Vis Comput (2012) 28:75–86
DOI 10.1007/s00371-011-0605-8

ORIGINAL ARTICLE

3D model retrieval based on color + geometry signatures

Yong-Jin Liu · Yi-Fu Zheng · Lu Lv · Yu-Ming Xuan · Xiao-Lan Fu

Published online: 2 June 2011

Abstract Color plays a significant role in the recognition of 3D objects and scenes from the perspective of cognitive psychology. In this paper, we propose a new 3D model retrieval method, focusing on not only the geometric features but also the color features of 3D mesh models. Firstly, we propose a new sampling method that samples the models in the regions of either geometry-high-variation or color-high-variation. After collecting geometry + color sensitive sampling points, we cluster them into several classes by using a modified ISODATA algorithm. Then we calculate the feature histogram of each model in the database using these clustered sampling points. For model retrieval, we compare the histogram of an input model to the stored histograms in the database to find out the most similar models. To evaluate the retrieval method based on the new color + geometry signatures, we use the precision/recall performance metric to compare our method with several classical methods. Experiment results show that color information does help improve the accuracy of 3D model retrieval, which is consistent with the postulate in psychophysics that color should strongly influence the recognition of objects.

Keywords 3D model retrieval · Color features · Shape signature

Y.-J. Liu (✉) · Y.-F. Zheng · L. Lv
Tsinghua National Lab for Information Science & Technology, Department of Computer Science and Technology, Tsinghua University, Beijing, P.R. China
e-mail: liuyongjin@tsinghua.edu.cn

Y.-M. Xuan · X.-L. Fu
State Key Lab of Brain and Cognitive Science, Institute of Psychology, Chinese Academy of Sciences, Beijing, P.R. China

1 Introduction

In computer vision and computer graphics, complex 3D objects are usually modeled by dense triangle meshes. With the rapid development of the internet and the multimedia technology, these 3D digital models have been widely used in our daily life. As the 3D movies and 3D games becoming flourishing in recent years, the featured 3D film, *Avatar,* for an example, more attention has been paid to 3D models in several computer science areas such as computer vision, computer graphics, and geographic information systems. With the large amounts of 3D model data on the internet, the need for obtaining 3D models which satisfy different requirements quickly and accurately has prompted a new trend of research on 3D model retrieval.

In most traditional 3D model retrieval methods, only geometric information is considered. Figure 1 shows two 3D models whose shapes are almost the same while their colors present different information. In this case, obviously, we are not able to distinguish them by just considering the geometric features. Neurophysiological research has revealed that the visual stimulus to the human vision system is multidimensional and the separated perceptual dimensions include brightness, color, movement. and depth [1]. So, it is significant to carry out a new method that considers not only the geometric features, but also color features in retrieving 3D models.

Since the information in 3D models is very rich, many 3D model retrieval methods have been proposed from different perspectives. We classify these retrieval methods in the following three categories from the perspective of users' input manner: (1) Keywords: the advantage of this class is that the input is concise and the results could be good if the text annotations of 3D models are sufficiently large. However, the database of 3D models is hard to establish since it demands

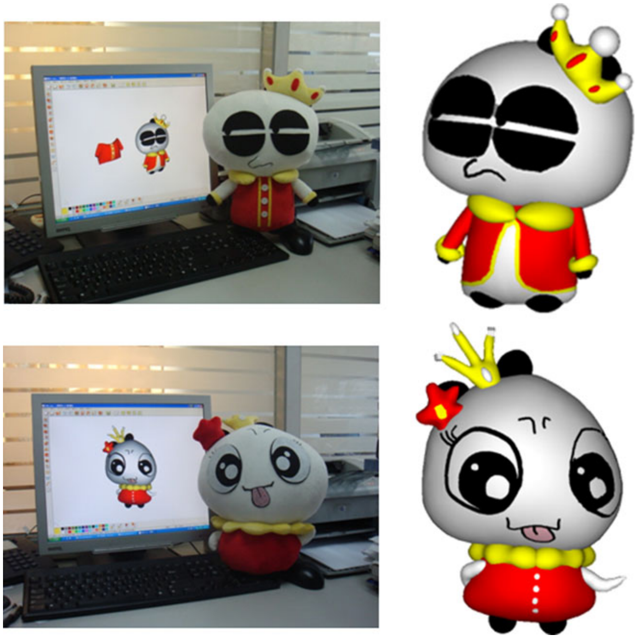


Fig. 1 Two 3D color models of *King* and *Queen*: these two physical models have similar shape but their colors offer different information

each model to be annotated with a lot of keywords, depending on many aspects across different knowledge domains. (2) 2D Images: the inputs of these methods are also simple and contain rich information about the original 3D models. However, as they lose the depth information, the retrieval accuracy is usually not high. (3) 3D models: the information of the original 3D models has been completely reserved in the input. So, the accuracy of this class is the highest in all the three classes.

In this paper, we focus on the third class of 3D model retrieval methods, i.e., the input of our method is a 3D model. In addition to utilizing the geometric information of the input 3D models, we will also consider the color information. Object retrieval using both geometry and color information has been studied in image process [2], neural computing [3], and 3D object recognition from color images [4]. In this paper, we study how to combine color and geometry-invariant features for 3D model retrieval. The color 3D models studied in this work include not only the synthesized computer models but also the real physical object models as shown in Fig. 1.

The rest of the paper is organized as follows. Section 2 is about related work. The method for feature extraction using both color and geometric information is introduced in Sect. 3. Section 4 presents the main algorithm for 3D model retrieval based on feature clustering and the associated feature histogram. The experimental results of our method and the comparison to several classic methods are presented in Sect. 5. Finally, we offer our conclusions in Sect. 6.

2 Related work

To solve the 3D model retrieval problem, most methods use geometry-based signatures of 3D models. Ankerst et al. [5] proposed a method, known as D2 algorithm, based on a 3D shape histogram for similarity search and classification in spatial database. The D2 algorithm was later refined in [6, 7]. Using the points randomly sampled on models, the D2 method calculates the Euclidean distance between every pair of two different points and establishes a shape histogram of a 3D model using these Euclidean distances. Then the similarity between two models is obtained by calculating the L_2 distance of their histograms. This method is simple to implement and its retrieval accuracy is also comparable high. Vranic and Saupe [8] proposed a 3D shape signature based on 3D Fourier transform. This method utilizes the frequency domain instead of the spatial domain. Another classical method is based on extended Gaussian images (EGI) [9, 10]. The principle of this method is to cast the area distribution of a model to the surface of a sphere whose central point is the center of the original model. The EGI method retrieves models well in convex polyhedrons, but it is not good for concave polyhedrons.

In addition to the global matching, Johnson and Hebert [11, 12] proposed the spin image algorithm that requires only partial surfaces as input. Since its computational complexity is relatively high, the spin image algorithm requires preprocessing the models in the database in order that all the models are in the same size. All the above methods use geometry-based signatures of the models. Another class of methods is based on topology of the models. One typical work in this class is proposed by Hilaga and his colleagues, which is called the Multiresolution Reeb Graphs (MRG) algorithm [13]. In this method, the approximate geodesic distance between different points on the 3D models is used to establish the multi-resolution Reeb graphs. The experimental results of this method are good but the process of establishing the topological structures is still complicated.

