

Local Shape Similarity Retrieval of Mechanical Parts

Jiale Wang, Jun Yin

College of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, China

wjl8026@zjgsu.edu.cn, junyin2@zjgsu.edu.cn

Keywords: 3D model retrieval; 3D shape analysis; design reuse; part database

Abstract: A scale-invariant shape feature is proposed. The shape feature is extracted on gray voxel models by a SIFT-like algorithm to describe the local shape information of 3D models. Shape feature vectors are constructed by a BoF method. A distance function that supports the part-in-whole matching is employed to compute the dissimilarity between 3D models. Experiments demonstrate that the proposed method achieves a satisfactory retrieval performance.

Introduction

When designing new parts, engineers rarely start from scratch. Instead, they often search in enterprise database for suitable existing parts and modify them. One of the challenges to reusing design is how to search 3D models effectively. Conventionally, companies use Product Data Management (PDM) systems to manage their databases of legacy CAD models and related data. Traditional PDM system implements the access and retrieval of models by indexing keywords in filename, part numbers or annotations. This text-based method is not robust due to the following reasons: 1. There are many previous models that are not properly named. Manually annotating them is a tedious work. 2. A short piece of text is not descriptive enough. It may be too narrow or too wide to characterize a part. 3. With time and people changing, there may be inconsistent in naming scheme.

A better way to retrieve CAD models is the shape-based method: users upload an example model, and then system finds out similar ones from the database. This method has two main advantages over the text-based retrieval: 1. The shape of an object overpasses the barrier of language and naming differences. 2. Measure the similarity between CAD models by shape is the natural way that designers think. Instead of textural descriptions, what's initially in a designer's mind is just "something like this".

In recent years, 3D object searching methods have been explored in several disciplines such as computer vision, computer graphics, pattern recognition and CAD/CAM. Most previous studies focus on extracting a "shape signature" from 3D objects. A shape signature describes the shape information of a 3D object in compacted numerical format. The similarity between 3D objects can be measured by computing the distance between signatures under a predefined metric. According to the genre of shape information that a signature describes, shape signature can be generally classified as follows: geometry based [1, 2, 3], topology based [4, 5] and feature based [6, 7].

In this paper, we introduce a local shape feature for 3D model retrieval. This local shape feature is extracted on gray voxel models by a SIFT-like algorithm. Shape feature vectors are constructed by a BoF method and compared by a distance function that supports the part-in-whole matching. Experiments demonstrate that the proposed method achieves a satisfactory retrieval performance.

Voxelization

3D models may be in different representing format, such as polygon mesh (mesh), Boundary Representation (B-Rep) and Constructive Solid Geometry (CSG) [8]. Shape analysis and signature extraction are tightly linked with the representation of 3D models. Different representation methods define a 3D shape by different ways. For example, mesh and B-Rep define 3D objects by surface information, while CSG uses both structure and volume information to define 3D solids. Here we use the voxel representation [9] to represent 3D models. A voxel model is composed by a set of volume

elements (voxels) in a discrete 3D space. A voxel represents a cubic unit of volume, and it can be seen as the 3D counterpart of the 2D pixel that represents a unit of area.

The process of converting a geometry object in 3D continuous space into a set of voxels that best approximates the object is called voxelization. Computer graphics community has developed many algorithms for voxelizing mesh, B-rep and CSG models. Because mesh is the most common representation method for 3D objects, in this paper we only consider mesh models.

Intuitively, voxelizing a 3D mesh model is like filling up a closed hull with volume elements. Voxelization algorithm converts the mesh model into a voxel collection. The voxelization is a process of intersection test. According to the intersection of facets and voxel grid, each voxel is set to one of the value 0 or 1. Value 0 represents the grid cell does not intersect with mesh facets, and value 1 represents the grid cell intersects with mesh facets. In order to get more information, we propose a gray voxel algorithm. A gray voxel model is a voxel mode that the voxel takes the value in a range, not just 0 or 1. There are two common ways to define the value of a gray voxel:

- (1) the value represents the area of facets inside a grid cell.
- (2) the value represents the Discrete Gaussian Curvature of facets that intersect with the grid cell.

The Discrete Gauss Curvature of a vertex is computed by the following equation:

$$K(v) = (1/A(v))(2\pi - \sum \theta_i) \quad (1)$$

In this equation, $A(v)$ is the Voronoi area of a the vertex and $\sum \theta_i$ is the sum of the angles of the neighborhood triangles.

