Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 19 (2009) 495–499
Progress in Natural Science
www.elsevier.com/locate/pnsc

Content-based similarity for 3D model retrieval and classification

Ke Lü [a,*], Ning He [b], Jian Xue [a]

[a] College of Computing & Communication Engineering, Graduate University of Chinese Academy of Sciences, Yuquan Road 19(A), Beijing 100049, China
[b] School of Mathematical Science, Capital Normal University, Beijing 100037, China

Received 2 April 2008; received in revised form 4 June 2008; accepted 4 June 2008

Abstract

With the rapid development of 3D digital shape information, content-based 3D model retrieval and classification has become an important research area. This paper presents a novel 3D model retrieval and classification algorithm. For feature representation, a method combining a distance histogram and moment invariants is proposed to improve the retrieval performance. The major advantage of using a distance histogram is its invariance to the transforms of scaling, translation and rotation. Based on the premise that two similar objects should have high mutual information, the querying of 3D data should convey a great deal of information on the shape of the two objects, and so we propose a mutual information distance measurement to perform the similarity comparison of 3D objects. The proposed algorithm is tested with a 3D model retrieval and classification prototype, and the experimental evaluation demonstrates satisfactory retrieval results and classification accuracy.

Keywords: 3D model; Retrieval and classification; Content-based 3D model retrieval; Similarity comparison

1. Introduction

The development of modeling tools, such as 3D scanners and 3D graphics hardware (or hardware-accelerated 3D graphics), has enabled access to three-dimensional materials of high quality both on the Internet and in the domain-specific databases. 3D models now play an important role in many applications, such as mechanical manufacture, games, biochemistry, art, and virtual reality. How to find the desired models quickly and accurately from 3D model databases and how to classify the 3D models have become practical problems. Researchers in many well-known institutions and universities all over the world are dedicating themselves to this research field, which has led to the development of experimental search engines for 3D shapes [1,2], such as the 3D model search engine at Princeton University, and the 3D model retrieval system at the National Taiwan University. Several feature representations have been explored: Zhang produced a local index for volume index [3]; Funkhouser described objects using a reflective symmetry descriptor [4] or a spherical harmonics descriptor; and Hilaga proposed the method based on a Reeb graph [5]. Many other methods [6,7] could also be included in this list. As for the pattern recognition problem, 2D image classification and recognition has been widely considered, while less work has been done in the case of 3D models [8]. For classification, the statistical learning algorithm, support vector machine (SVM), is used to solve some practical problems [9].

In this study, we present new techniques for content-based 3D model retrieval and classification, in which the combination of a distance histogram and moment invariants is used to improve the retrieval and classification performance. SVM is employed to solve the 3D model classification problem.

* Corresponding author. Tel.: +86 10 88256595; fax: +86 10 88259429.
E-mail address: luk@gucas.ac.cn (K. Lü).

1002-0071/$ - see front matter © 2008 National Natural Science Foundation of China and Chinese Academy of Sciences. Published by Elsevier Limited and Science in China Press. All rights reserved.
doi:10.1016/j.pnsc.2008.06.025

2. A similarity measurement of a 3D model

2.1. Distance histogram for 3D model representation

A distance histogram measures the distances between fixed points and random points on a surface. Similar to the centroid-radii modeling of 2D shapes, we use the centroid of the model's boundary as the fixed point. After obtaining all the distances, we first need to divide them by the maximum distance, so as to normalize the distances. On completion of this process, the value of all the normalized distances will be in the range of $[0,1]$ and so the method is invariant to scale. Then we separate the range of the distances into several ranges, say R ranges, and compute the number of distances in each range.

The distance histogram can be considered to be a global descriptor for a 3D model. Rather than being attached to the details of the 3D model, distance histograms give more importance to its general aspects. The main idea is to focus on the statistical distributions of a shape function measuring geometrical properties of the 3D model. Despite its simplicity, the method has several properties that are desirable for similarity matching: the distance histogram has transformation invariance properties, random sampling ensures that the distance histogram is robust to noise, and construction of the distance histogram for a database of 3D models is generally fast and efficient. Furthermore, the distance histogram is independent of its presentation, topology, or the application domain of the sampled 3D models.

2.2. 3D moment invariants

Moments are a traditional mathematical tool for measuring the spatial mass distribution of a shape. In the case of binary digital datasets, this is the distribution of pixels (in 2D) or voxels (in 3D) of a shape. It is possible to compute moment invariants of a 3D point distribution which are invariant to translation and rotation, in the same manner as 2D moment invariants.