Retrieving 3D models from 2D images is also a hot topic in 3D model retrieval research. The main idea of this class of methods is to get the information of the original 3D model by taking photos of the model in different views. Although the depth information is lost, the retrieval process is simplified by just comparing and matching 2D images. The typical work in this class includes the model-view matching [6] and the light field signatures [14]. The work in [6] takes photos of the models in their database from 13 different viewpoints and uses them as the representative views of each model. The main idea of the method in [14] is to represent 3D models with a group of 2D projected images. By calculating the binary images (in black and white) of orthogonal projections of different 3D models, the models are matched by finding the 2D shape similarity among the projected images.

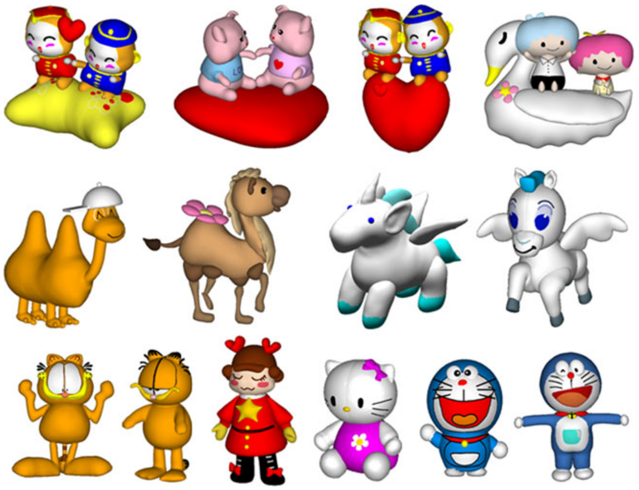


Fig. 2 Some 3D color models used in entertainment industry. The models in each row have similar geometric shape. In the *second row*, the first two models and the last two models have similar color distributions, respectively. In the *last row*, the models can only be successfully classified using color information

In spite of the good results accomplished by the methods mentioned above, none of them has considered the effect of the color information in 3D models. However, more and more 3D color models have become ubiquitous in different applications from virtual reality to entertainment industry (see Fig. 2). Meanwhile, the converging behavioral, neurophysiological, and neuropsychological evidence demonstrates that color has played a significant role in both low-level and high-level vision [15]. So, it is necessary for a 3D model retrieval method taking the color information of the models into account. A pioneering work was proposed in [4] that combines color and geometric features for 3D object recognition from 2D color images. In this paper, we present a new 3D model retrieval method using a 3D color model as input. Our experimental results demonstrate that the color features in 3D models help improve both the efficiency and the accuracy of retrieval when compared to the traditional geometry-based 3D model retrieval methods.

3 Feature point extraction

In vision research, the process of human vision system is separated into early vision and high-level vision [16, 17]. The early vision process concerns extraction and grouping of certain physical properties from visual input. In this section, we present a method that extracts low-level features on 3D color models. In the next section, we organize these low-level features into a feature histogram that serves as a shape signature to recognize and classify the objects.

Psychological model of the early vision process has postulated that the low-level features such as color and depth information are extracted from separated visual channels [18]. Given a 3D color model, two kinds of feature points are extracted: one concerns geometry and the other concerns color information. We extract geometric features based on curvature information since it is invariant in Euclidean and similarity transformation. For color feature extraction, we use the CIE-LAB color space [19] that is designed to approximate human vision and is more perceptually uniform than RGB and HSV color spaces.

3.1 Geometric feature extraction

The 3D model is represented by a triangle mesh, consisting of a vertex list and a triangle list. We randomly sample the model's surface by a set of points as follows. An array A is generated with the number of triangles in the model: $A[i]$ corresponds to the triangle t_i. The value stored in $A[i]$ is the sum of triangle areas accumulated so far, i.e., $A[i] = \sum_{j=1}^{i} \Delta t_j$, where Δt_j is the area of triangle t_j. A random number generator is used to sample between 0 and $A[n]$, where n is the number of triangles in the mesh model. For a generated random number x, the array index k is found which satisfies $A[k-1] < x \leq A[k]$ and a sample point is generated in the triangle t_k. The larger the triangle area Δt_k is, the more chances a sample point falls into t_k since $\Delta t_k = A[k] - A[k-1]$.

We use the Taubin's method [20] to compute the discrete curvatures at each vertex in the mesh. Taubin's method computes the numerical integral of the directional curvature at a vertex first and then decomposes the matrix obtained by the integral to get the mean and Gaussian curvature at that vertex. Since the Gaussian curvature vanishes everywhere in a developable surface [21] (i.e., plane, cylinder, cone, tangent surface, or a combination of them), we use the integral of the absolute Gaussian curvature over the surface as a measure of surface smoothness and use it to determine the number of sample points for each 3D model. We normalize a 3D model such that its total area is 100. For each normalized model M, the number of points we sampled on M is determined by

$$\max\left\{2000\left(\frac{G-a}{b}\right)^{c}, 2000\right\}$$

where G is the integral of the absolute Gaussian curvature on M, a, b, c are constants.[1]

Natural scenes usually contains millions of bits of information and human vision system has a remarkable ability to interpret complex scene in real time by selecting a subset of information at an early stage [22]. To select sample points representing visually salient regions [23] in the model, we use the scale-space theory [24] to generate a three-level curvature map for each model. Each level $C^i = C^{i-1} * L(\sigma)$ is

[1] We found that $a = 450$, $b = 2$, $c = 0.1$ works well in our experiments.

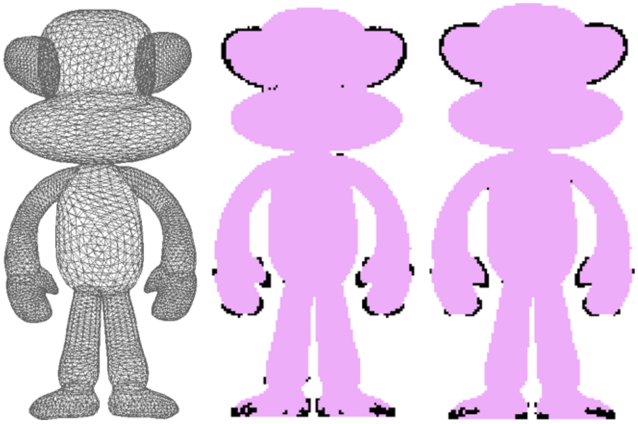


Fig. 3 Geometric feature selection using random samples. *Left*: the geometry of a 3D color model. *Middle*: feature points selected by mean curvature. *Right*: feature points selected by Gaussian curvature

smoothed by a Laplace operator [25] of the previous level. The difference-of-Laplacian space $D^i = C^i - C^{i-1}$ offers us a metric of saliency.

We extract the geometric feature points with both mean and Gaussian curvature by selecting the top 10% points with the highest values $D = D^1 \oplus D^2$ in each model. One of sampling results using mean curvature is shown in Fig. 3 (middle) and the result of the same model using Gaussian curvature is shown in Fig. 3 (right). Figure 3 (left) shows the geometry of the original 3D model. In order to display the feature points clearly, we hide the brightness and color effect of the model in Fig. 3.

After experiments with the models in databases [35, 40], we find that it is better to use mean curvature as the metric to extract geometric feature points in 3D models. Our experiments are consistent with the discovery in [26] that mean curvature is a good metric to indicate model saliency in low-level human visual attention. We also find that choosing points using mean curvature makes the feature points concentrated in some high-variational area. So, in our method, we extract the geometric feature points in the following way. First, we calculate the mean curvature of each point and then sort all the points by the value of difference-of-Laplacian of mean curvature from high to low. Then we traverse the sorted points in order and choose a point as the feature point if none of its neighboring points has been chosen until the number of the chosen points adds up to 10% of the number of total sample points in the model. One example of our results is presented in Fig. 4.

