ (p, q, r=0, 1, 3, …) as the (p, q, r) ($0\le p+q+r\le m$) rank moment while $r(\boldsymbol{u}_{ij})$ is the distance between the sampling point $\boldsymbol{u}_{ij}$ and the center of the model. As m increased from 2 to 6 the dimension of the corresponding feature vectors increased from 10 to 20, 35, 56 and 84 (the dimension was $(m+1)(m+2)(m+3)/6$). But using this kind of moment will yield only imperfect performance, mainly because if m was too small, the feature vector was not discriminating enough for the retrieval; on the other hand if m was too big, the feature vector was unsteady since any tiny variation on the surface would make the high-level moments change drastically. Another problem was that the sampling was not uniform which will be explained in the following subsection.

The main aim of the paper is to solve the two problems mentioned above. For the first problem, we can adopt multi-level moments instead of a single series of moments. The main idea is to map the 3D model onto several concentric spheres and then to calculate the moments of all the spherical surfaces. The advantage is obvious: To avoid high-level moments while having enough number of moments to represent the feature of 3D model. As for the second problem, we change the sampling method and make the sampling points distribute more uniformly. In a later subsection, we will expatiate it in detail.

Sampling on the spherical surface

As mentioned above, the sampling points ob-

tained by spherical coordinate are in fact not uniform at all, as shown in Fig.1a, the points of intersection are just the sampling points. As we know that regular polyhedra are uniform and have facets which are all of one kind of regular polygon and thus better tessellations may be found by projecting regular polyhedra onto the unit sphere after bringing their center to the center of the sphere. A division obtained by projecting a regular polyhedron has the desirable property that the resulting cells all have the same shapes and areas (Horn, 1984). Also, all cells have the same geometric relationship to their neighbors. So if we choose the centers of the cells as the sampling points, their distributing will satisfy the requirement of uniformity. Unfortunately there are only five regular polyhedra: tetrahedron, hexahedron, octahedron, dodecahedron, and icosahedron. And even the icosahedron, with twenty triangular cells, provides too coarse sampling, as shown in Fig.1b. If we desire still finer sampling, splitting each facet of a given tessellation further into more triangular facets is a possible way. For example we can divide the triangular cells into four smaller triangles according to the well known geodesic dome constructions. Fig.1c is the subdivision of an icosahedron. In this case the icosahedron has been subdivided twice, so now it has 20×4×4=320 spherical triangles, that is to say we can get 320 sampling points by this means, which is adequate to represent the distributing situation of one radius.

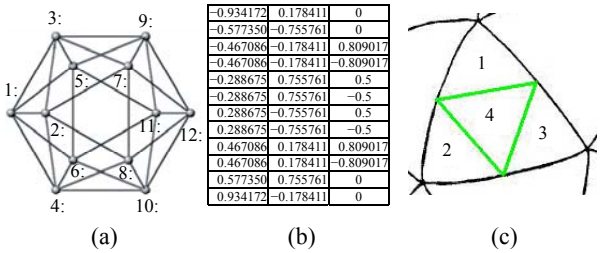


(a) (b) (c)

Fig.1 Sampling on the spherical surface. (a) Sampling based on spherical coordinate; (b) Tessellations of the sphere using regular icosahedron; (c) Tessellations of the sphere using a frequency four geodesic tessellation based on the icosahedrons

To see the distributing uniformity of these sampling points, we must compute the maximal difference among the areas of these triangular facets. Obviously, they are all spherical equilateral triangles. The calculation obtained is simplified.

Fig.2a is an observation of the icosahedron with radius of its circumcircle being 1 using the mode of parallel projection, and Fig.2b shows the coordinates of its twelve vertices. As all the cells are the same and have the same geometric relationship to their neighbors, we need only choose one cell on the spherical surface divided by the icosahedron as an example to discuss, as shown in Fig.2c. Obviously, the triangles with their serial number being respectively 1, 2, 3 have the same areas, so we just calculate the difference between them and the 4th triangle.