SIFT on Voxel Model

When we obtain the gray voxel model, we employ the 3D voxel SIFT to extract the scale-invariant shape feature of the voxel model. The 3D voxel SIFT is a variety of SIFT to work in 3D voxel space. The first step of 3D voxel SIFT is the computation of multi-scale DoG (Difference of C):

$$DOG(x, y, z, k) = I(x, y, z) * G(x, y, z, k\sigma) - I(x, y, z) * G(x, y, z, (k-1)\sigma) \quad (2)$$

In this equation, $*$ is convolution, and $I(x, y, z)$ is the function of the gray values of a voxel model. $G(x, y, z, k\sigma)$ is the Gaussian function. k is an integer in $[1,5]$ to indicate different levels of scale. σ is a constant that set to $\sqrt[3]{2}$. The extremums of DOG in voxel neighborhood space and in scale neighborhood space is the candidate points of scale-invariant shape feature.

To improve the stability of the shape features, the extremums should be filtered to remove the effects of edge response. The following equation is used to filter the extremums [10].

$$\frac{Trace^3(\mathbf{H})}{Det(\mathbf{H})} < \frac{(2r+1)^3}{r^2} \quad (3)$$

In this equation, \mathbf{H} is the Hessian matrix of DOG . r is a threshold usually being set to an integer in $[10,50]$. In the experiments, we set r to 30. $Trace$ is the matrix trace and Det is the determinant. Unless the value of DOG satisfy the equation, the point is chosen as a feature points. When all the feature points are obtained, we can construct scale-invariant shape feature vector according to the gradient direction of the neighborhood voxels of the feature points.

Local Shape Feature Vector and Dissimilarity Distance

The shape feature of the 3D models is defined by the set of scale-invariant shape feature vector set of each model. It is not appropriate to compare each pairs of these vectors of two 3D models, for comparing a large number of high dimensional vectors is computational consuming.

We employ the BoF (Bag of Features) to reduce the number of vectors to be compared. BoF is a common strategy for the comparison between two data sets. It comes from the Bag of Words in the field of text retrieval and natural language processing. The workflow of BoF is as follows.

Step 1: A large number of scale-invariant shape features are extracted from the 3D model of the training database, and then the shape features are clustered.

Step 2: Every class formed by cluster algorithm can be regarded as a typical shape feature, so that the class center of each class is constructed as a shape code. All the shape code of the centers of each class forms the shape codebook. A original scale-invariant feature vector of a 3D model can be represented by the latest shape code in codebook.

Step 3: When the shape codebook is ready, the shape code histogram of a 3D model can be constructed by the following steps. 1) Finding the shape code of every scale-invariant shape feature vector of the 3D model. 2) Forming a shape code histogram by counting the number of shape codes corresponding to each scale-invariant shape feature vector. The histogram represents the frequencies of shape codes appears on the 3D model. Figure 1 illustrates the process of generating a shape code histogram of a 3D model.