3.2 Color feature extraction

Several widely used color spaces such as RGB and HSV are device-dependent color models, and are unrelated to human perception. We use CIE-LAB color space [19] for color feature extraction, where L^* represents the lightness of the color, a^* represents the position between red/magenta and green, b^* represents the position between yellow and blue. Given an RGB color value, first it is converted into CIE XYZ system:

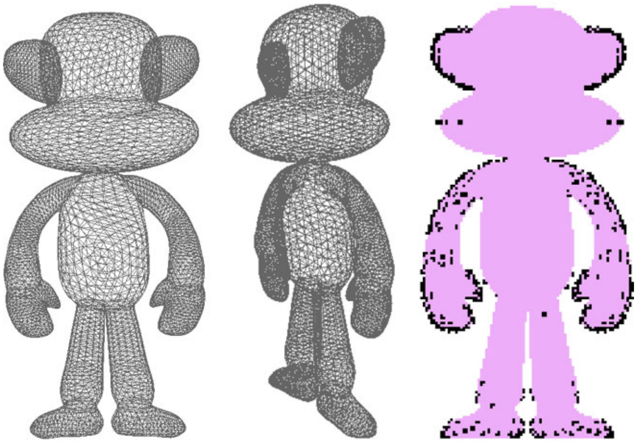


Fig. 4 Geometric feature selection using mean curvature and neighborhood constraints. The color rendering of the same model is shown in Fig. 5

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Then the CIE-LAB coordinates are obtained by

$$L^* = 116 f\left(\frac{Y}{Y_n}\right) - 16$$

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right]$$

$$b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right]$$

where

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > (\frac{6}{29})^3 \\ \frac{1}{3}(\frac{29}{6})^2 + \frac{4}{29} & \text{otherwise} \end{cases}$$

and X_n, Y_n, Z_n are the CIE XYZ tristimulus values of the reference white point [19]. The nonlinear relations for L^*, a^*, b^* mimic the nonlinear response of the retinal, and uniform changes of coordinates in CIE-LAB space correspond to uniform changes in perceived color [27, 28].

It was widely accepted that human vision system uses three different photo-sensitive substances in the cones, i.e., in the initial step in the system, the color is encoded in three separated visible spectrums for red, green, and blue (RGB). After light reaches the cones, nerve impulses are generated, which carry the color information to the brain. The nerve fibers along which the signals transform, are in the regions where rods and cones are interconnected. It seems unlikely that the color information is transformed in RGB form and the probability of transformation in opponent-color form is enhanced [29]. Therefore, defining measures in CIE-LAB color space has physiological support.

In our method, we measure the CIE-LAB color similarity between different 3D models based on the distributions of color difference in each model. Our implementation stores the color model data in the wavefront obj file format, which contains the lists of vertex coordinates, texture coordinates, normals and faces. A face in obj format is defined as

$$f \quad v1/vt1/vn1 \quad v2/vt2/vn2 \quad v3/vt3/vn3$$

where $(v1, v2, v3)$ represents the IDs of three vertices, $(vt1, vt2, vt3)$ represents the IDs of three texture coordinates, and $(vn1, vn2, vn3)$ represents the IDs of three normals. For every vertex in a face, there is a texture coordinate assigned to it. A texture coordinate represents a position in the texture image, from which a color value could be retrieved.

We partition the CIE-LAB color space into cells and all the points in each cell are regarded as having the same color. For each point p in the color space, we assign a color code to p that represents to which cell the point falls in. The color code is generated using the following rule:

$$code(L^*) = \begin{cases} 0 & \text{if } L^* \in [0, 15) \\ 16 & \text{if } L^* \in [15, 30) \\ 32 & \text{if } L^* \in [30, 45) \\ 48 & \text{if } L^* \in [45, 60) \\ 64 & \text{if } L^* \in [60, 75) \\ 80 & \text{otherwise} \end{cases}$$

$$code(a^*) = \begin{cases} 0 & \text{if } a^* \in [-100, -50) \\ 4 & \text{if } a^* \in [-50, 0) \\ 8 & \text{if } a^* \in [0, 50) \\ 12 & \text{otherwise} \end{cases}$$

$$code(b^*) = \begin{cases} 0 & \text{if } b^* \in [-100, -50) \\ 1 & \text{if } b^* \in [-50, 0) \\ 2 & \text{if } b^* \in [0, 50) \\ 3 & \text{otherwise} \end{cases}$$

The code number of point p is defined as $code(p) = code(L^*) + code(a^*) + code(b^*)$. Since this coding scheme is a combination of $16 \times (0 \sim 5) + 4 \times (0 \sim 3) + (0 \sim 3)$, two different cells cannot have the same coding, and thus the color code is unique for each cell.

We randomly sample the model's surface using a set of points. Note that different color values can be assigned to the same vertex. If one sample point is a vertex and less than 80% colors assigned to it have the same color code, we set this point as a color feature point. For a sample point that is not a vertex, we compare the color code of this point with the color codes of its neighboring points. If less than 80% colors have the same color code, this point is triggered as a color feature point.

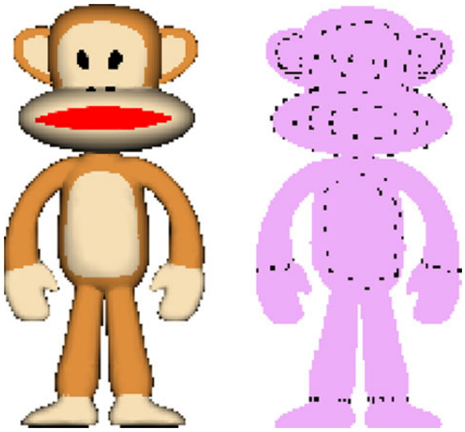


Fig. 5 Color feature selection. The wireframe rendering of the same model is shown in Fig. 4

To avoid too many points being sampled in a small area with a strong texture pattern, for each sample point, we first check whether there are any color feature points in its neighboring points. If at least one of its neighboring points has been chosen as a color feature point, the point cannot be chosen as a feature point. Figure 5 shows an example of color feature point selection.

4 Feature histogram as a shape signature

Given both geometric and color feature points, we establish a feature histogram serving as shape signature to distinct the model from others. We first use a modified ISODATA algorithm [30] to cluster the feature points into different clusters. Then we propose two signatures based on the clustered feature points. One signature is akin to the D2 algorithm [5, 7], which is based on the Euclidean distance, and the other is based on an angle invariance. Both signatures are invariant to similarity transformations.

The reason of implementing a D2-style signature is that in our previous work [31], we have compared five representative geometric signatures for 3D model retrieval: D2 [6, 7], G2 [32] (geodesic-bending-invariance), EGI [9, 10] (extended Gaussian images), GMT [33] (geometric moment invariants), and SPIN [11, 12] (spin images). Our results [31, 34] show that using McGill 3D shape benchmark [35], D2 shape signature has better performances than other methods.

In addition to D2 shape signature, we also propose an angle-based geometric invariance. This invariance is based on the postulate in [36] that interobject spatial relations and allocentric reference directions play an important role in learning process and spatial memory of human. Our experimental results presented in Sect. 5 show that using precision/recall plot [37] as an analytical tool, ClusterAngle + Color signature has better performance than the ClusterD2 + Color signature.

We use the ISODATA algorithm [30] to partition the feature points into clusters with the following properties: (1) the number of points in each cluster is almost the same, (2) the position variance of points in each cluster is almost the same, (3) the points inside a cluster are close to each other, while the points lying in different clusters are far from each other. Traditional ISODATA algorithm needs to assign the number of the clusters at the beginning. We use a modified version of ISODATA algorithm with the following changes: (1) instead of inputting a fix number of clusters, the algorithm automatically finds an optimal number by iteration, (2) we give equal chance to split one cluster into two or merge two clusters into one. The details of this modified ISODATA algorithm is presented in the Appendix.