−0.934172	0.178411	0
−0.577350	−0.755761	0
−0.467086	−0.178411	0.809017
−0.467086	−0.178411	−0.809017
−0.288675	0.755761	0.5
−0.288675	0.755761	−0.5
0.288675	−0.755761	0.5
0.288675	−0.755761	−0.5
0.467086	0.178411	0.809017
0.467086	0.178411	−0.809017
0.577350	0.755761	0
0.934172	−0.178411	0

(a) (b) (c)

Fig.2 (a) Icosahedron; (b) Coordinates of icosahedron's twelve vertices; (c) Divide the triangular cells into four smaller triangles according to the well known geodesic dome constructions

Fig.2a shows that the 1st, 2nd, 3rd vertex can be used to construct a spherical triangle on its circumcircle, so we can suppose that the spherical triangle in Fig.2c is just the one. So the coordinates of this spherical triangle are respectively (−0.934172, 0.178411, 0), (−0.57735, −0.755761, 0) and (−0.467086, −0.178411, −0.809017). We can easily calculate the coordinates of the 4th triangle shown in Fig.2c, which are respectively (−0.934172, −0.178411, 0), (−0.866025, 0, 0.5) and (−0.645641, −0.577266, −0.499927). The area of the 4th spherical triangle in Fig.2c is then 0.159 and that of the other three is all 0.157. There is only 1.27% difference between them and if we split it once more the difference will be slighter, so we can regard them as equal. Since the sampling points are the centers of these triangles, we can assume that the disturbing is approximately uniform.

Mapping the 3D model onto a series of concentric spheres

From Subsection 3.1, we get an approximately uniform sampling, so that we can now see the surface of the sphere as a spherical image with each triangle being a pixel. The next step is to map the 3D model

onto a series of concentric spheres. A simple method is to directly calculate the intersection between the 3D model and the concentric spheres, that is to say if a point on the surface of the 3D model falls into the trigonal pixel, the value of this pixel is 1, otherwise 0. But this method has two shortcomings: first, the calculation is very complicated as the surface of the 3D model is irregular; second, only the triangles on the 3D model which intersect these spheres have effect on the resulting spherical images, an inevitable consequence of which is that this method may not be robust when the model is remeshed or in difference LOD. To overcome these two shortcomings, we adopt the following three steps to achieve the aim: (1) Pose normalization; (2) Voxelization; (3) Mapping.

1. Pose normalization

3D models have arbitrary position, orientation and scaling in 3D space. In order to capture its invariant feature, a feasible scheme is to place the model in a canonical coordinate frame to get the pose normalized. Then, if a model is scaled, translated or rotated, the placing into the canonical frame would be still the same which makes the moments comparable since the extracted feature is not invariant to position, orientation and scaling.

The pose normalization step is done through PCA (Principal Component Analysis) also known as Karhunen-Loeve transform (Chen and Chen, 2002; Chen and Ouhyoung, 2002b; Tangelder and Veltkamp, 2003). Let $P=\{\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_n\}$ ($\boldsymbol{P}_i=(x_i, y_i, z_i)\in\mathbb{R}^3$, $i=1,2,\ldots,n$) be the set of vertices on the surface of this model. The goal of pose normalization is to find an affine map: $\tau: \mathbb{R}^3\rightarrow\mathbb{R}^3$ in such a way that a model of any translation, rotation and scaling can be put normatively by this transformation. The translation invariance is accomplished by finding the barycenter of the model:

$$x_c=\frac{1}{n}\sum_{i=1}^{n}x_i,\ y_c=\frac{1}{n}\sum_{i=1}^{n}y_i,\ z_c=\frac{1}{n}\sum_{i=1}^{n}z_i.$$

So the point $\boldsymbol{P}_c(x_c, y_c, z_c)$ is the center of this model, we move $\boldsymbol{P}_c$ to coordinate origin. Thus for each point $\boldsymbol{P}_i=(x_i, y_i, z_i)\in P$, a corresponding transformation $\boldsymbol{P}_i'=(x_i-x_c, y_i-y_c, z_i-z_c)$ is performed. Based on the transformation we define points set $P'=\{\boldsymbol{P}_1', \boldsymbol{P}_2', \ldots, \boldsymbol{P}_n'\}$. Then we calculate the covariance matrix $\boldsymbol{C}$:

$$\boldsymbol{C}=\sum_{k=1}^{n}\boldsymbol{P}_k^{\mathrm{T}}\boldsymbol{P}_k=\begin{bmatrix}\sum_{k=1}^{n}x^2 & \sum_{k=1}^{n}xy & \sum_{k=1}^{n}xz\\ \sum_{k=0}^{n}xy & \sum_{k=1}^{n}y^2 & \sum_{k=1}^{n}yz\\ \sum_{k=1}^{n}xz & \sum_{k=1}^{n}yz & \sum_{k=1}^{n}z^2\end{bmatrix}.$$