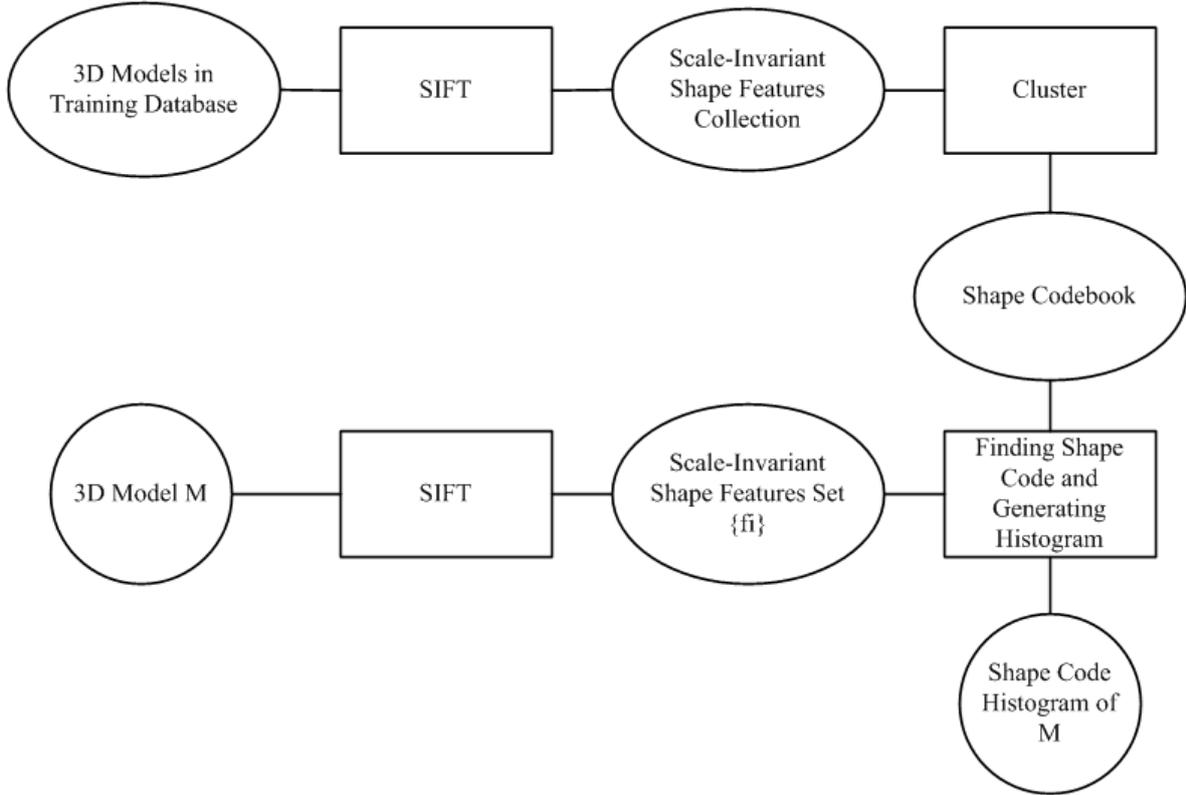


Fig. 1 Process of generating shape code histogram

The part-in-whole shape match is not symmetrical. For example, an human arm can be matched with a human body but human body should not be matched with an arm. We employ the Kullback-Leibler distance [11] to assess the shape dissimilarity between 3D models. Kullback-Leibler distance is a kind of asymmetric distance. Suppose $P=\{p_i\}$ and $Q=\{q_i\}$ are two vectors representing two histograms. KL distance is computed by the following equation:

$$D_{kl}(P \parallel Q) = \sum p_i \log \frac{p_i}{q_i} \quad (4)$$

The computation of KL distance involves log function, which is computational consuming when there are a large number of high dimension vectors of histograms to be compared. The dimensions of a shape code histogram is namely the number of shape code in codebook. The dimensions may be between hundreds and thousands according to different 3D model database. We can design a fast log function computation algorithm by a constructing a lookup table.

Experiments

We employ the “precision and recall” curve to measure the performance of our prototype system. “Precision” measures the ability of the system to retrieve only models that are relevant and “recall”

measures the ability of the system to retrieve all models that are relevant. There is a trade-off between precision and recall. In precision-recall diagrams, a perfect retrieval result would produce a horizontal line at the top of the plot. Otherwise, a curve closer to the upper-right corner represents better performance.

We have tested the retrieval performance of scale-invariant shape feature with different gray voxelization process (Curvature vs. Area). We compare the retrieval performance of scale-invariant shape feature with two other shape feature (Ohbuchi [12] and Cornea [13]). Figure 2 illustrate the Precision-Recall curves of these methods. In Figure 2, the Curvature stands for the scale-invariant shape feature with gray voxelization using curvature, and Area stands for the scale-invariant shape feature with gray voxelization using area.