After partitioning all the feature points (including both geometric and color feature points) into clusters, the next step is to calculate a histogram of the 3D model using those clustered feature points. We compute two types of histograms: ClusterD2 + Color and ClusterAngle + Color. Suppose that in the model there are c clusters and F_i is the set of feature points in the ith cluster with the number of points in F_i being n_i.

The following steps compute the histogram of ClusterD2 + Color:

1. For each point $f_{ip} \in F_i$ and $f_{jq} \in F_j$, where $i \neq j, i, j = 1, 2, \dots, c,\ p = 1, 2, \dots, n_i,\ q = 1, 2, \dots, n_j$, compute the Euclidean distance $d_{i_p j_q} = \| f_{ip} - f_{jq} \|$ and store all the distances in an array D.
2. Find the maximum value $d_{\max}$ and the minimum value $d_{\min}$ in D.
3. Normalize all the values in D by $d'_i = \frac{d_i - d_{\min}}{d_{\max} - d_{\min}}$.
4. Convert the normalized array D into a histogram with 20 bins.

The following steps compute the histogram of ClusterAngle + Color.

1. For each point $f_{ip} \in F_i$, $f_{jq} \in F_j$, and $f_{kr} \in F_k$, where $i \neq j \neq k,\ i, j, k = 1, 2, \dots, c,\ p = 1, 2, \dots, n_i,\ q = 1, 2, \dots, n_j, r = 1, 2, \dots, n_k$, compute the angle spanned by vectors $\overline{f_{ip} f_{jq}}$ and $\overline{f_{kr} f_{jq}}$, i.e.,
$$\angle f_{ip} f_{jq} f_{kr} = \arccos \frac{\overline{f_{ip} f_{jq}} \cdot \overline{f_{kr} f_{jq}}}{\| \overline{f_{ip} f_{jq}} \| \| \overline{f_{kr} f_{jq}} \|},$$
and store all the angles in an array A.
2. Find the maximum value $a_{\max}$ and the minimum value $a_{\min}$ in A.
3. Normalize all the values in A by $a'_i = \frac{a_i - d_{\min}}{a_{\max} - a_{\min}}$.
4. Convert the normalized array A into a histogram with 20 bins.

In the above histogram definitions for both ClusterD2 + Color and ClusterAngle + Color, we use bin number being 20. Representing a histogram with a finite number of bins will inevitably produce some quantization error. In Sect. 5, we present the model retrieval results of experiments using bin number 10, 15, 20, 25. The experimental results show that bin number 10 has the lowest retrieval accuracy and there is no significant difference among bin number 15, 20, 25. Thus, we choose the bin number 20 as a good trade-off between the retrieval accuracy and the concise representation of a shape histogram.

Figure 6 shows three examples of ClusterD2 + Color and ClusterAngle + Color histograms, from which it is observed that (1) for the same object *Monkey* with two different poses, their ClusterD2 + Color and ClusterAngle + Color histograms are similar; (2) for a different object *Bear*, the ClusterD2 + Color and ClusterAngle + Color histogram has a major discrepancy with the histograms of Monkey. In Sect. 5, experimental results are presented, which show that ClusterAngle + Color histogram has a better retrieval performance than ClusterD2 + Color histogram.

Given a 3D model, we regard its shape histogram as an independent and identically-distributed sample of a random variable, and explore its structure by kernel density approximation [38]:

$$\widehat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

where $n = 20$ and kernel K is a standard Gaussian function with mean zero and variance 1. We use a B-spline curve C of degree 3 to interpolate the histogram and the bandwidth h is estimated by [39]:

$$h = \left(\frac{R(K)}{\mu_2^2(K) R(C'')}\right)^{\frac{1}{5}} n^{-\frac{1}{5}}$$

where $R(K) = \int K(t)^2\,dt$ and $\mu_2^2(K) = \int t^2 K(t)\,dt$. Given two models $\mathcal{M}_1, \mathcal{M}_2$ with distributions $\widehat{f}_h(\mathcal{M}_1)$, $\widehat{f}_h(\mathcal{M}_2)$, Jensen–Shannon divergence D_{JS} is used to define a probability dissimilarity of two models:

$$D_{JS}\big(\widehat{f}_h(\mathcal{M}_1), \widehat{f}_h(\mathcal{M}_2)\big) = \frac{1}{2} D_{KL}\big(\widehat{f}_h(\mathcal{M}_1) || \widehat{f}_h(\mathcal{N})\big) + \frac{1}{2} D_{KL}\big(\widehat{f}_h(\mathcal{M}_2) || \widehat{f}_h(\mathcal{N})\big)$$

where

$$\mathcal{N} = \frac{1}{2}(\mathcal{M}_1 + \mathcal{M}_2), \quad D_{KL}(\widehat{p} \| \widehat{q}) = -\int \widehat{p}(x) \log \frac{\widehat{p}(x)}{\widehat{q}(x)}\,dx$$

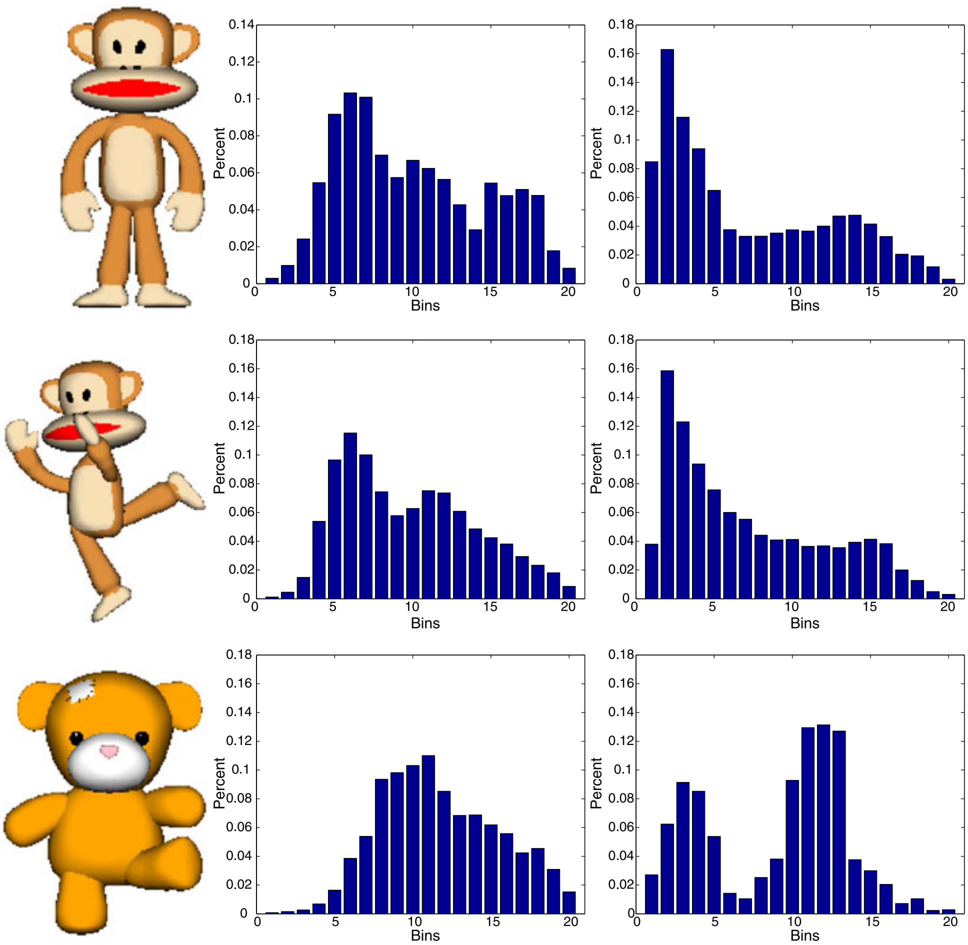


Fig. 6 Histograms of three color 3D models. *Left column*: the color 3D model. *Middle column*: the ClusterD2 + Color histogram. *Right column*: the ClusterAngle + Color histogram