This step must be done after the translation of $\boldsymbol{P}_c$ to coordinate origin. Obviously the matrix $\boldsymbol{C}$ is a real symmetric one, therefore its eigenvalues are non-negative real numbers. Then we sort the eigenvalues in non-increasing order and find the corresponding eigenvectors. The eigenvectors are scaled to Euclidean unit length and we form the rotation matrix $\boldsymbol{R}$ which has the scaled eigenvectors as rows. We rotate all points in P' and a new point set is formed:

$$P''=\{\boldsymbol{P}_i''\mid \boldsymbol{P}_i''=\boldsymbol{P}_i'\boldsymbol{R}, \boldsymbol{P}_i'\in P',\ i=0,\ \ldots,\ n).$$

When this step has been done, the model is rotated so that the x-axis maps to the eigenvector with the biggest eigenvalue, the y-axis maps to the eigenvector with the second biggest eigenvalue and the z-axis maps to the eigenvector with the smallest eigenvalue.

But this method often does not provide robust normalization in many cases. We must notice that the distribution of points may not be uniform, for example, a large quadrangle in the surface of a model can be decomposed into two triangles and four points while some fine detail needs hundreds of triangles and points. Thus the result of pose normalization will not be perfect. We can easily modify the method described above. A feasible way is that the points set P is taken uniformly from the surface of a model with Montecarlo arithmetic (Osada *et al*., 2001) instead of the original set of vertices. Fig.3 is an example of a 3D model before and after PCA.

2. Voxelization

After the model has been pose-normalized, the following step is to rasterize its surface into a $2N\times2N\times2N$ voxels grid, assigning a voxel a value of 1 if it was within one voxel of a point on the boundary, and a value of 0 otherwise. The model is now composed of regular voxels. At the same time it is aligned

so that its center of mass is at the center of the grid, and so that its bounding sphere has radius N, as shown in Fig.4.

(a) (b) (c)

Fig.3 An effect of PCA. (a) Original model; (b) Uniform sampling; (c) Model after PCA

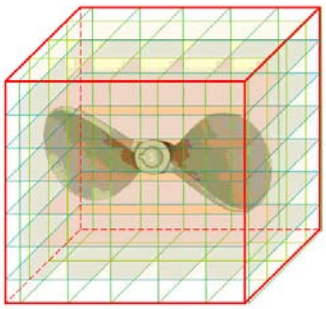


Fig.4 Rasterization of the 3D model

3. Mapping

Now we can use N concentric spheres with their radii being from 1 to N to syncopate the voxelized 3D model, as shown in Fig.5. If one of these voxels intersects a trigonal pixel described above on these concentric spheres, then the value of this pixel is 1, otherwise 0. Thus N spherical images are formed.

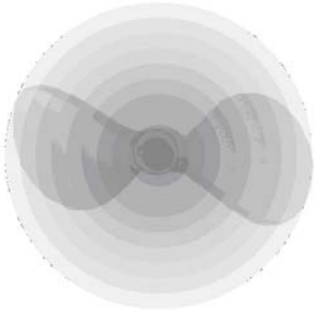
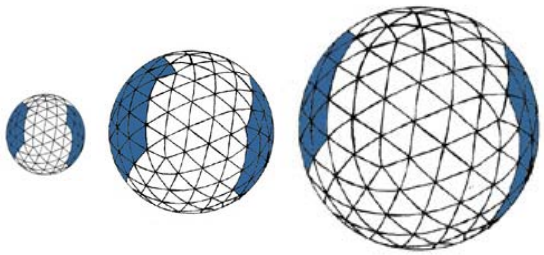


Fig.5 Syncopation of 3D model

The advantage of this method is two fold. First, the computation is reduced as the trigonal pixels and the cubic voxels are all regular, the process of intersection is greatly simplified. Second, the voxels have certain volume, so slight change on 3D model will not result in great difference on the spherical images, that is to say, this method is more robust than the direct one as described above. Fig.6 shows three spherical images with different radii.