Let $\rho(x,y,z)$ be a local continuous density function. For example, this can be 1 inside voxels belonging to an object, and 0 in free space. Let $(\bar{x},\bar{y},\bar{z})$ be the centroid of the object. The use of centroid in the moment calculation given below gives translation invariance. The 3D moments of order $n=p+q+r,\ n\in N$ for a 3D density function $\rho(x,y,z)$ are defined as [10]:

$$\rho(x,y,z)=\begin{cases}1: & \text{Object}\\ 0: & \text{Background}\end{cases} \tag{1}$$

$$M_{pqr}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\bar{x})^p(y-\bar{y})^q(z-\bar{z})^r\rho(x,y,z)\mathrm{d}x\mathrm{d}y\mathrm{d}z \tag{2}$$

The set of moments $\{M_{pqr}\}$ has a fundamentally important property: it uniquely determines, and is uniquely determined by the object. The moments defined above contain both geometrical and internal information of the patterns. Approximating this formula to a digital voxel space is straightforward by using summation instead of integration.

In the absence of prior knowledge, 3D models have an arbitrary scale, orientation and position in the 3D space. Before we get the moment signature, we need to align the coordinates. Here, we consider the problem for a 2D image case and then extend the method to a 3D model.

The procedure for aligning the coordinates consists of the following three steps:

Step 1. Let μ_{kr} denote the central moments of order $(k+r)$ in the 2D case. The covariance matrix of a given image is defined as:

$$\boldsymbol{C}=\begin{bmatrix}\mu_{20} & \mu_{11}\\ \mu_{11} & \mu_{02}\end{bmatrix} \tag{3}$$

Step 2. Find the eigenvalues and eigenvectors of $\boldsymbol{C}$. Let λ_1 and λ_2 be the eigenvalues of $\boldsymbol{C}$; with $\boldsymbol{e}_1=[e_{1x}e_{1y}]^{\mathrm{T}}$ being the eigenvector associated with λ_1, and $\boldsymbol{e}_2=[e_{2x}e_{2y}]^{\mathrm{T}}$ associated with λ_2. Then we get the rotation matrix $\boldsymbol{E}$

$$\boldsymbol{E}=\begin{bmatrix}e_{1x} & e_{1y}\\ e_{2x} & e_{2y}\end{bmatrix} \tag{4}$$

Step 3. Transform the coordinate system by translating the origin to the image center and multiplying the coordinates by matrix $\boldsymbol{E}$. Thus, the new coordinates depend on the eigenvectors of $\boldsymbol{C}$. Let $[x'y']^{\mathrm{T}}$ denote the new coordinates. Then

$$\begin{bmatrix}x'\\ y'\end{bmatrix}=\boldsymbol{E}\begin{bmatrix}x-\bar{x}\\ y-\bar{y}\end{bmatrix} \tag{5}$$

In the case of binary digital datasets, this is the distribution of pixels in 2D for images and voxels in 3D for digital geometry objects. It is possible to compute moment invariants of the 3D point distribution which are invariant to translation and rotation, in the same manner as the 2D moment invariants. We thus apply the above procedure to the 3D case. Then the covariance matrix becomes

$$\boldsymbol{S}=\begin{bmatrix}M_{200} & M_{110} & M_{101}\\ M_{110} & M_{020} & M_{011}\\ M_{101} & M_{011} & M_{002}\end{bmatrix} \tag{6}$$

We should transform all the objects so that their centers are at $(0,0,0)$ and any dependence on translation or spatial position is eliminated. In order to make the final result unique, we further make sure that the third-order moments, M_{300} and M_{030}, are positive after the transformation. The above algorithm for pose estimation is fairly simple and efficient.

3. 3D model retrieval principle and multi-class support vector machine

3.1. 3D model retrieval principle

The choice of distance function can drastically influence the retrieval performance. We employ two distance measures, mutual information for the distance histogram and Euclidean distance for moments. Mutual information is an effective similarity measure for comparing images. The mutual information between two variables is a concept with roots in information theory and essentially measures the amount of information that one variable contains about the other [10]. As a similarity measure, it has many advantages. It assumes a statistical relationship that can be captured by analyzing the image joint entropy. Mutual information is closely related to joint entropy. Let X and Y be two n-bin histograms. Then the mutual information between X and Y can be defined as:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{7}$$

where $H(X)$ is the Shannon entropy of histogram X computed from the probability distribution of the bin counts, and $H(X, Y)$ is their joint entropy. Mutual information can be defined as the joint probability distribution of the histogram:

$$I(X;Y) = \sum_{x,y} p_{X,Y}(x,y) \log \frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)} \tag{8}$$