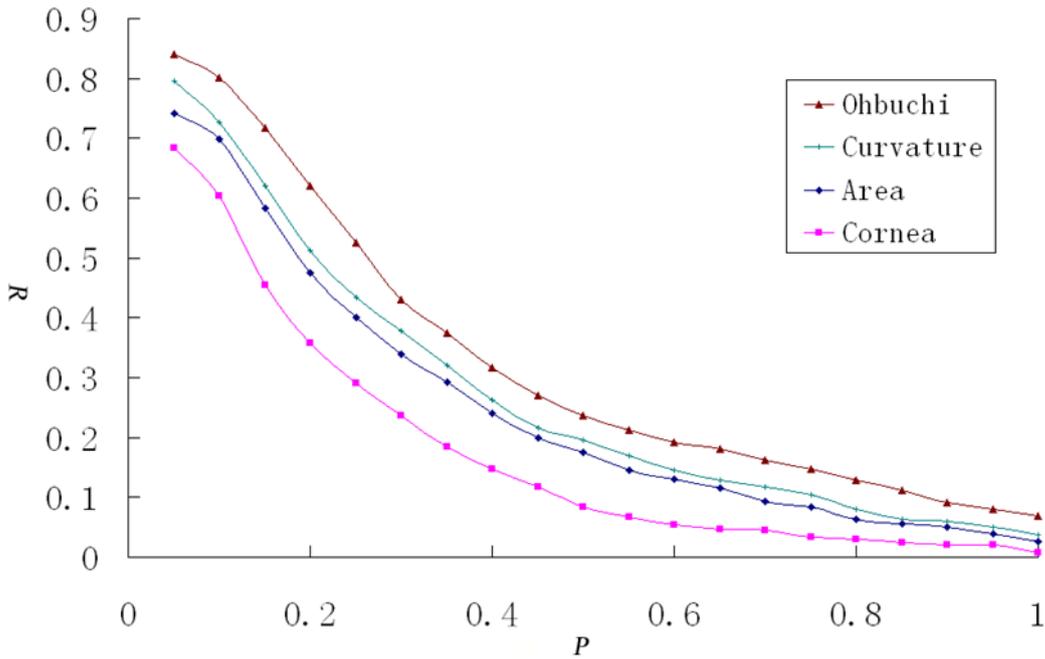


Fig. 2 Precision-Recall curves

Table 1 lists the computation time of feature extraction. Due to the Ohbuchi’s method needs to render and compute lots of 2D projections of 3D models, it takes a much longer time for feature extraction. Cornea’s method is the secondary time consuming, for it needs to generate the skeleton of a 3D model.

Table 1 Computation time of feature extraction

	Area	Curvature	Ohbuchi	Cornea
Computation time	3.12(s)	4.53(s)	8.64(s)	5.71(s)

Conclusions

We introduce a local shape feature for 3D model retrieval in this paper. This local shape feature is extracted on gray voxel models by a SIFT-like algorithm. Shape feature vectors are constructed by a BoF method and compared by a distance function that supports the part-in-whole matching. Experiments demonstrate that the proposed method achieves a satisfactory retrieval performance.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61202200) and the Scientific Research Fund of Zhejiang Provincial Education Department (No.Y201018008).

References

- [1] M. Ankerst, G. Kastentmüller, H.-P. Kriegel, and T. Seidl. Nearest Neighbor Classification in 3D Protein Databases. In Proc. ISMB, 1999.
- [2] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D Models with Shape Distributions. Shape Modeling International, pages 154–166, May 2001.
- [3] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors. In Symposium on Geometry Processing, June 2003.
- [4] Hilaga, M., Shinagawa Y., Kohmura, Taku, and Kunii, T. L., 2001, Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. In ACM SIGGRAPH, Los Angeles, CA, August.
- [5] H. Sundar, D. Silver, N. Gagvani, and S. Dickenson. Skeleton Based Shape Matching and Retrieval. In SMI 2003, pages 130–139, 2003.
- [6] Cicirello, V. A., and Regli, W. C., 2001, Machining Feature-based Comparisons of Mechanical Parts. In International Conference on Shape Modeling and Applications, Genova, Italy, May.
- [7] Srinivas, G., Fasse, D. E., and Marefat, M. M., 1998, Retrieval of Similarly Shaped Parts from a CAD Database. In IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, October.
- [8] Hesham Anan, Digital Library Services for 3D Models, Phd dissertation, Old Dominion University, 2004
- [9] Kaufman, D. Cohen, and R. Yagel, “Volume Graphics,” IEEE Computer. Vol. 26, No. 7, July 1993, pp. 51-64.
- [10] Flitton GT, Breckon TP, Megherbi N. Object Recognition Using 3D SIFT in Complex CT Volumes[C]//In Proceedings of the British Machine Vision Conference. Aberistwyth, UK: BMVA Press, 2010: 1-12.
- [11] Kullback S, Leibler RA. On Information and Sufficiency[J]. Annals of Mathematical Statistics, 1951, 22(1):79–86.
- [12] Ohbuchi R, Osada K, Furuya T, et al. Salient Local Visual Features for Shape-Based 3D Model Retrieval[C]// In IEEE International Conference on Shape Modeling and Applications. New York, USA: IEEE Computer Society, 2008: 93-102.
- [13] Cornea ND, Demirci MF, Silver D, et al . 3D Object Retrieval Using Many-to-Many Matching of Curve Skeletons[C]//In IEEE International Conference on Shape Modeling and Applications. Cambridge, MA, USA: IEEE Computer Society, 2005: 368-373.