(a) (b) (c)

Fig.6 Spherical images N=32. (a) R=5; (b) R=10; (c) R=15

Feature extraction

Now that we have obtained N spherical images, the next step is to calculate their moments and extract the feature vectors based on these moments. For each image, we calculate the moments as following: $sm^{pqr}=\sum_{i=0}^{K-1}\frac{gray_i x_i^p y_i^q z_i^r}{R^{p+q+r}}$ $(0\le p+q+r\le m)$. K is the count of trigonal pixels on a sphere image, *gray* is the value of pixels, (x_i, y_i, z_i) is the center of the ith pixel while R is the radius of this sphere. The denominator R^{p+q+r} makes the moments independent from radius so all spherical images with different radii play a coequal role. Then for each sphere there are $(m+1)(m+2)\times(m+3)/6$ moments, so we can construct a feature vector of the model which has $N(m+1)(m+2)(m+3)/6$ dimensions by using moments of all the spheres. The similarity of the two models can be realized by comparing the feature vectors using Euclidean distance.

Since different spherical moments represent different physical meanings, they have different scopes of values and cannot be compared directly. We must place these moments at the same status so that they play the same roles in comparison to feature vectors. To achieve this aim, a method based on Gauss unitization is used. The steps are as follows:

(1) Calculate the average value sm_{aver} of the $N(m+1)(m+2)(m+3)/6$ spherical moments;

(2) Calculate the covariance σ of the $N(m+1)\times(m+2)(m+3)/6$ spherical moments;

(3) Then we use $\left(\frac{sm_i - sm_{\text{aver}}}{3\sigma}+1\right)\Big/2$ to replace sm_i itself.

Provided that these spherical moments accord with Gauss distribution, then for each spherical moment sm_i (i=0, 1, …, $N(m+1)(m+2)(m+3)/6-1$), the probability that the moment falls into the range of [0,1] is over 99%.

EXPERIMENTS AND RESULT

According to the method described in Section 3, we have developed a 3D mechanical parts retrieval system using Visual C++ on a PC with Pentium IV 1.8 G CPU and Windows 2000 Server Operation System consisting of two parts: off-line process module and on-line retrieval process module, as shown in Fig.7. In the off-line process, features of 3D models are extracted and stored in database. To efficiently search a large collection online, an indexing data structure and searching algorithm is available with B+ Tree. In the on-line process, users select a 3D part as the query mode, then system engine compares the distances between the query and the models in the parts library, so that finally several models most similar to the input one are returned.

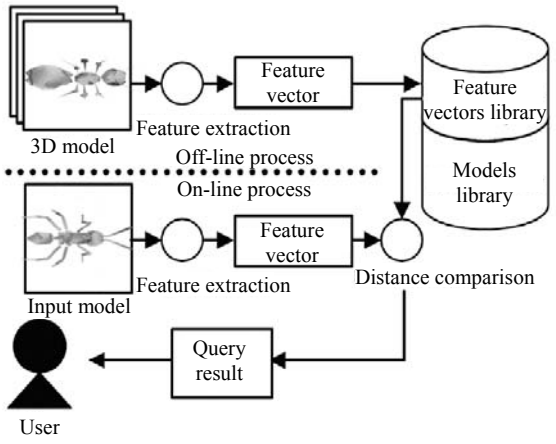


Fig.7 Architecture of 3D model retrieval system

For the experiments, we have collected 965 models from Web in total with 2341 points and 3981 triangles per model on average. The models are not all well defined trigonal meshes, many of them contain cracks, self-intersections, missing polygons, etc. To fairly test the performance, we use the Princeton Shape Benchmark (PSB), which is a defacto criterion of 3D model database and is introduced in (Shilane *et al*., 2004) in detail, to test our system. It is a collection of 1814 models which are manually divided into 161 classes, each containing at least 4 models. The performance can be decided by two measures: *precision* and *recall*. "*precision*" measures the system ability to retrieve only models that are relevant while "*recall*" measures the system ability to retrieve all models that are relevant. Let C be the number of relevant models in the database, containing the number of models of the class to which the query model belongs. Let N be the number of relevant models that are actually retrieved in the top A retrievals. Then, *recall* and *precision* are defined as follows: $precision=N/A$, $recall=N/C$.

