A novel sketch-based 3D model retrieval method by integrating skeleton graph and contour feature

Haopeng LEI∗,∗∗, Yuhua LI∗∗∗, Helian CHEN†, Shujin LIN††, Guifeng ZHENG††† and Xiaonan LUO†††∗∗

∗ School of Computer and Information Engineering, Jiangxi Normal University
No.99, Ziyang Road, Nanchang, Jiangxi Province, 330022, China

∗∗ National Engineering Research Center of Digital Life, School of Information Science & Technology, Sun Yat-sen University
No.132 East Waihuan Road, Panyu District, Guangzhou 510006, China

∗∗∗ Software Engineering College, Zhengzhou University of Light Industry
No.5 Dongfeng Road, Zhengzhou, Henan Province, 450002, China

† School of Software, Sun Yat-sen University
No.132 East Waihuan Road, Panyu District, Guangzhou 510006, China

†† School of Communication & Design, Sun Yat-sen University
Higher Education Mega Education Center, Panyu, 510006 Guangzhou, China

††† Beijing Key Laboratory of Multimedia and Intelligent Software Technology, College of Metropolitan Transportation
Beijing University of Technology
No.100 Ping Le Yuan, Chaoyang District, Beijing, 100124, China
E-mail: zhenggf@mail.sysu.edu

Received 30 December 2014

Abstract

With the rapid growth of 3D models on the Web, effective methods to retrieve appropriate 3D models are becoming crucial. In this paper, we propose a novel sketch-based 3D model retrieval approach which integrates the skeleton graph and contour feature. Skeleton graph represents the topology structure of the query sketch as well as representative 2D views of 3D models, and contour feature captures their detailed information. Moreover, a hierarchical coarse-to-fine searching strategy is employed in this method. Firstly, a skeleton graph based matching method is used to eliminate those 3D models which are dissimilar to the query sketch. Then, the selected candidate models are refined by measuring the similarity of their contour feature. In addition, we adopt a graph transduction method to improve the ranking order of the retrieved 3D models. The proposed method was tested on the public sketch benchmarks and compared with other leading sketch-based 3D model retrieval methods. The experiment results demonstrate that our approach can accurately retrieve desired 3D models for users and has a superior performance to other state-of-the-art approaches.

Key words: Sketch-based 3D model retrieval, Skeleton graph, Contour feature, Rank improving, Graph transduction.

1. Introduction

With the development of 3D scanning and modeling techniques, a large number of 3D models are available on the Web. So designing an effective retrieval system to assist users to find a desired 3D model from a large 3D model database has become an important task in computer graphics applications. 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. However, it is difficult for a user to have an appropriate example 3D model at hand. An alternative way is to use a 2D hand-drawn sketch as a query where users can describe a target 3D model by quickly drawing it. But a 2D hand-drawn 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.
How to develop efficient feature descriptors is the most important part for a sketch-based 3D model retrieval algorithm. Several sketch-based 3D model retrieval algorithms have been proposed recently (Funkhouser, et al., 2003; Chen, et al., 2003; Li, et al., 2013; Yoon, et al., 2010; Saavedra, et al., 2011; Eitz, et al., 2012). These existing techniques can be generally classified into two classes: boundary-based methods and region-based methods. Funkhouser et al. (2003), Chen et al. (2003) and Li et al. (2013) extracted feature descriptors from the exterior boundary to describe the query sketch and 2D views of 3D models, such as the Spherical Harmonics descriptor, Light Field descriptor and Shape Context features. Yoon et al. (2010), Saavedra et al. (2011) and Eitz et al. (2012) divided the query sketch and 2D views of 3D models into several small regions and encoded interior information in each region by using descriptors such as Diffusion tensor, HELO (Histogram of Edge Local Orientations) and GALIF (Gabor Local Line-Based Feature). Very recently, Wang et al. (2014) utilized both the global and local visual features extracted from the input sketch and 2D views of 3D models. In the practical applications, we find that these features are not sufficient and reliable to evaluate the relevance between the query sketch and 3D models. There are basically two disadvantages of the current sketch-based 3D model retrieval methods. First, both the boundary-based and region-based methods only consider describing semantic relationships between the query sketch and 3D models based on a single kind of feature which may cause some serious problems. For example, boundary-based methods ignore the rich information contained in the input sketch and 2D views of 3D models, and cannot recognize their interior content. Region-based methods are unable to discriminate two obviously different sketches when they have multiple similar regions, because this kind of method cannot capture global structural information. Second, a human-drawn sketch is a simple line drawing which has a high level of abstraction, inherent ambiguity and unavoidable noise. Moreover, different users may draw different styles of sketches even if these sketches represent the same 3D model. But, most existing retrieval approaches lack the ability to handle different sketching styles and return the relevant models to the users. Therefore, it is necessary to develop more discriminative and effective feature descriptors that can be applied in the sketch-based 3D model retrieval system.

