

A novel sketch-based 3D model retrieval approach based on skeleton

Jing Zhang, Bao sheng Kang, Bo Jiang, Di Zhang

Northwest University/Department of Information Science and Technology, China

Article Info

Article history:

Received Oct 9, 2018

Revised Dec 20, 2018

Accepted Jan 11, 2019

Keywords:

Advanced skeleton strength map
Histogram feature comparison
Skeleton features
Sketch-based 3D Model
Retrieval

ABSTRACT

Since the skeleton represents the topology structure of the query sketch and 2D views of 3D model, this paper proposes a novel sketch-based 3D model retrieval algorithm which utilizes skeleton characteristics as the features to describe the object shape. Firstly, we propose advanced skeleton strength map (ASSM) algorithm to create the skeleton which computes the skeleton strength map by isotropic diffusion on the gradient vector field, selects critical points from the skeleton strength map and connects them by Kruskal's algorithm. Then, we propose histogram feature comparison algorithm which adopts the radii of the disks at skeleton points and the lengths of skeleton branches to extract the histogram feature, and compare the similarity between two skeletons using the histogram feature matrix of skeleton endpoints. Experiment results demonstrate that our approach which combines these two algorithms significantly outperforms several leading sketch-based retrieval approaches.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Jing Zhang,
Department of Information Science and Technology,
Northwest University,
Chang An Qu., Xi'an, China.
Email: mywangwang123@163.com

1. INTRODUCTION

With the rapid increase in the number of available 3D models, accurately and effectively search 3D model is important for many applications such as industrial design, engineering, and manufacturing area. The most popular way for retrieving 3D models is example-based paradigm, where the user provides an existing 3D model as query input and the retrieval system can return similar 3D models from the database [1]. However, it is difficult for a user to have an appropriate example 3D model at hand. An alternative way is to use 2D sketch as a query where users can describe a target 3D model by quickly drawing it. But a 2D sketch is merely a coarse and simple representation which only contains partial information of an original 3D model. Hence, it is more challenging to realize a sketch-based retrieval than an example-based retrieval [2].

The skeleton is widely recognized as one of the most important shape descriptors in sketch-based 3D model retrieval process. The skeleton extraction algorithms generally are classified into five types: the thinning algorithm [3], the discrete domain algorithm based on the Voronoi diagrams [4], the analytic computation of the object contour algorithm [5], the algorithm based on mathematical morphology [6], and the algorithm based on distance transform [7]. Bertrand G et al [8] found that the thinning algorithm is quite sensitive to noise and can achieve large amount of calculation, which is expected to eliminate local noise without introducing distortions and retain significant features of the pattern. Giesen J et al [9] researched on the discrete domain algorithm based on the Voronoi diagrams which calculates the Voronoi diagrams via a sufficient large set of points sampled from its boundary. If the amount of sampled points is extremely large, it would lead to the numerical instability in the computation of Voronoi diagrams. Krinidis S et al [10] found that the analytic computation of the object contour algorithm would lead to the algorithm's inaccuracy due to

the polygonal approximation is mostly not sufficient for biological shape description. Mitra A et al [11] found that the algorithm based on mathematical morphology can localize the accurate skeleton, but may not guarantee the exact connectivity of the skeleton. Yan T Q et al [12] found that the algorithm based on distance transform compute the skeleton by detecting ridges on the distance transform surface, which ensures the accurate localization of skeleton points. However, the algorithm based on distance transform is lack of connectivity and completeness, which means, the extracted branches may be disconnected and not be able to represent all the significant visual parts.

How to compare the skeleton similarity between the query sketch and 2D views of 3D model is crucial. There are three types of algorithms to compare the skeleton similarity, respectively tree matching algorithm [13], path similarity algorithm [14] and graph representation algorithm [15]. Dragan F F et al [16] found that the tree descriptor represents the topological features of the skeleton, but the maximal isomorphic subtrees are obtained by searching for the longest matching substrings which would lead to higher time complexity. Bai X et al [14] matched the skeleton graphs by comparing the geodesic paths among skeleton endpoints. However, their approach is only motivated by the visually similar skeleton graphs, without considering the topological graph structure. Youssef R et al [17] found that the graph representation algorithm is widely investigated regarding matching issues, due to the correspondence between skeleton branches with graph edges and nodes is natural and intuitive. However, the solutions in the literature are based on shock graphs [18] or attributed relational graphs [19] which are only used in certain application domains and usually with higher time complexity.

To tackle these problems, this paper proposes a novel sketch-based 3D model retrieval approach which combines the advanced skeleton strength map (ASSM) algorithm with histogram feature comparison algorithm to create the skeleton and compare the similarity. Our ASSM algorithm origins from the algorithm based on distance transform, which ensures the accurate localization of skeleton points. Further, our ASSM algorithm selects critical points from the skeleton strength map and connects them by Kruskal's algorithm [20], which solves the connectivity and completeness problem of previous algorithm. The histogram feature comparison algorithm origins from the graph representation algorithm, which expresses the skeleton branches naturally and intuitively [21]. Our histogram feature comparison algorithm adopts the radii of the disks at skeleton points and the lengths of skeleton branches to extract the histogram feature. The radii of the disks and lengths of skeleton branches are invariant under the environment of non-rigid transformations which would help to get high-precision retrieval result. Additionally, we compare the similarity between two skeletons using the histogram feature matrix of skeleton endpoints, with relatively less quantity than skeleton points, which leads to a lower computational complexity. To evaluate our approach, we test our approach on the public standard dataset and also compared with other leading 3D model retrieval approaches. The experiments demonstrate that our approach is significantly better than several leading sketch-based retrieval approaches.