In this method, we need choose two appropriate parameters: the dimension of voxel grid N and the upper limit of moment rank m.

As for N, if N is small, the partition is too coarse and large distortion appears; on the other hand, if N is big, the computation is large though the voxelized model can commendably approximate the original model. The same discussion is applicable to m, since the number of moments belonging to one spherical image accords with $O(m^3)$ complexity, so m cannot be too big, as at the same time high-rank moments are unstable. In our experiments we tested 12 cases, in which N={16, 24, 32, 40} and m={1, 2, 3}, thus the dimension of the feature vector ranges from 64 to 800. For each case, we tested various kinds of models including animals, plants, cars, characters, and so on; from each kind we choose ten models as the query models and test the results. Results show that when N=32 and m=2 the arithmetic yields optimal performance, and in this case the dimension of the feature vector is 320, and Fig.8 is the corresponding Precision-Recall (P-R) plot. Fig.9 shows a comparatively successful retrieval of dogfish, and the first model is the input one.

To test the time for feature extraction, we use mesh simplification tool QSlim developed by Garland and Shaffer (2002), to make the 3D model composed of about 1000, 5000 and 9000 triangles respectively while remaining the basal shape. We tested 100 models in the database, with the average time consumed being presented in Table 1.

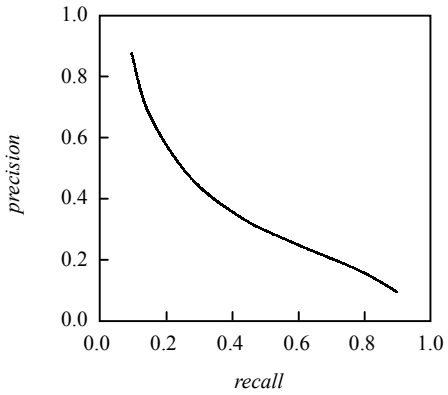


Fig.8 P-R plot when *N*=32 and *m*=2

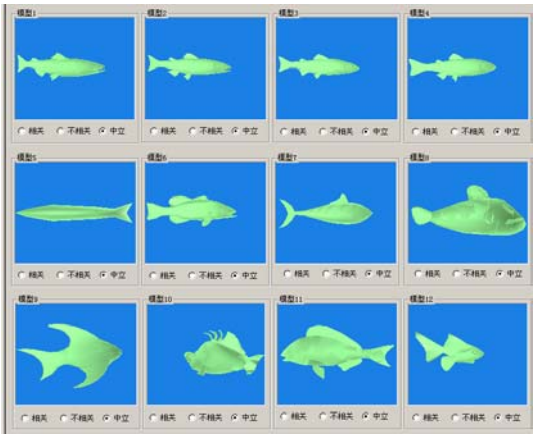


Fig.9 A retrieval of dogfish

Table 1 Time costs

Triangles	Time (s)
1000	0.4
5000	2.1
9000	4.1

CONCLUSION AND FUTURE WORK

In this paper, we propose a multi-level spherical moment method for 3D model retrieval. For more uniform sampling, we constructed a segmentation of spherical surface based on icosahedron. Then we voxelized the 3D model to make the process of mapping easier and to improve its robustness. Experiments showed that our method yields fairish performance while costing acceptable time.

As for further work, we plan to orderly organize and delaminate the models library to store similar models in the same categories, so the system can quickly eliminate the obvious dissimilar models and accelerate the retrieval rate. We consider using Artificial Neural Network to achieve this so that each input model can automatically find its category and be stored in corresponding position.

References

Ankerst, M., Kastenmller, G., Peter, K.H., 1999. 3D Shape Histograms for Similarity Search and Classification in Spatial Database. Proceeding of the 6th International Symposium on Large Spatial Database, Hong Kong, China, p.207-226.

Chen, S.C., Chen, T., 2002. Retrieval of 3D Protein Structure. Proceedings of International Conference on Information Processing (ICIP 2002), Rochester, NY, p.34-43.