To tackle these problems, this paper proposes a novel sketch-based 3D model retrieval method which combines the skeleton graph and contour feature. Skeleton graph represents structural information of an object, which can be used to depict the spatial relations between the subparts of the query sketch and 2D views of 3D models (Wu and Liu, 2012). Although different people sketch the same object by using a variety of different styles, the notion of subparts of the object is always the same. For instance, a chair is composed of legs and a seat, and the seat must be drawn above the legs. By adopting skeleton graph as a feature descriptor in the sketch-based 3D model retrieval system, we can determine the basic drawing content of an input sketch and understand what the user is looking for. On the other hand, the input sketch and the 2D views of 3D models are both formed by edge lines. Contour feature can describe the information contained in the edge lines and it can be used to determine the content of query sketch more precisely. Our contour feature captures details about not only external boundaries but also internal structures of the sketch and 2D views of 3D models. Thus, two similar 3D models from the database can be further differentiated with a higher confidence. For these reasons, we combine these two feature descriptors to achieve complementary information. Skeleton graph is used to match at coarse level to first filter out a large amount of 3D models whose views are dissimilar to the input sketch. At a finer level, contour feature is used to select more relevant models from the remaining set. After performing the above steps, we can achieve an initial retrieval result to the query sketch. As mentioned above, a human-drawn sketch is rough and inaccurate, it may show little resemblance to the corresponding real 3D models. But most existing methods only consider measuring the similarity between the sketch and 3D models that may result in an incorrect ranking order of retrieval results. Therefore, we take into account the relationships among 3D models in the initial retrieval result and exploit the graph transduction technique (Bai, et al., 2010) to achieve a new similarity between the query sketch and 3D models. And, we re-rank the order of these models in the initial retrieval result according to the new similarity. Overall, the key contributions of this paper are mainly two-fold:

1. We adopt the integrated feature descriptors by combining the skeleton graph and contour feature which can achieve a higher retrieval accuracy.

2. We employ graph transduction method to re-compute the similarity measurement between the query sketch and 3D models that can improve ranking order of the retrieval results.

To evaluate our method, we tested our system on the public standard dataset and also compared with other leading sketch-based 3D model retrieval approaches. The experimental results demonstrate that our method can achieve better retrieval performance than other previously proposed approaches.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 describes the details of the proposed methodology. In Section 4, the experimental results are discussed. Finally, a brief summary and conclusion.
appear in Section 5.

2. Related Work

Computer aided retrieval of 3D model has been a hot research topic over the last decades. According to the different types of queries for users, 3D model search engine can be classified into three categories: text-based search methods, example-based search methods and 2D sketch-based search methods. Since most of the information exists on the Web in the form of text, text-based search methods have already been adopted in some large online commercial 3D model search engines, such as Google Warehouse. Although this method is simple to implement, it requires humans to annotate every model in the database beforehand. So the retrieval accuracy of text-based methods can be influenced by a multiplicity of different factors, such as language, culture, habit, etc. Besides, it is difficult for users to describe a target model clearly by using few keywords.

Example-based search methods are the most popular way for retrieving 3D models, there exists a huge amount of work on this area, such as shape distribution(Osada, et al., 2002), Reeb graph(Hilaga, et al., 2001), moment(Elad, et al., 2001), heat kernel(Bronstein, et al., 2009), V-system(Song, et al., 2012), etc. A good comprehensive survey of example-based 3D model retrieval methods can be found in Tangelder and Veltkamp(2008). In this paper, we mainly review some related work on example-based 3D model retrieval which also used skeleton graph as the feature descriptor. Sundar et al.(2003) used skeleton graph to encode geometric and topological information of 3D models. They computed a skeletal graph directly from the volumetric object, and connected the skeletal points to generate undirected acyclic shape graph. They matched two graphs by using a greedy algorithm. Then, the best matches for a given 3D model can be queried from the database. Their method also supported matching of articulated objects and partial matching. Iyer et al.(2005) represented a 3D model as a set of voxels and obtained a skeleton graph by eroding voxels iteratively. The skeleton graphs by an algorithm detecting graph/subgraph isomorphism using a decision-tree based method. Mademlis et al.(2008) presented an example-based retrieval method which can be used for partial and global 3D model retrieval. They firstly extracted the medial surface of a model and decompose it into meaningful parts. Then, they converted the media surface to an undirected vertex-attributed graph according to the connectivity of the parts and applied an attributed graph-matching technique to match graphs. This method utilized both topological and geometrical features for 3D model matching.