To estimate the joint probability $p_{X,Y}(x,y)$ for histograms X and Y, the most straightforward approach is to compute the co-occurrence matrix of the corresponding bin count values. The entries of the co-occurrence matrix record the number of times the bin counts in X having a value x coincide with the corresponding bin counts in Y having a value y. This is similar to the co-occurrence matrix used in texture characterization. The marginal distribution of $p_X(x)$ and $p_Y(y)$ can be obtained by summation over the rows or columns of the co-occurrence matrix. Here, we use the following information distance measure (MID):

$$MID(X,Y) = H(X,Y) - I(X;Y) \tag{9}$$

MID satisfies the axioms for a distance

$$MID(X,X) = 0$$
$$MID(X,Y) = MID(Y,X)$$
$$MID(X,Y) + MID(Y,Z) \geqslant MID(X,Z)$$

Euclidean distance is used to compare the similarity between moment features. Comparison can be done by calculating the distances of all the features and then by weighting the distances by the total distance.

3.2. Multi-class support vector machine

SVM was originally designed for binary classification. How to extend it for effective multi-class classification is still an on-going research issue. Currently, there are two main approaches for a multi-class SVM [11]. Different possibilities include modifying the design of the SVM to incorporate the multi-class learning directly in the quadratic solving algorithm. The schemes which have been proposed for solving the multi-class problem are as follows:

(i) Using k one-to-rest classifiers.
(ii) Using $k(k-1)/2$ pairwise classifiers with one of the voting schemes listed below:
Majority voting
Pairwise coupling.
(iii) Extending the formulation of the SVM to support the k-class problem.

Construct the decision function by considering all the classes at once.

Construct a decision function for each class by only considering the training data points belonging to that particular class.

Here, we choose one-to-one matching to perform the classification as it is more suitable for practical use. The one-to-one method constructs $k(k-1)/2$ classifiers where each one contains training data from two classes. Given l training data, (x_1, y_1), ..., (x_l, y_l), where $x_i \in R^n$, $i = 1$, ..., l and $y_i \in \{1,\ldots,k\}$ is the class of x_i, and the training data x_i are mapped to a higher dimensional space by the function ϕ and c is the penalty parameter, and $c\sum_{j=1}^{l}\xi_j^i$ is a penalty term that can reduce the number of training errors.

There are different methods for doing the test after all $k(k-1)/2$ classifiers are constructed. The following voting strategy is used: if the sign of $((w^{ij})^{\mathrm{T}}\phi(x) + b^{ij})$ denotes that x is in the ith class, then the vote for the ith class is incremented by one. Otherwise, the jth vote is incremented. The new prediction is that x is in the class with the largest vote. The voting approach described above is also called the max wins strategy. In the case that two classes have identical votes, we simply select the one with the smaller index. Practically we solve the dual problem whose number of variables is the same as the number of data in the two classes. The Gaussian RBF kernel is used which has been proved to provide good generalization capabilities.

The good performance is due to the superior generalization ability of SVM in high dimensional spaces. The dimensions of the feature vector affect the classification accuracy. When the feature vector dimension is 10, it makes the performance of classification worse. According to the error similarity, we choose the dimension of the feature vector to be 56: 40 for the histogram, and 16 for the moments (two and three order). The differences in classification granularity have an impact on the classification results. The database should not be too coarse or too fine. The 3D models are partitioned equally into the training and test sets.

4. Experiments and analyses

The method described above has been implemented using Visual C++ and tested on a 3D model database containing about 2500 models, some of which are models of the same object with different polygon tessellations. We manually arrange the database into different classes. The tests are run on a Windows-PC with SQL Server 2000. As for the query interface, we adopted the query-by-example approach. In the database, an entry stores a 3D model along with a pre-calculated feature vector for the model. Retrieval results are shown in Figs. 1 and 2. We carry out several experiments to find the number of random points that give perfect retrieval, where good retrieval performance can be determined by choosing 5000 random points.

We select about 1000 models for the retrieval and classification performance evaluation. Retrieval performance of the method is evaluated using the normalized recall (NR) measure [12] instead of precision vs recall, because NR reflects the position in which the set of relevant images appears in the retrieval sequence.

NR is formulated as AVRR/IAVRR, where AVRR is the average rank of all relevant, displayed images, and IAVRR is the ideal average rank which is maximum when all the relevant images are retrieved as shown in Fig. 2. Normally, NR is larger than 1. However, for ideal retrieval, NR is equal to 1. We get $\mathrm{NR} = 1.19$ when combining moments and a distance histogram, where $\mathrm{ANRS} = 1.32$ for the histogram and $\mathrm{ANRS} = 1.43$ for the moments. It can be seen that the distance histogram represents the distribution of the global features, while moment signatures represent the global features directly. The combination of the two features achieves better retrieval performance.