2. RESEARCH METHOD

The framework of sketch-based 3D model retrieval based on advanced skeleton strength map and histogram feature comparison algorithm is proposed as shown in Figure 1.

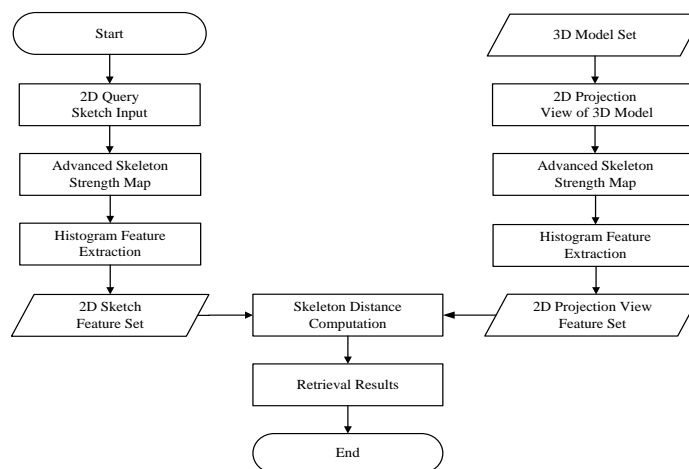


Figure 1. The framework of our sketch-based 3D model retrieval

2.1. Pre-processing

In the database each 3D model has an arbitrary position, orientation and scale in the spatial space, it is necessary to normalize each 3D model before project them into 2D views. After 3D models in the database have been normalized, we compare the query sketch with 60 projection views of each 3D model using light field descriptors [22]. In addition, we utilize 2D sketch-3D model alignment algorithm to select some candidate views from the 2D projection views for efficiently comparing the similarity.

Our light fields are distributed uniformly and positioned on vertices of a regular dodecahedron. For regular dodecahedron, each of the 20 vertices is connected by 3 edges, which results in 60 different rotations ($20 \text{ vertices} \times 3 \text{ edges}$) for our projections, so that we need rotate the camera system 60 times. When the cameras switch onto different vertices [23] we measure the similarity between 3D model and sketch. Here is a typical example to explain our approach. As shown in Figure 2, in the left is a pig 3D model, in the right display 60 projection views which are rendered from vertices of a dodecahedron for the pig 3D model.

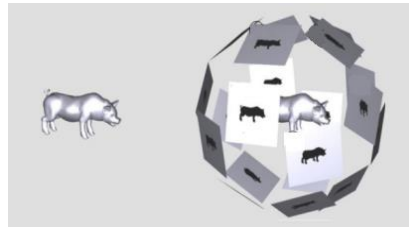


Figure 2. The light field descriptors of 3D model projection

To enhance the retrieval accuracy and performance, we adopt 2D sketch-3D model alignment algorithm [24] to choose the candidate views in the sketch-based 3D model retrieval. We choose candidate views [25] by keeping a certain percentage T with top similarities between the sketch and all the 2D projection views, e.g. $T = 20\%$ means that the number of our candidate views is $60 \times 20\% = 12$.

2.2. Advanced skeleton strength map (ASSM)

The algorithm of skeleton extraction by ASSM is described as follows:

1. Step 1. Extraction of external boundary. For each query sketch and 2D views of 3D models, the external boundary is extracted firstly, which provides much of the image visual information for the SSM value.
2. Step 2. Computation of SSM value [26]. We perform distance transform on boundary and then compute the SSM value by isotropic diffusion on the gradient vector field.
3. Step 3. Refined SSM value. We adopt the non-maximal suppression algorithm [27] for the refined SSM value.
4. Step 4. Selection of critical points. We select the critical points from the refined SSM value.
5. Step 5. Skeleton trace. The final skeleton is obtained by connecting the critical points. We propose a connecting critical points' method which uses the Kruskal's algorithm to decide the order of the connecting path.

2.2.1. Computation of SSM value

We define a function $f(\vec{r})$ to get the accurate skeleton [26] :

$$f(\vec{r}) = 1 - \|\nabla G_{\sigma}(\vec{r}) * dt(\vec{r})\| \quad (1)$$

where $G_{\sigma}(\vec{r})$ represents a Gaussian kernel function, $dt(\vec{r})$ represents distance transform which is the distance from an interior point \vec{r} to the nearest boundary point, σ is its standard covariance, and $*$ is the convolution operator.

We compute the gradient of $f(\vec{r})$ as follow:

$$(u_0, v_0) = \nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \quad (2)$$

The isotropic diffusion of $f(\vec{r})$ is performed:

$$\begin{cases} \frac{du}{dt} = u\nabla^2 u - (u - f_x)(f_x^2 + f_y^2) \\ \frac{dv}{dt} = u\nabla^2 u - (v - f_y)(f_x^2 + f_y^2) \end{cases} \quad (3)$$

where u, v are two components of the diffused gradient vector field. f_x and f_y are the two components of $f(\vec{r})$. Initializing u, v with u_0, v_0 in (2), the partial differential (3) can be solved iteratively by finite difference technique.