Compared with other two kinds of 3D model retrieval methods, less work focuses on sketch-based 3D model retrieval. Funkhouser et al.(2003) proposed a 3D model retrieval system which can support 2D hand-drawn sketches as queries. Each 3D model in the database is projected into thirteen 2D views firstly from different orthographic directions. Then they utilized Spherical Harmonics descriptor to computed similarities between the input sketch and each view. Chen et al.(2003) proposed a sketch-based 3D model retrieval system based on the Light Field descriptor. Ten different 2D projections for each 3D model are rendered from vertices of a dodecahedron over a hemisphere. Then, the 2D projections and input sketches are encoded by using Zernike moments and Fourier descriptors. Yoon et al.(2010) obtained 2D views from 3D models by using suggestive contour and computed descriptors by using diffusion tensor fields. They measured similarity between the query sketch and suggestive contour image based on orientation histogram. Saavedra et al.(2011) presented a structure-based local approach (STELA), which used structural and locality information to compare query sketch with the 2D views of 3D models. The proposed descriptors in the method is invariant to position, scale, and rotation changes. Shao et al.(2011) vectorized contours as the 2D shape representation and organized them into transformation graphs. They also developed a sampling-based shape matching algorithm with acceleration strategies. Eitz et al.(2012) proposed a new sketch-based 3D model retrieval system based on bag-of-features framework. They used Gabor filter as a feature extraction tool to encode information of input sketch and 2D views of 3D models, and each 2D view was represented as a histogram of visual word frequency. This method is the first to apply bag-of-features techniques into sketch-based 3D model retrieval. Li et al.(2013) presented a sketch-based 3D model retrieval algorithm that can adaptively select the number of 2D projected views according to the visual complexity of a 3D model. They used viewpoint entropy distribution to measure visual complexity and employed the shape context method to compare the sketch with each 2D view. Wang et al.(2014) provided a sketch-based retrieval approach by utilizing both global feature and local feature. They used Fourier descriptors, Zernike moments, circularity and eccentricity to represent global feature and utilized Speeded Up Robust Features(SURF) as local feature descriptor. This approach can extract more information than other previously proposed approaches, but it requires the user to input a very well-drawn sketch. Li et al.(2014) also made a good survey of sketch-based 3D shape retrieval algorithms and provided a comprehensive comparison for some state-
of-the-art approaches. For domain-specific 3D CAD model retrieval based on hand-drawn sketch inputs, Pu et al.(2005) obtained 2D views by projecting 3D CAD model into the planes of a bounding box and used 2D shape distribution method to compute the similarity between 2D views and input sketches. Hou and Ramani(2007) combined 2.5D spherical harmonic, Fourier descriptors and Zernike moments as shape descriptors to generate multi-classifiers, the probability of the input sketch belonging to each predefined category of 3D CAD model can be estimated by using this classifier. Liu et al.(2013) proposed a sketch-based 3D CAD model retrieval method which can be adaptive to different users’ habits. They used bag-of-features techniques to convert each 3D CAD model into a vocabulary of visual words, and defined the importance weight of each visual word based on sketching history of users.

In summary, feature descriptors adopted in current sketch-based model retrieval approaches are not sufficient to describe the semantic relationships between the query sketch and 3D models. Some methods only consider capturing exterior boundary information of the views or sketches and ignore the rich contour details of their internal structures, such as Spherical Harmonic descriptor(Funkhouser, et al., 2003) and Light Field descriptor(Chen, et al., 2003). Other methods is unable to differentiate the views and sketches having similar inside regions because they do not take into account global topological structural information, such as Diffusion Tensor Fields(Yoon, et al., 2010) and GALIF(Eitz, et al., 2012). Although Wang et al.(2014) utilized both global features and local features to retrieve 3D model, the descriptors used in their method can not handle various sketching styles. Furthermore, all of these existing sketch-based 3D model retrieval methods do not exploit similarities among 3D models from the database to improve retrieval accuracy. So these limitations motivate us to develop a new sketch-based retrieval approach through an integration of the skeleton graph and the highly discriminative contour feature to achieve a better ranking performance by combing with the graph transduction method.