5 Experimental results

We use two publicly available benchmarks of 3D models, the McGill 3D shape benchmark [35] and the engineering shape benchmark [40], to test the proposed shape signatures, as well as comparisons to several well known shape signatures. We also develop a database of 210 color 3D models. In each benchmark, the 3D models are organized into classes, according to their functions and forms. We use the precision/recall plot [37] to measure the retrieval quality of different shape signatures. The precision/recall plot in our evaluation is computed as follows:

1. For a model in the database, we match it to all the models in the database (including itself) and the results are ranked according to the similarity scores.
2. Suppose that a model $\mathcal{M}$ is in some class $C_{\mathcal{M}}$ with c members. For the ith retrieved relevant result from the same class ($i = 1, 2, \ldots, c$), the recall value is i/c.
3. Given a recall value i/c ($i = 1, 2, \ldots, c$), we find the ranking r of the ith model of this class $C_{\mathcal{M}}$ in the retrieved results. Then the precision is i/r.
4. For each model in the database, we compute the precision/recall plot of that model and the final output of plot is averaged over all the models' plots.

At the same recall value, the higher the precision value is, the better the retrieval method will be. For a whole precision/recall plot, the more area the whole plot enclosed with two coordinate axes, the better performance the underlying method has.

Accuracy on geometric signatures. 3D models can be classified into graphics data or engineering part data. The graphics 3D models usually have plenty of geometric details, while the 3D models of engineering parts have complex topological type, i.e., a high genus number. For testing on graphics 3D models, we compare the geometry-based signatures, ClusterD2 and ClusterAngle signatures (not including color), with three well-known signatures in graphics models, i.e., EGI [9, 10], SPIN [11, 12], and D2 [6, 7], using the McGill 3D shape benchmark [35]. The results shown in Fig. 7 reveal that (1) ClusterD2 performs slightly better than D2 and (2) ClusterAngle signature has the best performance among all the five signatures. We then compare the ClusterAngle signature using the engineering shape benchmark [40], with three well-known signatures in engineering

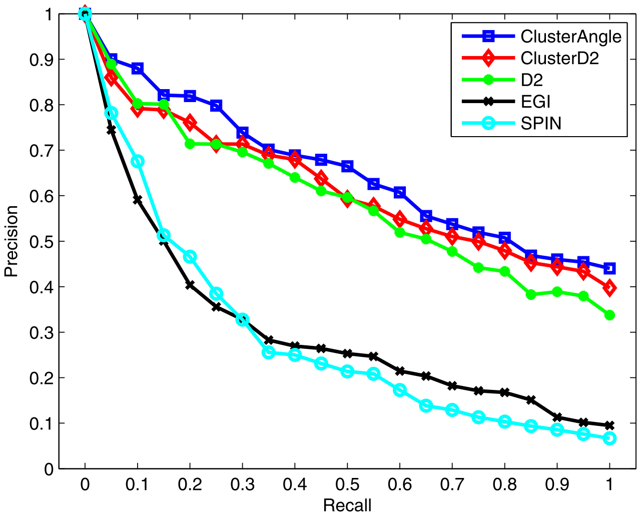


Fig. 7 Precision/recall plot of five geometric signatures: EGI [9, 10], SPIN [11, 12], D2 [6, 7], ClusterD2 and ClusterAngle signatures. Comparison is done using the McGill 3D shape benchmark [35]

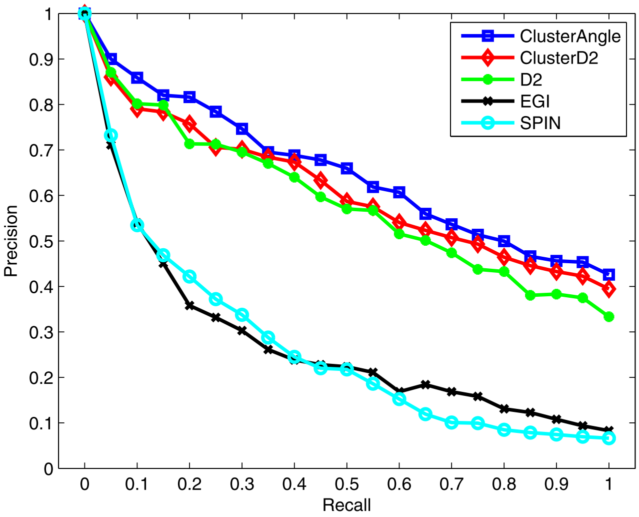


Fig. 9 Precision/recall plot of five geometric signatures against noise disturbance: EGI [9, 10], SPIN [11, 12], D2 [6, 7], ClusterD2, and ClusterAngle signatures

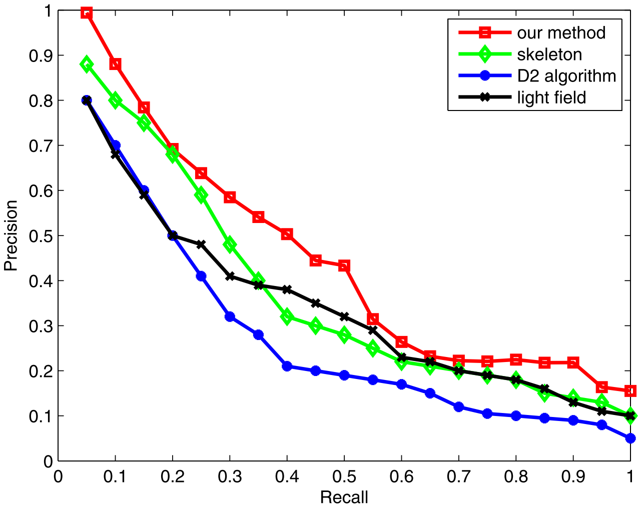


Fig. 8 Precision/recall plot of four geometric signatures: skeleton [41], D2 [6, 7], light field [14], and our method using ClusterAngle signatures. Comparison is done using the engineering shape benchmark [40]

Table 1 Classification of 3D color models in the database

Class name	Model num	Class name	Model num
Queen	22	QQ Tang	19
Monkey	19	Panda	22
Dancer	19	Pikachu	18
Cat	17	Doll	19
Garfield	16	Bingo	18
Bear	21		

part model retrieval, i.e., skeleton [41], D2 [6, 7], and light field [14]. The results are shown in Fig. 8, demonstrating the proposed ClusterAngle signature outperforms other signatures.

Robustness on geometric signatures. To test the noise sensitivity of the proposed signatures, we add noise to the McGill 3D shape benchmark by disturbing each vertex along its normal direction. The disturbance is randomly determined between the range of $(-\mathit{maxvalue}, \mathit{maxvalue})$ with zero mean, and *maxvalue* is chosen as 0.05 times the diagonal length of bounding box of each model. The comparison results of five geometry-based signatures against noise is shown in Fig. 9. From the results, it is observed that (1) D2 [6, 7], ClusterD2, and ClusterAngle signatures are robust to noise disturbance and (2) EGI [9, 10] and SPIN [11, 12] are more sensitive to disturbance by noises: this may be explained by that both EGI and SPIN need to utilize vertex normal information, which is very sensitive to noises.