The isotropic diffusion makes the vectors of $f(\vec{r})$ propagate towards to the object centre, and the intersections of vectors determine the actual location of the skeleton points.

Then, we consider the initial gradient vector field $gvf(\vec{r})$:

$$gvf(\vec{r}) = (I(\vec{r}) - I(\vec{r}')) \times \frac{\vec{r}' - \vec{r}}{\|\vec{r}' - \vec{r}\|} \quad (4)$$

where $I(\vec{r})$ is the intensity value at \vec{r} , and \vec{r}' represents one of the eight immediate neighbors of \vec{r} .

The SSM value at each point indicates the probability of being a skeleton point. The higher value at a point, the more probable this point is a skeleton point. We compute the SSM value by:

$$SSM(\vec{r}) = \max(0, \sum_{\vec{r}' \in N(\vec{r})} \frac{gvf(\vec{r}') \cdot (\vec{r}' - \vec{r})}{\|\vec{r}' - \vec{r}\|}) \quad (5)$$

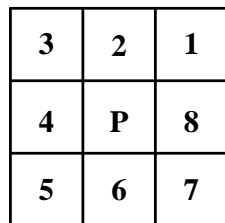
where $N(\vec{r})$ is the set of the eight immediate neighbours of \vec{r} .

2.2.2. Refined SSM value

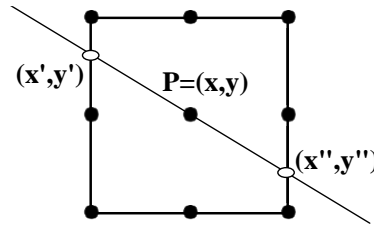
When we have been computed the value of SSM, use non-maximal suppression [27] algorithm to obtain SSM refined edge. The non-maximal suppression algorithm is described as follows:

Figure 3(a) shows the eight directions of point $P = (x, y)$ within the 3×3 region. In Figure 3(b), the quadrangle is formed by connecting the eight directions of point $P = (x, y)$. Then, we use the direction of $gvf(\vec{P})$ to make a straight line, which intersect with the quadrangle at point (x', y') and (x'', y'') .

If $SSM(x, y) > SSM(x', y') \& SSM(x, y) > SSM(x'', y'')$, then the point $P = (x, y)$ will be retained. If not, the point $P = (x, y)$ will be deleted. When all points have been utilized the non-maximal suppression algorithm, we would get the refined SSM value.



(a)



(b)

Figure 3. (a) The eight directions of point, (b) The quadrangle formed by connecting the eight directions of point

2.2.3. Selection of critical points

For the refined SSM value, we select the critical points with the lowest value of the gradient magnitude $\|\nabla G_\sigma(\vec{r}) * dt(\vec{r})\|$.

Our critical points correspond to significant visual parts of the object. Therefore, the obtained skeleton contains branches representing all significant visual parts. Notice that in the definition of the critical point, we also include the endpoints, because those points usually do not have minimum gradient magnitude. If they are not selected, the skeleton branches may be shortened.

2.2.4. Skeleton trace

We consider the critical points' Euclidean distance matrix to represent an undirected weighted graph. Each critical point chooses the top 3 minimum Euclidean distance with other points in the process of Kruskal's algorithm [20]. The Kruskal's algorithm adds edges by weight ascending order, which forms a tree that includes every vertex. The total weight of all the edges in the tree is minimized. Below are the steps of connecting critical points using Kruskal's algorithm.

Assume $G = (V, \{E\})$ represent graph which formed by critical points' Euclidean distance. Where V is the critical points set of the graph G , and $\{E\}$ is the edge set of the critical points' Euclidean distance. We set the number of critical points is N , and sort all the edges by weight ascending order.

1. Step1. We define a set of N independent vertices. Since N critical points need be connected, we consider the N critical points separately.
2. Step2. We choose the edge by weight ascending order. If the edge of two vertices satisfies in different vertex sets, we add this edge to the minimum spanning tree's edge set, and merge the two different vertex sets into one vertex set. If not, we would consider the vertices of next edge.
3. Step3. Repeat step2 until all the vertices are in the same vertex set.

As shown in Figure 4 (a), it is an example to explain our approach which includes 7 critical points to use the Kruskal's algorithm. In Figure 4 (b), the Euclidean distance between 7 critical points represents an undirected complete graph. In Figure 4(c), it's the minimum spanning tree of 7 critical points which use the Kruskal's algorithm to create.

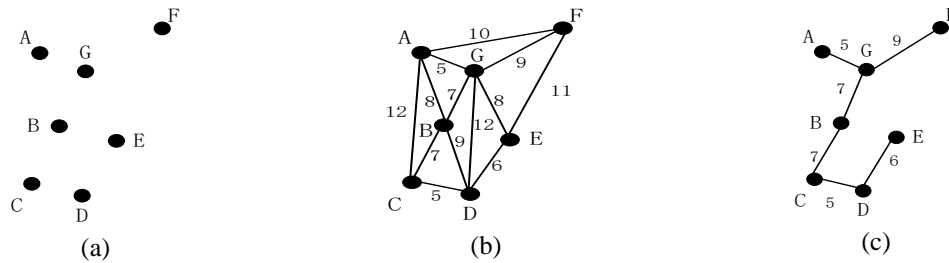


Figure 4. (a) An example including 7 critical points, (b) The Euclidean distance between 7 critical points, (c) The minimum spanning tree of Kruskal's algorithm

2.3. Histogram Feature Comparison

Our histogram feature comparison algorithm consists of histogram feature extraction and skeleton distance computation algorithm. The histogram feature extraction algorithm adopts the radii of the disks at skeleton points and the lengths of skeleton branches to extract the histogram feature. The radii of the disks and lengths of skeleton branches are invariant under the environment of non-rigid transformation [28] which would help to get high-precision retrieval result. Additionally, the skeleton distance computation algorithm compares the similarity between two skeletons using the histogram feature matrix of skeleton endpoints, with relatively less quantity than skeleton points which leads to a lower computational complexity.