3. The Proposed Methodology
3.1. Overview

The framework of our sketch-based 3D model retrieval system is illustrated in Fig.1. Our algorithm is composed of the following four modules, preprocessing module, feature extraction module, similarity measuring module and rank improving module. In the preprocessing module, each 3D model in the database is firstly normalized to achieve translation, rotation and scale invariance. Then, we select a set of representative viewpoints for each model and adopt hybrid line rendering techniques to generate 2D views. For the user’s input sketch, we perform a sketchy stroke de-noising method on the sketch to remove all kinds of noise that may impact the results of the following module. In the feature extraction module, the skeleton graph and contour feature from each 2D view of the 3D models are extracted respectively. After the user submits a hand-drawn sketch, our system performs the same feature extraction method on this sketch. In the similarity measuring module, we firstly match the skeleton graph between the query sketch and each 2D view of 3D models, which is used to exclude those 3D models whose 2D views are quite different from the input sketch. Then, we compute similarity of contour feature between the query sketch and the remaining 3D models in the database. After that, a set of relevant 3D models can be selected as an initial retrieval result. In the rank improving module, we use graph transduction method on these initial retrieved 3D models to learn a new similarity measure and re-rank these 3D models according to this new similarity measure. The proposed method will be described in detail in the following section.

![Fig. 1 The overview of our proposed sketch-based 3D model retrieval system.](image-url)
3.2. Pre-processing

Since each 3D model in the database 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. The purpose of normalization is to transform all the 3D models from the database into a same canonical coordinate frame by translating, rotating and scaling. One of the normalization methods, PCA has been widely used in the image retrieval (Wang, et al., 2014). However, the size of triangles of the 3D models from the database significantly differ, so we may obtain widely varying normalized coordinate frames by applying PCA normalization method on these 3D models. So we addressed this problem by utilizing an modifications of PCA method-CPCA(Continuous Principal Component Analysis)(Vranic, et al., 2001) to normalize 3D models in our sketch-based retrieval system. For a given 3D model, we first calculate the center of gravity of this model by taking into account the area of each triangle in it. Then, we translate the barycenter to the origin of the coordinate system. Next, we rotate the 3D model into an optimal orientation by performing the standard PCA method and symmetry techniques. In the end, we scale all 3D models into a same unit size.

After 3D models in the database have been normalized, we project them into 2D views that can match to the query sketch. Most existing sketch-based 3D model retrieval algorithms (Funkhouser, et al., 2003; Chen, et al., 2003; Yoon, et al., 2010; Saavedra, et al., 2011; Shao, et al., 2013) predefine a fixed number of viewing directions of each 3D model to generate 2D views. But the user can choose any views when drawing a sketch, the selected viewing directions by the user may not be included in these predefined viewing directions. So it is necessary to predict which viewpoints are most likely to be selected by a user. Obviously, users choose different viewpoints when they depicting different classes of 3D models. For example, most of users prefer to choose a side viewpoint or a front viewpoint to draw a chair, rarely users sketch the chair from a bottom viewpoint. But when sketching a plane, users tend to draw it from a bottom viewpoint rather than a front viewpoint. For these reasons, we need to select a few representative viewpoints to acquire 2D views of a 3D model. In addition, the number of selected representative viewpoints should be small, less representative views mean more fast speed and higher performance. The work of Dutagaci et al.(2010) shows that the representative viewpoints should provide the maximum information of a model. For instance, a plane is more likely to be sketched from a bottom viewpoint rather than a front viewpoint because the bottom viewpoint can capture a higher amount of information of a plane. This method also provides a benchmark for quantitative evaluation of the goodness of a viewpoint. Follow this approach, we takes into account several view descriptors to predict human preferred viewpoints.