Color + geometry signature improvement. To compare the ClusterD2 + Color and ClusterAngle + Color signatures, we build up a database of 210 3D color models. According to their semantics and geometric similarity, these models are classified into 11 classes (22 in Queen class, 19 in Monkey class, 19 in Dancer class, 17 in Cat class, 16 in Garfield class, 21 in Bear class, 19 in QQ Tang class, 22 in Panda class, 18 in Pikachu class, 19 in Doll class, and 18 in Bingo class) and Table 1 presents the classification. We test all the models in this database and compare the retrieval performance of two pairs of signatures: (*ClusterD2*, *ClusterD2 + Color*), (*ClusterAngle*, *ClusterAngle + Color*). The performance results are summarized in Figs. 10 and 11. These results show that (1) geometry + color signatures have better performance than using the geometric signatures only and (2) ClusterAngle signature has better performance than

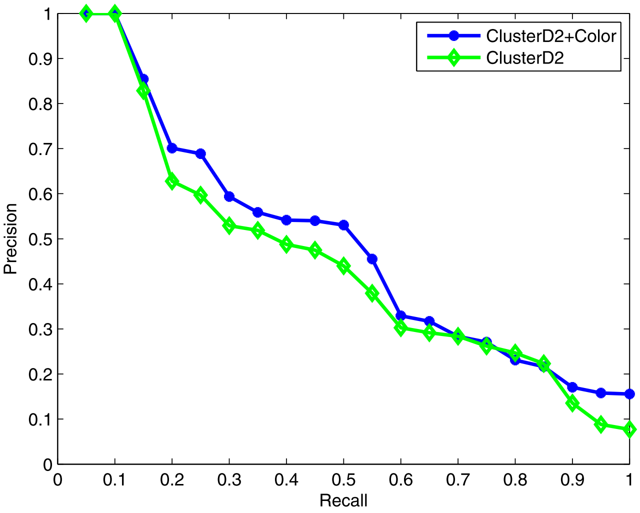


Fig. 10 Precision/recall plot of two signatures of ClusterD2 and ClusterD2 + Color, testing in the database of 3D color models summarized in Table 1

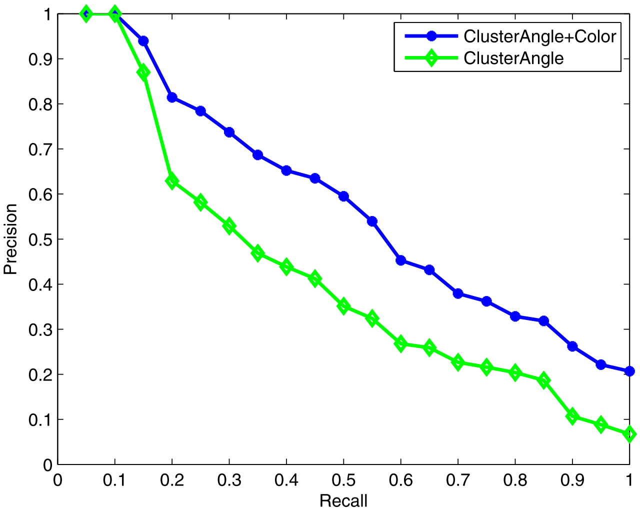


Fig. 11 Precision/recall plot of two signatures of ClusterAngle and ClusteAngle + Color, testing in the database of 3D color models summarized in Table 1

ClusterD2 signature. The tests were performed in a prototype retrieval system of 3D color models. Its interface is shown in Fig. 12 in which users input a 3D color model and the system feedbacks the ranked 3D color models in the database indexed by descending order of similarity values.

Optimal number of bins. In our method, we discretize the shape histogram using 20 bins (see Fig. 6). If we use less bins to represent the histogram, the shape signature will be shorter and the retrieval time can be reduced. However, the accuracy of histogram representation is decreased. If we use more bins, the histogram accuracy is improved but the retrieval time is increased. We test the bin numbers of 10, 15, 20, 25, using the 3D color model database with the ClusterAngle + Color signature. The results are shown in

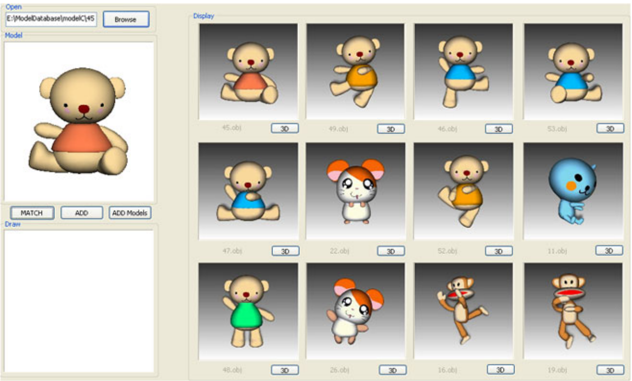

E:\ModelDatabase\modelC\45

Fig. 12 The interface of a prototype 3D color model retrieval system

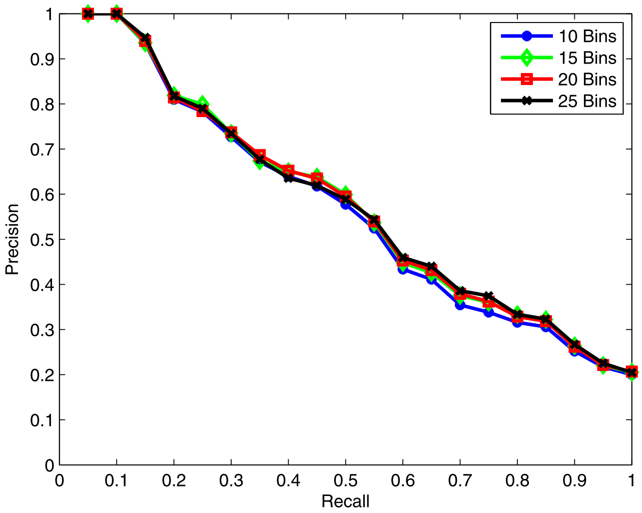


Fig. 13 The comparison of different bin numbers 10, 15, 20, and 25

Fig. 13, from which we observed that bin number 10 has the worst retrieval performance and there is no significant difference between bin numbers 15, 20, and 25. We thus choose bin number 20 as a good trade off between retrieval time and accuracy.