2.3.1. Histogram Feature Extraction

We define a 2D dataset $S = \{(a_i, b_i) | i = 1, 2, \dots, T\}$, and the $H(S, X)$ represents the 2D histogram matrix of S , where $X = \{(x_i, y_j) | i = 0, 1, \dots, m; j = 0, 1, \dots, n\}$ is the histogram parameter, with constraints $x_{i-1} < x_i, y_{i-1} < y_i, a_i \in [x_i, x_m], b_j \in [y_j, y_n]$. Define the 2D histogram matrix $H(S, X)$ as follow:

$$H(S, X) = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & & \vdots \\ h_{m1} & h_{m2} & \cdots & h_{mn} \end{bmatrix} \quad (6)$$

where h_{ij} represents the number of S dataset elements distributing in the rectangular area $([x_{i-1}, x_i], [y_{j-1}, y_j])$, x_i, y_j usually is arithmetic sequence or geometric sequence.

As shown in Figure 5, the point Q represents the skeleton centre which is the skeleton point with radius of the maximal disk, and $v_i (i = 1, 2, \dots, T)$ is the set of skeleton points. Then we use feature dataset P_Q to replace the above 2D dataset S :

$$P_Q = \{p_Q(v_i) \mid p_Q(v_i) = (ske(Q, v_i), R(v_i))\} (i = 1, 2, \dots, T) \quad (7)$$

where $ske(Q, v_i)$ is the shortest path from skeleton centre Q to skeleton point v_i . $R(v_i)$ represents the radii of disks of skeleton point v_i . v'_i is the corresponding point of v_i at the contour. $p_Q(v_i)$ represents the relative distance from skeleton centre Q to contour point v'_i .

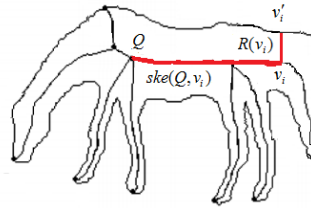


Figure 5. The radius of the maximal disk and contour point

For the 2D histogram feature matrix $H(P_Q, X)$, X is the corresponding histogram parameter.

$$X = \{(x_i, y_j) \mid x_i = l^i, y_j = r^j\} (i = 0, 1, \dots, m; j = 0, 1, \dots, n) \quad (8)$$

The determinations of parameters l, r, m, n are under the constraint of radii of disks and the lengths of skeleton branches.

As shown in Figure 6, we consider the endpoint A as the base research point, the definition of P_A is similar to P_Q . The $H(P_A, X)$ represents the histogram feature matrix.

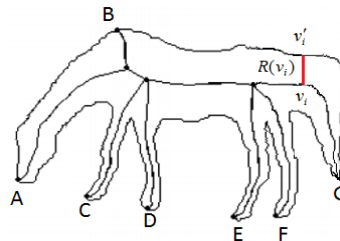


Figure 6. The shortest path of skeleton endpoints

The Figure 7 shows the 2D histogram of skeleton endpoint A generated by the length of all curves distributing in each rectangular area. The $ske(A, v_i)$ is the abscissa which represents the shortest path from the endpoint A to skeleton point v_i . The $R(v_i)$ is the ordinate which represents the radius of disk of skeleton point v_i .

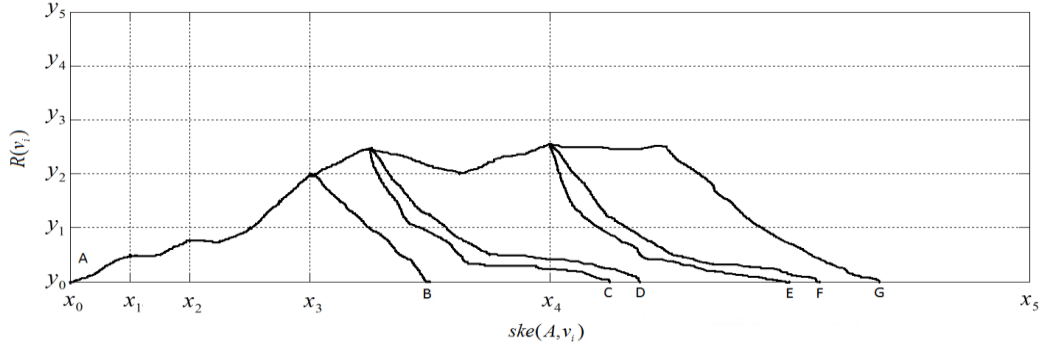


Figure 7. The 2D histogram of skeleton endpoint A

2.3.2. Skeleton Distance Computation

We define two skeletons $G=(v_1, v_2, \dots, v_m)$ and $G'=(v'_1, v'_2, \dots, v'_n)$, where v_i and v'_j represent endpoints from different skeletons. With the histogram feature matrix $H_i(k)$ and $H'_j(k)$ of skeleton endpoints v_i and v'_j , we get the distance between 2D histogram of two skeleton endpoints:

$$d(v_i, v'_j) = \sum_{k=1}^{mn} w_k |H_i(k) - H'_j(k)| \quad (9)$$

where

$$w_k = \frac{\cos\left(\frac{H_i(k) + H'_j(k)}{Z} \frac{\pi}{2}\right)}{\sum_{k=1}^{mn} \cos\left(\frac{H_i(k) + H'_j(k)}{Z} \frac{\pi}{2}\right)} \quad (10)$$

$$Z = 2 \max_k (H_i(k) + H'_j(k)) \quad (11)$$

satisfying $\frac{H_i(k) + H'_j(k)}{Z} < 1$.

When we compare the similarity between two skeletons, choose one of the skeletons as a benchmark, and compute all the distance with the other skeleton endpoints.

$$D(G, G') = \begin{bmatrix} d(v_1, v'_1) & d(v_1, v'_2) & \dots & d(v_1, v'_n) \\ d(v_2, v'_1) & d(v_2, v'_2) & \dots & d(v_2, v'_n) \\ \vdots & \vdots & & \vdots \\ d(v_m, v'_1) & d(v_m, v'_2) & \dots & d(v_m, v'_n) \end{bmatrix} \quad (12)$$

In the all endpoints distance matrix $D(G, G')$, we select the minimum value of each line $\min d(v_1, v'_j), \min d(v_2, v'_j), \dots, \min d(v_m, v'_j) (j = 1, 2, \dots, n)$ to compute the distance between two skeletons. Then the distance of two skeletons is expressed as:

$$\text{dis}(G, G') = \sum_{i=1}^m \min_j d(v_i, v'_j), j = 1, 2, \dots, n \quad (13)$$

3. RESULTS AND ANALYSIS

We implement our sketch-based 3D model retrieval method in C++ under Windows. As shown in Figure 8, The left side of the interface is a canvas for sketching the model. The user can erase and modify if they do not satisfy with the drawn sketch. The right side is displaying page for the retrieved 3D models with a relevant JPEG image. The user can click the blue button to download the corresponding 3D model.

The system consists of off-line feature extraction and on-line retrieval processes. In the off-line process, the features are extracted in a PC with a Pentium III 800MHz CPU and GeForce2 MX video card. In the on-line process, the retrieval system consists by a PC with an Intel Xeon CPU E5520@2.27 GHz and 12.0 GB of RAM. Our sketch-based 3D model retrieval benchmark is built on the well-known National Taiwan University(NTU) [29] database and the latest collection of human sketches.

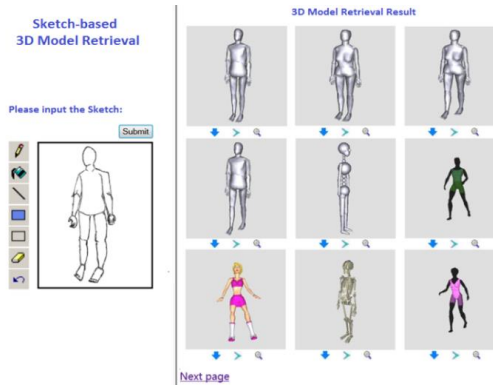


Figure 8. Our sketch-based 3D model retrieval system

3.1. Skeleton Extraction by ASSM

The proposed ASSM method is able to compute skeleton branches in all significant visual parts. Figure 9 shows leaf skeleton extraction process by ASSM approach. As shown in Figure 9(a), it's the original leaf image. In Figure 9(b), shows the computation of the SSM value after isotropic diffusion on the gradient vector field. In Figure 9(c), the SSM value is refined by the non-maximal suppression algorithm. In Figure 9(d), the critical points set extracted from the SSM refined value. In Figure 9(e), shows the final skeleton connected by Kruskal's algorithm.

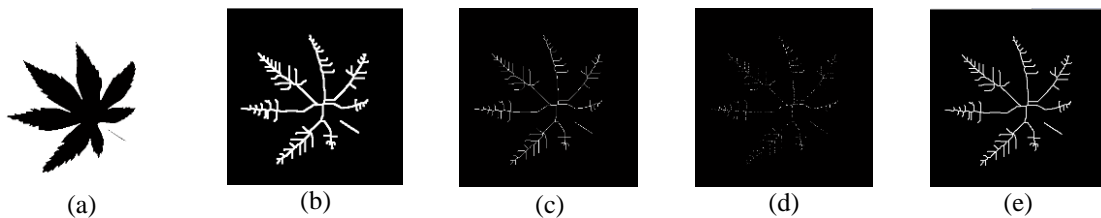


Figure 9. Illustration of leaf skeleton extraction process by ASSM approach
(a) the original leaf image, (b) shows the SSM value, (c) the SSM value is refined by the non-maximal suppression algorithm, (d) the critical point set extracted from (c), (e) the final skeleton connected by Kruskal's algorithm

3.2. Skeleton Matching Based on Histogram Feature

In order to show the effectiveness of the proposed histogram feature comparison algorithm, we tested three skeleton matching experiments between sketch and 2D view of 3D model. Figure 10 shows the correspondence between a cat and another deformed cat. It demonstrates that our histogram feature comparison algorithm can realize the skeleton matching of deformed object. Figure 11 shows the correspondence between the persons with different numbers of arms. It illustrates that our histogram feature comparison algorithm works correctly if object parts are significantly altered. Figure 12 shows the correspondence between an elephant and another elephant with a stick. It demonstrates that our matching process has strong performance when the compared objects have the redundant parts.