We firstly set each 3D model at the center of its bounding sphere and put a virtual camera above this model. Then, we rotate the model 360 degrees at 20 degrees per step(in both longitude and latitude directions) to generate a total of 162 viewing directions that can cover all sides of this model, such as in the Fig.2. To select the candidate viewpoints that the user prefers to choose, we employ six view descriptors on these generated 162 viewpoints to measure the information each viewpoint contains, including projection area(Dutagaci, et al., 2010)-the area of the projection of a 3D model as seen from a particular viewpoint; viewpoint entropy(Vazquez, et al., 2003)-the entropy of the ratio of the projected area of a face to the total projected area of the whole model; silhouette length(Polonsky, et al., 2005)-the length of the silhouette in a projection of a 3D model; silhouette entropy(Page, et al., 2003)-the entropy of the curvature distribution of the projected
silhouette edge as seen from a particular viewpoint; Gaussian curvature entropy (Secord, et al., 2011)-the entropy of the Gaussian curvature distribution over the visible surface of a 3D model and depth distribution (Secord, et al., 2011)-the distribution of the depth of the 3D model as seen from a particular viewpoint. So each viewpoint can be represented by a feature vector, then we use the X-means clustering method (Pelleg and Moore, 2000) to cluster these viewpoint into different classes. Each cluster center can be regarded as a representative viewpoint that the user prefers to choose. The main advantage of using X-means cluster algorithm is that the optimal number of representative viewpoints can be adaptively chosen according to the geometric complexity of a 3D model instead of pre-determining the number of clusters.

After viewpoint selection for each 3D model in the database, the next step is to find appropriate line rendering techniques to generate 2D views under the selected viewing directions. Inspired by Eitz et al. (2012), we combine silhouettes, occluding silhouettes, suggestive contours (DeCarlo, et al., 2003) and Canny lines rendering techniques to generate 2D views for a 3D model. This hybrid line rendering technique can portray the richest information of the model and closely resembles the way most users draw sketches. Figure.3 gives the representative 2D views for a given chair model by using hybrid line rendering techniques.

![Fig. 3 The representative 2D views of a chair model.](image)

Since the limitations of drawing skills of users, a free-hand drawn sketch may contain some noises which influence the effect of feature extraction, such as the sketchy stroke is usually disjoint or form a cross when the user try to draw a closed shape. To remove these noises and restore the original user’s intension, we employ the sketchy stroke denoising method proposed in Jin et al. (2002), which is composed of the following three steps, polygonal approximation, agglomerate points filtering and end point refining. Polygonal approximation is used to reduce redundant points in the sketchy strokes and approximately represents sketchy strokes as polylines. Agglomerate points filtering can be used to smooth circle-like or self-intersection sketchy strokes by examining the point densities. End point refining is used to close a nearly-closed shape properly. After performing these pre-processing steps, we can extract features from the generated 2D views and the query sketch.

### 3.3. Feature Extraction

One of the challenges for developing an effective sketch-based 3D model retrieval system is to define appropriate feature descriptors, which is capable of capturing essential properties of query sketch and 3D models. In our retrieval system, two types of feature descriptors are generated respectively for each representative 2D view of 3D models and query sketch, skeleton graph and contour feature. Below we will describe the algorithms how to extract these two feature descriptors.

#### 3.3.1. Skeleton Graph Extraction

The skeleton graph is an important shape descriptor and it describes geometrical and topological features of an object. We can achieve component parts of an object by analyzing its skeleton. There exist various skeleton extraction methods, such as a thinning algorithm, distance transform, Voronoi diagrams, mathematical morphology and so on. However, these traditional skeleton extraction methods are highly sensitive to the noise and deformation of the object boundary, which will result in the same-class objects having different skeleton representations and create some unwanted branches on the output skeleton. This will cause a disaster when we measure the similarity between the query sketch and 2D views of 3D models. Therefore, we need to remove those redundant branches which are not representing key components to achieve a stable skeleton of an object.

For each 2D view of a 3D model and input sketch (such as in the Fig.4(a)), we first extract their external boundary and generate silhouette images (such as in the Fig.4(b)). Then, we use thinning algorithm (Arcell and Baja, 1985) on the silhouette images to extract the corresponding skeletons. Compared with other skeleton extraction methods, thinning algorithm can preserver more topology information for the original object and is more convenient for the later skeleton extraction.
graph generation. After that, we utilize skeleton pruning method proposed in Shen et al. (2011) to remove redundant skeleton branches. This method prunes the skeleton by using a significance measure called bending potential ratio (BPR). Since each skeleton branch corresponds to a contour segment of the shape, BPR measures the significance of contour segments and can make correct decision about which skeleton branches should be removed. Another advantage is that BPR is insensitive to boundary deformation and noise. For these reasons we adopt this skeleton pruning method in our retrieval to achieve an optimized skeleton for each 2D view of 3D model and input sketch. Figure 4(c) provides the extracted skeleton by performing above steps.