Chen, D., Ouhyoung, M., 2002a. A 3D Object Retrieval System Based on Multi-Resolution Reeb Graph. Proceedings of Computer Graphics Workshop, Taiwan, China, p.16-20.

Chen, D., Ouhyoung, M., 2002b. A 3D Model Alignment and Retrieval System. Proceedings of International Computer Symposium, Workshop on Multimedia Technologies, Hualien, Taiwan, p.1436-1443.

Chen, D., Tian, X., Shen, Y., 2003. On visual similarity based 3D model retrieval. *Computer Graphics Forum*, **22**(3): 223-232. [doi:10.1111/1467-8659.00669]

Garland, M., Shaffer, E., 2002. A Multiphase Approach to Efficient Surface Simplification. Proceedings of IEEE Visualization, Boston, MA, p.117-124.

Hilaga, M., Shinagawa, Y., Kohmura, T., 2001. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Los Angeles, CA, p.203-212.

Horn, B., 1984. Extended Gaussian images. *Proceedings of the IEEE*, **72**(12):1671-1686.

Jobst, L., 2000. Content-Based Retrieval of 3D Models in Distributed Web Databases by Visual Shape Information. IEEE International Conference on Information Visualization, London, UK, p.82-87.

Min, P.K., Chen, J., Funkhouser, T., 2002. A 2D Sketch Interface for a 3D Model Search Engine. Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH 2002 Technical Sketch, San Autonio, Texas, p.334-341.

Osada, R., Funkhouser, T., Chazelle, B., 2001. Matching 3D Models with Shape Distributions. Proceedings International Conference on Shape Modeling and Applications SMA-01, Genoa, Italy, p.154-166. [doi:10.1109/SMA.2001.923386]

Osada, R., Funkhouser, T., Chazelle, B., 2002. Shape distributions. *ACM Trans. on Graphics*, **21**(4):807-832. [doi:10.1145/571647.571648]

Patrick, M., 2004. A 3D Model Search Engine. Ph.D Thesis,

Princeton University.

Paquet, E., Rioux, M., 1997. A Query by Content Software for Three Dimensions Database Management. Proceedings International Conference on Recent Advances in 3-D Digital Imaging and Modeling (Cat No 97TB100134) IM-97, Ottawa, Canada, p.345-352. [doi:10.1109/IM.1997.603886]

Paquet, E., Rioux, M., 1998. A Content-based Search for VRML Database. Proceedings 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat No 98CB36231) CVPR-98, Santa Barbara, CA, p.541-546. [doi:10.1109/CVPR.1998.698658]

Shilane, P., Min, P., Kazhdan, M., Funkhouser, T., 2004. The Princeton Shape Benchmark. Shape Modelling International, Genova, Italy, p.373-385.

Suzuki, M.T., Kato, T., Otsu, N., 2000. A Similarity Retrieval of 3D Polygonal Model Using Rotation Invariant Shape Descriptors. Proceeding of IEEE International on Systems, Man, and Cybernetics, Nashville, Tennessee, p.2946-2952.

Tangelder, J.W., Veltkamp, R.C., 2003. Polyhedral model retrieval using weighted point sets. *International Journal of Image and Graphics*, **3**(1):209-229. [doi:10.1142/S021946780300097X]

Vranic, D.V., Saupe, D., 2001a. 3D Shape Descriptor Based on 3D Fourier Transform. Proceeding of the EURASIP Conference on Digital Signal Processing of Multimedia Communications and Services, Budapest, Hungary, p.271-274.

Vranic, D.V., Saupe, D., 2001b. Tools for 3D Object Retrieval: Karhunen-Loeve Transform and Spherical Harmonics. IEEE Workshop on Multimedia Signal Processing (MMSP'2001), Cannes, p.293-298.

Vranic, D.V., Saupe, D., 2002. Description of 3D Shape Using a Complex Function on Sphere. Proceedings IEEE International Conference on Multimedia and Expo ICME-02, Lausanne, Switzerland, p.177-180. [doi:10.1109/ICME.2002.1035747]

Zhang, C., Chen, T., 2001. Efficient Feature Extraction for 2D/3D Objects in Mesh Representation. ICIP 2001.