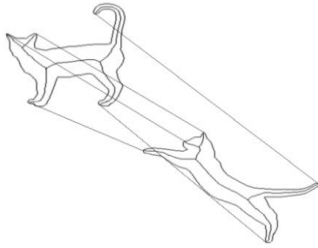


Figure 10. The correspondence between a cat and another deformed cat

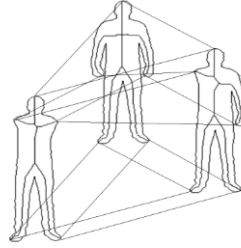


Figure 11. The correspondence between the persons with different numbers of arms

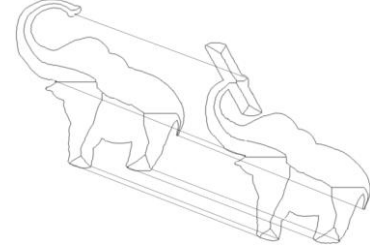


Figure 12. The correspondence between an elephant and another elephant with a stick

3.3. Comparison with other approaches

We considered classical Precision and Recall metrics averaged over the set of processed queries[30] to measure the retrieval effectiveness. Recall measures the ability of the system to retrieve all models that are relevant. Precision measures that the ability of the system to retrieve only models that are relevant. They are defined as:

$$Recall = \frac{\text{relevant correctly retrieved}}{\text{all relevant}} \quad (14)$$

$$Precies = \frac{\text{relevant correctly retrieved}}{\text{all retrieved}} \quad (15)$$

To have a comprehensive evaluation of our algorithm, we further provide the results for other performance metrics including Nearest Neighbour (NN), First Tier (FT), Second Tier (ST), E-measure (E), Discounted Cumulative Gain (DCG) and Average Precision (AP). The meaning of the above performance metrics is as follows [30]. NN measures the percentage of the closest matches that are relevant models. FT represents how much percentage of a class has been retrieved among the top C list, where C is the cardinality of the relevant class of the query sketch. It defines as:

$$FT = \frac{\text{relevant correctly retrieved}}{\text{top } (C - 1) \text{ retrieved}} \quad (16)$$

ST represents how much percentage of a class has been retrieved among the top $2(C - 1)$ list, where C has the same meaning with FT metric.

$$ST = \frac{\text{relevant correctly retrieved}}{\text{top } 2(C - 1) \text{ retrieved}} \quad (17)$$

E is used to measure the performance of the retrieval results with a fixed length, e.g. the first 32 models. It combines both the Precision P and Recall R :

$$E = \frac{2}{\left(\frac{1}{P} + \frac{1}{R}\right)} \quad (18)$$

DCG is defined as the summed weighted value related to the positions of the relevant models.

$$DCG = w_1 + \sum_{k=2}^P \frac{w_k}{\log_2 k} \quad (19)$$

where w_k denote weighted value of each retrieval result, and k denotes the index of retrieval result. P is the number of retrieval result.

AP can be computed by counting the total area under the Precision-Recall curve. The higher Precision-Recall curve would get a better AP value. We compare our approach with other four leading sketch-based 3D model retrieval algorithms, which utilize skeleton characteristics as the features to describe the object shape. Sundar H et al. [31] are the most representative in the field of skeleton based shape matching and retrieval. They utilized the thinning algorithm and clustering algorithm which help lessen the effect of many small perturbations on the surface and reduce the number of nodes necessary for skeletal graph construction. Lei H et al. [32] used thinning algorithm on the silhouette images to extract the corresponding skeletons and utilized skeleton pruning method. Lin S et al. [33] got the skeleton of 3D model through skeleton extraction algorithm based on mesh simplification and mesh contraction. Sirin Y et al. [34] started by drawing circles of increasing radius around skeletons, which lead to each skeleton corresponded to the centre of a maximally inscribed circle. We illustrate 3D models of car, lamp, plane and chair in the Figure 13 and the average comparison result is shown in Table 1.

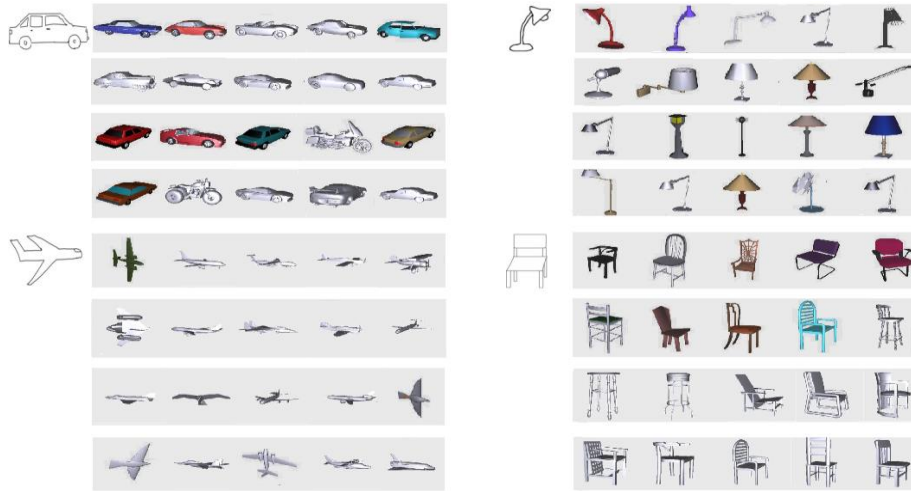


Figure 13. Examples of sketch-based retrieval results

Table 1. Metrics for the Performance Comparison between Our Approach and Other Approaches