![Figure 4](image_url)

(a) The input sketch and a 2D view of a 3D model. (b) The silhouette images. (c) The extracted skeletons. (d) The generated skeleton graphs.

We then convert the extracted skeleton into a graph $G = \{V, E\}$ for storing in the database, where $V$ represent nodes and $E$ represent edge in the skeleton graph. There are two kinds of nodes in the skeleton graph: end nodes and junction nodes. A skeleton point which has only one skeleton point in its eight neighbors is defined as an ending node and a junction node is referred as a skeleton point having at least three points in its eight neighbors. Other series of skeleton points between end nodes or junction nodes can be regarded as the connecting edges in the skeleton graph. The end nodes and junction nodes are the nodes that determine the topology of the object. Figure 4(d) shows an example of the generated skeleton graphs for a query sketch and a 2D view of a 3D model respectively by using this method, and the red points denote the junction nodes and the blue points denote the end nodes.

### 3.3.2. Contour Feature Extraction

Contour feature is used to describe the detailed information about the query sketch and provide a more discriminative capability to refine related models in the database. In this paper, we make use of IDSC (inner distance shape context) (Ling and Jacobs, 2007) method to encode contour feature of the query sketch and 2D views of 3D models. Since input sketch and 2D views of 3D model are just made up of edge lines, IDSC utilizes the inner-distance and inner-angle to capture spatial relationships of the edge lines which is good at representing complex internal structures of the sketch and 2D views.

We uniformly sample $n$ points on each 2D view of a 3D model and the input sketch randomly. Note that our sampling strategy is different from the original IDSC method. The sample points $\{p_1, \ldots, p_j, \ldots, p_n\}$ are not all on the external boundary, each sample point $p_i$ can locate on any edge lines of the given object. Our purpose is to encode the contour feature from both internal and external contours of an object. For each sample point $p_i$, the contour feature of $p_i$ is defined as a histogram $h_i$ of the relative coordinates of the remaining $n - 1$ points:

$$h_i(k) = \# \{p_j, j \neq i : (p_j - p_i) \in \text{bin}(k)\} \quad (1)$$

We firstly calculate the relative inner-distance and inner-angle between these sample points, Inner-distance is defined as the length of the shortest path between a pair of sample points and this path should be contained within the given object. Inner-angle is defined as the tangential direction at the starting point of the shortest path connecting these two points. Then, we use log-polar space to construct bins for histogram $h_i$. In our experiment, we set the number of sample points $n = 200$, the relative inner-distance is uniformly partition into 5 intervals for the log distance log $r$ and the relative inner-angle is uniformly partition into 12 intervals for the polar angle $\theta$. Figure 5 shows an example of computing contour feature for a query sketch and a 2D view of a 3D model, both of them have two marked point $p_1$ and $p_2$ where $p_1$ is on the external contour and $p_2$ is on the internal contour. The second row in the figure is the contour feature at $p_1$, the third row is the contour feature at $p_2$. In the histograms, the x axis denotes the orientation bins and the y axis denotes log distance bins. We can find that the sample points on these two different shapes have similar contour feature.
3.4. Similarity Measuring

To efficiently retrieve 3D models from a large database, we propose a hierarchical coarse-to-fine searching strategy which is composed of two stages of similarity measurement. In the first stage, a coarse similarity measuring between query sketch and 2D views of 3D models are evaluated by matching their skeleton graphs. Therefore, the 3D models with dissimilar skeleton graphs can be quickly filtered out. In the second stage, a finer similarity measuring between query sketch and 2D views of the remaining 3D models are evaluated by matching their contour features. Then a set of relevant 3D models can be selected as initial retrieved results for the next rank improving module. This searching strategy adopted in our algorithm not only reduces the number of necessary comparisons between the query sketch and 3D models, but also enhances the effectiveness of feature descriptors.