From the training sample, the first example (x_1, y_1) is removed, and the resulting sample is used for training, leading to a classification rule. This classification rule is tested on the removed example (x_1, y_1). This process is repeated for all the training examples. The number of misclassifications divided by n is the leave-one-out estimate of the generalization error. Table 1 shows the average error rates of the classification ($\sigma = 1/k$, where k is the number of classes and $c = 1000$).

Fig. 1. Models sorted by their similarity to the query object.

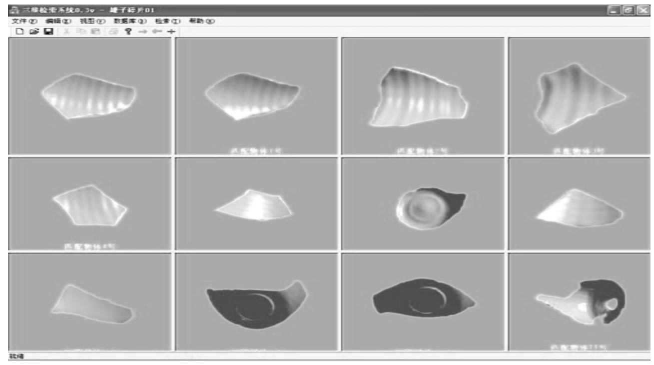

Fig. 2. Part matching retrieval result from 3D database.

Table 1
Average error rates of the classification (%).

Method	2-class	4-class	6-class
Distance histogram	19	22	23
Moment	20	24	29
Distance histogram and moment	13	15	19

The results show that the combination provides better results than using a single descriptor, because the method combines the shape features and the shape distribution features of a 3D model.

5. Conclusions

We have proposed a new algorithm for determining the content-based similarity of 3D models, given as a 3D representation. Two main issues are considered to measure the similarity of two models. The first is the selection of a distance function for the similarity measurement. The second is the multi-class SVM classification. We have developed a prototype of a 3D model retrieval and classification system. There are, however, still many open research issues that need to be solved, which are listed as follows:

(i) Many models look very similar from a visual appearance, but they belong to different classes. Therefore, special attention should be paid to the semantic issues when making a classification.
(ii) It is very difficult for one method to be applied to all 3D models. Therefore, it is expected that better retrieval performance will be achieved for larger databases and in application areas such as molecular biology, mechanical industry, and game characters. On the other hand, we should determine how well our method can discriminate classes of 3D models in a larger and more diverse database.
(iii) Performance evaluation criterion and a standard test dataset are needed. Any technique is advanced by its domain's evaluation criterion. Good metrics will lead the technique in the right direction, while bad ones

may mislead the research effort. An equally important task is to establish a well-balanced large-scale test dataset.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 60602062 and 60532080).

References

[1] Funkhouser T, Min P, Kazhdan M, et al. A search engine for 3D models. ACM Trans Graph 2003;22(1):83–105.
[2] Yang YB, Liu H, Zhu Q. Content based 3D model retrieval: a survey. J Comput 2004;27(10):1297–309, [in Chinese].
[3] Zhang C, Chen T. Efficient feature extraction for 2D/3D objects in mesh representation. In: Proceedings of the international conference on image processing; 2001. p. 935–8.
[4] Osada R, Funkhouser T, Chazell B. Matching 3D model with shape distributions. In: Proceedings of the international conference on shape modeling & applications, Genova, Italy; 2001. p. 154–68.
[5] Hilaga M, Shinagawa Y, Kohmura T. Topology matching for fully automatic similarity estimation of 3D shapes. In: Proceedings of ACM SIGGRAPH, Los Angeles, USA; 2001. p. 203–12.
[6] Bustons B, Keim DA, Saupe D. Feature-base similarity search in 3D object databases. ACM Comput Surv 2005;37(4):345–87.
[7] Besl PJ, Jain RC. Three-dimensional object recognition. Comput Surv 1985;17(1):75–145.
[8] Pontil M, Verri A. Support vector machines for 3-D object recognition. IEEE Trans Pattern Anal Mach Intell 1998;20(6):637–46.
[9] Vapnik VN. Statistical learning theory. New York: Wiley; 1998. p. 234–89.
[10] Mackay D. Information theory, interface, and learning algorithms. Cambridge: Cambridge University Press; 2003. p. 174–89.
[11] Weston J, Watkins C. Multi-class support vector machines. Technical report CSD-TR-98-04. Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, England; 1998. p. 230–9.
[12] Wei N, Geng GH, Zhou MQ. An overview of performance evaluation in content based image retrieval. J Image Graph 2004;9(11):1271–6, [in Chinese].