Approaches	NN	FT	ST	E	DCG	AP
Our Approach	0.387	0.316	0.383	0.374	0.589	0.394
Lei H's Approach	0.365	0.268	0.331	0.352	0.541	0.335
Sundar H's Approach	0.347	0.251	0.327	0.321	0.533	0.326
Lin S's Approach	0.285	0.236	0.294	0.268	0.491	0.278
Sirin Y's Approach	0.211	0.208	0.243	0.202	0.466	0.204

It is obvious that our approach outperforms other leading sketch-based 3D model retrieval approaches. The approaches of Sundar H et al [31] and Lei H et al [32] utilized the thinning algorithm which is quite sensitive to noise and can achieve large amount of calculation. Lin S et al [33] used the front view of

the skeleton to represent 3D models, in this way 3D models can be represented in 2D form, which is not sufficient for 3D model description. In contrast, we compare the query sketch with 60 projection views of each 3D model in the Sec.2.1. Sirin Y et al [34] computed the total number and ratio of pixels which help to distinguish shapes with similar skeletons. Since it is necessary to normalize each 3D model with high standard, the comparison algorithm based on the pixels is inaccuracy. Figure 14 shows the Precision-Recall plots of our approach with other four leading sketch-based 3D model retrieval algorithms. The experiments demonstrate that our approach is significantly better than any other retrieval techniques.

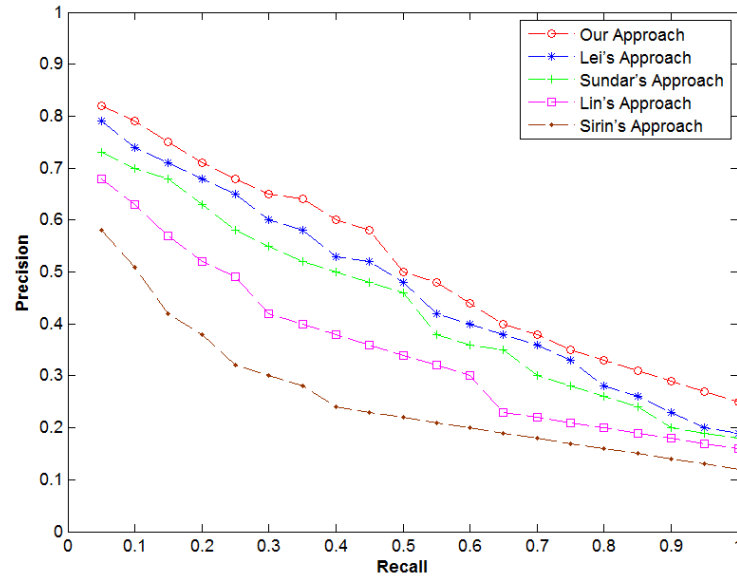


Figure 14. Overall retrieval performance comparison result

4. CONCLUSION

In this paper, we proposed a novel sketch-based 3D model retrieval approach which combines ASSM and histogram feature comparison algorithms. Firstly, the ASSM algorithm is proposed for extracting the skeleton of 2D query sketch and 2D view of 3D model. The ASSM algorithm computes the skeleton strength map by isotropic diffusion on the gradient vector field, selects critical points from the skeleton strength map and connects them by Kruskal's algorithm, which solves the connectivity and completeness problem of previous algorithm. Then, the histogram feature comparison algorithm adopts the radii of the disks at skeleton points and the lengths of skeleton branches to extract the histogram feature. The radii of the disks and lengths of skeleton branches are invariant under the environment of non-rigid transformation which would help to get high-precision retrieval result. Additionally, we compare the similarity between two skeletons using the histogram feature matrix of skeleton endpoints, with relatively less quantity than skeleton points, which leads to a lower computational complexity. Experiment results demonstrate that our approach which combines these two algorithms significantly outperforms several leading sketch-based retrieval approaches.

REFERENCES

- [1] Wang, F., Lin, L. F. and Tang, M. A new sketch-based 3D model retrieval approach by using global and local features. *Graphical Models*. 2014, 76(3):128-139.
- [2] Shao T, Xu W, Yin K, et al. Discriminative Sketch-based 3D Model Retrieval via Robust Shape Matching. *Computer Graphics Forum*. 2011, 30(7):2011-2020.
- [3] Tabedzki M, Saeed K, Szczepański A. A modified K3M thinning algorithm. *International Journal of Applied Mathematics & Computer Science*. 2016, 26(2):439-450.
- [4] Liu Y J, Chen Z, Tang K. Construction of Iso-Contours, Bisectors, and Voronoi Diagrams on Triangulated Surfaces. *IEEE Transactions on Pattern Analysis & Machine Intelligence*. 2011, 33(8):1502-17.
- [5] Gorelick L, Galun M, Sharon E, et al. Shape representation and classification using the Poisson equation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*. 2004, 2(12):1991-2005.
- [6] Morales S, Naranjo V, Angulo J, et al. Determination of retinal network skeleton through mathematical morphology. *Signal Processing Conference IEEE*. 2014:1691-1695.