Limitation of the proposed methods. As revealed by the results in Figs. 7 and 9, ClusterAngle and ClusterD2 have better performance than the SPIN signature [11, 12]. However, SPIN signature has a well-recognized feature by which partial surface match can be done very well. It is interesting to ask whether or not ClusterAngle + Color and ClusterD2 + Color have also this nice property. We perform experiments as follows. For each color model, eight directions are randomly determined and for each direction a partial surface with color (in the form of range image) is generated in computer by simulating range scanning. Then we use the generated partial surfaces to search the database and the precision/recall plot is generated. The results are shown in Fig. 14, from which it is observed that ClusterAngle + Color and ClusterD2 + Color cannot do better for model re-

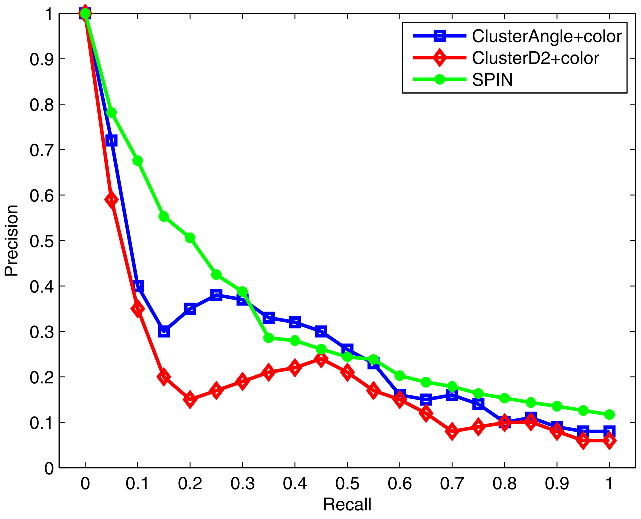


Fig. 14 Precision/recall plot of ClusterD2 + Color, ClusterAngle + Color and SPIN [11, 12] in the application of partial surface matching

trieval than SPIN in the case of partial surface matching. It is possibly because that we use Jensen-Shannon divergence as a global metric with fixed size that is not suitable for partial surface matching. In this scenario, the variable-size description of distributions, such as the earth mover's distance [42], may be more appropriate, and we put this exploration in the future work.

6 Conclusions

In this paper, we propose a 3D model retrieval method based on geometry + color signatures. By randomly sampling a model, we extract geometric feature points using difference-of-Laplacian of mean curvatures and extract color feature points in the CIE-LAB color space. These feature points are clustered by a modified ISODATA algorithm and histograms of ClusterD2 + Color and ClusterAngle + Color are generated as shape signatures. Finally the similarity of two models are computed by Jensen–Shannon divergence, which is a measure of statistic information. For retrieval performance evaluation, 210 3D color models are used in a database with 11 classes. We use precision/recall plots to evaluate different shape signatures. Experimental results show that color information does help improve the accuracy of 3D model retrieval, which is consistent with the postulate in psychophysics that color should strongly influence the recognition of objects [15].

Acknowledgements The authors thank the reviewers for their comments that help improve this paper. The authors also appreciate McGill Shape Analysis Group and Purdue PRECISE lab for making their data publicly available. This work was supported by the National Basic Research Program of China (2011CB302202), the Natural Science Foundation of China (60970099), and Tsinghua University Initiative Scientific Research Program (20101081863).

Appendix

The modified ISODATA algorithm used in Sect. 4 works as follows. Denote the set of N sample points as $\{p_1, p_2, \ldots, p_N\}$. Some necessary parameters are defined below:

- P_n: the minimum number of sample points in a cluster. We set this parameter being 10% of the total number of sample points.
- P_s: the parameter of standard deviation. If the deviation in a cluster is greater than this parameter, we split the cluster. In our method, we set this parameter according to the model resolution which is defined as the mean length of all the edges in a model.
- P_c: the parameter for merging two clusters. If the distance between two centers of two different clusters is less than this parameter, we merge the two clusters into one. We set this parameter also according to the model resolution.
- I: the number of maximum iterations. We set this parameter being 100.
- y: the input set of sample points as $\{p_1, p_2, \ldots, p_n\}$.
- *Flag*: represent whether the algorithm is convergent. We initialize this parameter as false.

The algorithmic steps are as follows:

S1 Initialize the cluster number c as 5 and randomly pick c points to be the centers m_i, $i = 1, 2, \ldots, n$ of the c clusters.

S2 Partition all the samples into different clusters according to the following rule

$$y \in D_j \text{ if } \|y - m_j\| < \|y - m_i\|,$$
$$i, j = 1, 2, \ldots, c, \; i \neq j$$

where D_j is the jth cluster and m_j is its center.

S3 For each cluster D_j, if the number of samples in it is less than P_n, we delete this cluster and let $c = c - 1$. The samples that were previously assigned to D_j are re-assigned to other clusters, according to step S2.

S4 Recalculate the center of each cluster by the equation

$$m_j = \frac{1}{N_j} \sum_{y \in D_j} y$$

where N_j is the number of samples in cluster D_j.

S5 Calculate the mean distance d_j of all the distances between each sample to the center in cluster D_j, $j = 1, 2, \ldots, c$:

$$d_j = \frac{1}{N_j} \sum_{y \in D_j} \|y - m_j\|$$

S6 Calculate the mean of all the mean distance d_j:

$$d = \frac{1}{N} \sum_{j=1}^{c} N_j d_j$$

S7 If *Flag* $= true$ or the number of maximum iteration is reached, the algorithm is terminated and clustering result is returned. Otherwise, let the current iteration number is x. If x is an even number, turn to step S8. If x is an odd number, turn to step S11.

S8 For each cluster j, we calculate its standard deviation Q_j:

$$Q_j = [q_{j1}, q_{j2}, \ldots, q_{jr}]^t$$

$$q_{ji} = \sqrt{\frac{1}{N_j} \sum_{y_k \in D_j} (y_{ki} - m_{ji})^2}$$

where r is the dimension of the samples, y_{ki} is the ith component of the kth sample, m_{ji} is the ith component of the center of the jth cluster, q_{ji} is the standard deviation of the ith component of the jth cluster.

S9 For each cluster, find out $q_{j\max}$, $j = 1, 2, \ldots, c$, which is the maximum in all the components in the standard deviation.

S10 For each $q_{j\max}$ which is greater than P_s, if $d_j > d$ and $N_j > 2(P_n + 1)$, we split D_j into two clusters, and the centers of these two new clusters are m_{ja} and m_{jb}, respectively. Meanwhile, we delete the original cluster m_j and let $c = c + 1$, where m_{ja} and m_{jb} are calculated as follows:

(a) A value k between 0 and 1 is given. In our experiment, we use $k = 0.5$.

(b) Set $v_j = k[0, \ldots, q_{j\max}, \ldots, 0]^t$.

(c) Set $m_{ja} = m_j + v_j$ and $m_{jb} = m_j - v_j$.

Set $I = I + 1$ and *Flag* $=$ *false*. Turn to step S2. If no cluster needs to be split and *Flag* $=$ *ture*, just turn to step S2. If no cluster needs to be split and *Flag* $=$ *false*, set *Flag* $=$ *ture* and move to next step S11.

S11 Calculate the distances among the centers of different clusters:

$$s_{ij} = \|m_i - m_j\|, \quad i = 1, 2, \ldots, c; \; j = i + 1, \ldots, c$$

S12 Compare each s_{ij} with P_c and sort the ones which are less than P_c in an ascending order.

S13 Beginning with the minimum s_{ij}, if D_i and D_j have not been merged in the current iteration, we merge the two clusters and the new center of the merged cluster is

$$m_0 = \frac{1}{N_i + N_j}(N_i \times m_i + N_j \times m_j)$$

Let $I = I + 1$, $c = c - 1$, *Flag* $=$ *false* and turn to step S2. If no clusters have been merged in this step S13 and *Flag* $=$ *true*, just turn to step S2. If no clusters have been merged in this step S13 and *Flag* $=$ *false*, set *Flag* $=$ *true* and turn to step S8.