- [7] Guo Y, Sengur A. A novel 3D skeleton algorithm based on neutrosophic cost function. *Applied Soft Computing*. 2015, 36(C):210-217.
- [8] Bertrand G, Couprie M. Isthmus Based Parallel and Symmetric 3D Thinning Algorithms. *Graphical Models*. 2015, 80(C):1-15.
- [9] Giesen J, Miklos B, Pauly M. The medial axis of the union of inner Voronoi balls in the plane. *Computational Geometry*. 2012, 45(9):515-523.
- [10] Krinidis S, Chatzis V. A Skeleton Family Generator via Physics-Based Deformable Models. *Image Processing IEEE Transactions on*. 2009, 18(1):1-11.
- [11] Mitra A, Roy S, Setua S K. Shape analysis of decisive objects from an image using mathematical morphology. *Third International Conference on Computer, Communication, Control and Information Technology, IEEE*. 2015:1-6.
- [12] Yan T Q, Zhou C X. A Continuous Skeletonization Method Based on Distance Transform. *Communications in Computer & Information Science*. 2012, 304:251-258.
- [13] Liu J, Liu W, Wu C. Objects Similarity Measurement Based on Skeleton Tree Descriptor Matching. *IEEE International Conference on Computer-Aided Design and Computer Graphics*. 2007:96-101.
- [14] Bai X, Latecki L J. Path Similarity Skeleton Graph Matching. *IEEE Transactions on Pattern Analysis & Machine Intelligence*. 2008, 30(7):1282-92.
- [15] Qiao G, Zong G, Sun M, et al. Automatic neutrophil nucleus lobe counting based on graph representation of region skeleton. *Cytometry Part A the Journal of the International Society for Analytical Cytology*. 2012, 81A (9):734-742.
- [16] Dragan F F. Tree-Like Structures in Graphs: A Metric Point of View. *Graph-Theoretic Concepts in Computer Science*. Springer Berlin Heidelberg. 2013:1-4.
- [17] Youssef R, Kacem A, Sevestre-Ghalila S, et al. Graph Structuring of Skeleton Object for Its High-Level Exploitation. *Image Analysis and Recognition*. Springer International Publishing. 2015:419-426.
- [18] Pandit P M, Akojwar S G. Performance Evaluation Of Object Recognition Using Skeletal Shock Graph: Challenges And Future Prospects. *International Journal of Advances in Engineering & Technology*. 2012.
- [19] Ruberto C D. Recognition of shapes by attributed skeletal graphs. *Pattern Recognition*. 2004, 37(1):21-31.
- [20] Guohua Geng. *Data structure*. Beijing: Higher Education Press. 2011:241-242.
- [21] Saavedra J M, Bustos B. An Improved Histogram of Edge Local Orientations for Sketch-Based Image Retrieval. *Pattern Recognition*. Springer Berlin Heidelberg. 2010:432-441.
- [22] L. McMillan, G. Bishop. Plenoptic Modeling: An Image-Based Rendering System. *Proc. of ACM SIGGRAPH*. 1995: 39-46.
- [23] D. Chen, X. Tian, Y. Shen, M. Ouhyoung. On visual similarity based 3D model retrieval, *Computer Graphics Forum*. 2003, 22 (3):223-232.
- [24] Bo Li, Henry Johan. Sketch-based 3D model retrieval by incorporating 2D-3D alignment. *Multimedia Tools & Applications*. 2013, 65(3):363-385.
- [25] Huttenlocher D, Ullman S. Object recognition using alignment. In: *Proc. of IEEE international conference on computer vision (ICCV)*. 1987: 102-111.
- [26] Q Li, X Bai, W Liu. Skeletonization of gray-scale image from incomplete boundaries, In *International Conference on Image Processing 2008*, 10:877–880.
- [27] J. Canny, A computational approach to edge detection, *IEEE Trans. PAMI* 1986, 8(6): 679-698.
- [28] H.C. LonguetHiggins. The symmetry groups of non-rigid molecules. *Molecular Physics*. 1963, 6(6):1079-1086.
- [29] NTU 3D Model Database, ver.1 <http://3d.csie.ntu.edu.tw/>.
- [30] Shilane P, Min P, Kazhdan M, Funkhouser T. The Princeton shape benchmark. In: *Proc. of shape modelling and applications*, 2004:167-178.
- [31] Sundar H, Silver D, Gagvani N, et al. Skeleton Based Shape Matching and Retrieval[C]. *Shape Modeling International*. IEEE, 2003:130.
- [32] Lei H, Li Y, Chen H, et al. A novel sketch-based 3D model retrieval method by integrating skeleton graph and contour feature[J]. *Journal of Advanced Mechanical Design Systems & Manufacturing*, 2015, 9(4).
- [33] Lin S, Guo Y, Liang Y, et al. 3D Model retrieval based on skeleton[C]. *IEEE International Conference on Networking, Architecture and Storage*. IEEE, 2015:321-325.
- [34] Sirin Y, Demirci M F. 2D and 3D shape retrieval using skeleton filling rate[J]. *Multimedia Tools & Applications*, 2016:1-26.