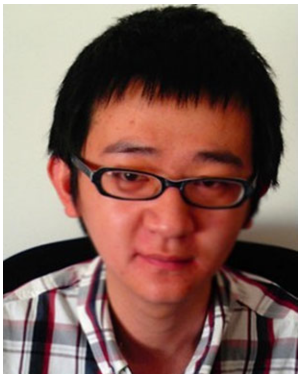
References

1. Livingstone, M.D., Hubel, D.H.: Psychophysical evidence for separate channels for the perception of form, color, movement, and depth. J. Neurosci. **7**, 3416–3468 (1987)
2. Gevers, T., Smeulders, A.: PicToSeek: combining color and shape invariant features for image retrieval. IEEE Trans. Image Process. **9**(1), 102–119 (2000)
3. Mel, B.W.: SEEMORE: combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition. Neural Comput. **9**, 777–804 (1997)
4. Slater, D., Healey, G.: Combining color and geometric information for the illumination invariant recognition of 3D objects. In: Proc. Fifth International Conference on Computer Vision (ICCV'95), pp. 563–568 (1995)
5. Ankerst, M., Kastenmulle, G., Kriegel, H., Seidl, T.: 3D shape histograms for similarity search and classification in spatial databases. In: Proceedings of the 6th International Symposium on Advances in Spatial Databases, pp. 207–228 (1999)
6. Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jacobs, D.: A search engine for 3D models. ACM Trans. Graph. **22**(1), 83–105 (2003)
7. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Matching 3D models with shape distributions. In: Shape Modeling International, pp. 154–166 (2001)
8. Vranic, D., Saupe, D.: 3D shape descriptor based on 3D Fourier transform. In: Proc. ECMCS 2001, pp. 271–274 (2001)
9. Horn, B.K.P.: Extended Gaussian images. In: Proceedings of the IEEE, pp. 1671–1686 (1984)
10. Vranic, D.: An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. In: IEEE International Conference on Image Processing (ICIP 2003), vol. 3, pp. 757–760 (2003)
11. Johnson, A.E., Hebert, M.: Surface matching for object recognition in complex 3-D scenes. Image Vis. Comput. **16**, 635–651 (1998)
12. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell. **21**, 433–449 (1999)
13. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L.: Topology matching for fully automatic similarity estimation of 3D shapes. In: ACM SIGGRAPH'01, pp. 203–212 (2001)
14. Chen, D., Tian, X., Shen, Y., Ouhyoung, M.: On visual similarity based 3d model retrieval. Comput. Graph. Forum (Eurographic'03) **22**(3), 223–232 (2003)
15. Tanaka, J., Weiskopf, D., Williams, P.: The role of color in high-level vision. Trends Cogn. Sci. **5**(5), 211–215 (2001)
16. Marr, D.: Vision: A Computational Investigation into the Human Representation and Process of Visual Information. Freeman, San Francisco (1982)
17. Ullman, S.: High-level Vision: Object Recognition and Visual Cognition. MIT Press, Cambridge (1996)
18. Cavanagh, P.: Reconstructing the third dimension: interactions between color, texture, motion, binocular disparity, and shape. Comput. Vis. Graph. Image Process. **37**, 171–195 (1987)
19. International Commission on Illumination: Recommendations on uniform color space, color-difference equations, psychometric color terms, Supplement No. 2 to CIE Publication No. 15 (E-1.3.1/(TC-1.3)) (1978)
20. Taubin, G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In: Proc. Fifth International Conference on Computer Vision (ICCV'95), pp. 902–907 (1995)
21. do Carmo, M.: Differential Geometry of Curves and Surfaces. Prentice-Hall, New York (1976)
22. Olshausen, B.A., Anderson, C., van Essen, D.: A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. J. Neurosci. **13**(11), 4700–4719 (1993)

23. Hoffman, D., Singh, M.: Salience of visual parts. Cognition **63**, 29–78 (1997)
24. Lindeberg, T.: Scale-space theory: a basic tool for analysing structures at different scales. J. Appl. Stat. **21**(2), 224–270 (1994)
25. Desbrun, M., Meyer, M., Schroder, P., Barr, A.H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: ACM SIGGRAPH'99, pp. 317–324 (1999)
26. Lee, C.H., Varshney, A., Jacobs, D.: Mesh saliency. In: ACM SIGGRAPH'05, pp. 659–666 (2005)
27. Optical Society of America: Uniformly Spaced Color Samples. Washington, DC (1977)
28. Billmeyer, F.W., Saltzman, M. Jr.: Principles of Color Technology, 2nd edn. Wiley, New York (1981)
29. Conway, B.R.: Neural Mechanisms of Color Vision: Double-Opponent Cells in the Visual Cortex. Springer, Berlin (2002)
30. Bian, Z.Q., Zhang, W.G.: Pattern Recognition. Tsinghua University Press, Beijing (1999)
31. Lv, L.: 3D model retrieval based on invariants of hierarchical geometry transformations. Master Thesis, Department of Computer Science, Tsinghua University (2010)
32. Elbaz, A.E., Kimmel, R.: On bending invariant signatures for surfaces. IEEE Trans. Pattern Anal. Mach. Intell. **25**(10), 1285–1295 (2003)
33. Xu, D., Li, H.: Geometric moment invariants. Pattern Recognit. **41**(1), 240–249 (2008)
34. Liu, Y.J., Chen, Z.Q., Tang, K.: Construction of iso-contours, bisectors and Voronoi diagrams on triangulated surfaces. IEEE Trans. Pattern Anal. Mach. Intell. (2011). doi:10.1109/TPAMI.2010.221
35. McGill: 3D Shape Benchmark. http://www.cim.mcgill.ca/~shape/benchMark/ (2011)
36. Mou, W., Xiao, C., McNamara, T.P.: Reference directions and reference objects in spatial memory of a briefly viewed layout. Cognition **108**, 136–154 (2008)
37. van Rijsbergen, C.: Information Retrieval, 2nd edn. Dept. of Computer Science, Univ. of Glasgow, Glasgow (1979)
38. Silverman, B.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)
39. Turlach, B.A.: Bandwidth selection in kernel density estimation: a review. Discussion Paper 9317, Institut de Statistique, UCL, Louvain La Neuve, 1993
40. Jayanti, S., Iyer, N., Kalyanaraman, Y., Ramani, K.: Developing an engineering shape benchmark for cad models. Comput. Aided Des. **38**(9), 939–53 (2006)
41. Wang, J., He, Y., Tian, H., Cai, H.: Retrieving 3d cad model by free-hand sketches for design reuse. Adv. Eng. Inform., **22**(3), 385–392 (2008)
42. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. Int. J. Comput. Vis. **40**(2), 99–121 (2000)

Yong-Jin Liu received his Ph.D. from the Hong Kong University of Science and Technology, Hong Kong, in 2003. He is now an Associate Professor with the Department of Computer Science and Technology, Tsinghua University Beijing, China. His research interests include computer graphics and computer-aided design.

Yi-Fu Zheng received his B.Eng. from Tsinghua University, Beijing, in 2010. He is now a master student in the Department of Computer Science, Columbia University, New York, United States. He enrolled Tsinghua University in 2006 and his research interests focus on computer graphics.



Lu Lv received his masters degree in Computer Science and Technology from the Tsinghua University, Beijing, China in 2010. He is now a software engineer of game development. His research interests focus on computer graphics.

Yu-Ming Xuan received his Ph.D. from Institute of Psychology, Chinese Academy of Sciences, in 2003. He is now an Associate Professor in State Key Lab of Brain and Cognitive Science, Chinese Academy of Sciences. His research interests includes visual cognition and human-computer interaction.

Xiao-Lan Fu is a Professor in Institute of Psychology, Chinese Academy of Sciences. She received her B.S. and M.S. in Psychology from Peking University in 1984 and 1987, and her Ph.D. in Institute of Psychology, Chinese Academy of Sciences, in 1990. Her research interests includes human cognitive processing, affective computing, and human-computer interaction.