3.4.1. Skeleton Graphs Similarity Measuring

For a skeleton graph, the junction nodes are salient features as they contain the topological information for connecting subparts of an object. Most existing example-based 3D model retrieval methods using 3D curve skeletons as features measure the similarity of skeleton graphs by mainly matching the properties of the junction nodes. However, a serious issue is that junction nodes is instable which may cause two visually very similar objects have structurally different skeleton graphs. A more effective way of matching skeleton graph is to measure similarity of skeleton paths between end nodes. To match a pair of skeleton paths in two different skeleton graphs, Bai and Lateck(2008) firstly sampled a number of equidistant skeleton points along the skeleton path, and then computed the distance between a pair of skeleton path by matching the sampled skeleton points one-by-one at the same index order. Unfortunately, the fact is that a pair of sample points at the same index may not be corresponding skeleton points due to the difference between two skeleton paths. Hence, it is essential to obtain a correct correspondence between sample points instead of matching them at the same index order.

In fact, the skeleton paths similarity measuring can be regarded as curve matching problem, and we adopt the continuous dynamic time warping measure (CDTW)(Munich and Perona, 1999) to estimate the distance between two skeleton paths. The main reason we adopt CDTW is because that this method can establish the accurate correspondence between sample points on two concerned curves. Moreover, compared with traditional DTW method, CDTW can establish much denser pointwise correspondence between two input curves in a continuous formulation. We denote $S_q$ and $S_e$ as two skeleton graphs to be matched, where $S_q$ represents the skeleton graph of the query sketch and $S_e$ represents the skeleton graph of a 2D view of a 3D model. Assume submitted that $Q_1 = \{q_1(t), t = 1, \ldots, T_1\}$ and $V_1 = \{v_1(t), t = 1, \ldots, T_2\}$ are two skeleton paths in the skeleton graphs $S_q$ and $S_e$ respectively. Let $c = [\alpha(t), \beta(t)]^T$ be a correspondence map between $Q_1$ and $V_1$ such that a sample point $q_1(\alpha(t)) \in Q_1$ corresponds to a sample point $v_1(\beta(t)) \in V_1$, for $t \in \{1, \ldots, \max(T_1, T_2)\}$. Then, the distance between the two skeleton paths $Q_1$ and $V_1$ can be estimated as:

$$pd(Q_1, V_1) = \sum_{t=2}^{T} ||q_1(\alpha(t))v_1(\beta(t)) - q_1(\alpha(t-1))v_1(\beta(t-1))||^2$$

where $\| \cdot \|$ is the $l_2$ norm. According to the definition of path distance, we can solve this problem by finding an optimal correspondence map that can minimizes the skeleton path distance $pd(Q_1, V_1)$:

$$\tilde{c} = \arg \min_c \ pd(Q_1, V_1) \quad (3)$$

The optimal correspondence map $\tilde{c}$ can be found with the shortest path on a warping plane which is defined as a grid by the sample points on the skeleton paths $Q_1$ and $V_1$. More precisely, if the shortest path passes the vertex $(i, j)$ on the warping plane.
plane, a pair of sample points \( q_i(i) \in Q_1 \) and \( v_j(j) \in V_1 \) can be regarded as corresponding points. In addition, CDTW method can obtain a translation invariant path distance for any pair of skeleton paths. But we also need to guarantee the path distance is invariant to rotation and scale. As suggested in Li et al.(2014), we align two skeleton graphs \( Q_1 \) and \( V_1 \) by transforming their bounding boxes positions and orientations to achieve rotation invariance. For the scale invariance, we normalize their corresponding path lengths. Figure.6 provides the examples of the correspondence results of the skeleton paths by utilizing CDTW method.

After we compute the distance for each pair of skeleton paths on the skeleton graphs \( S_q \) and \( S_v \), their dissimilarity value can be defined as:

\[
D_{3d}(S_q, S_v) = \sum_i p_d(q_i, v_{n(i)})
\]

where \( q_i \) and \( v_{n(i)} \) denote the best matching skeleton paths from the skeleton graphs \( S_q \) and \( S_v \) respectively, which can be obtained by using Hungarian method. Then, we can eliminate a large amount of irrelevant 3D models according to the dissimilarity of the skeleton graphs by using Eq.(4). In general, we only retain a tenth of 3D models from the database in our experiment.

\[\text{Fig. 6 The example results of computing correspondence between a pair of skeleton paths. The skeleton paths are marked in red and the corresponding sample points along the skeleton paths are linked with lines.}\]

### 3.4.2. Contour Feature Similarity Measuring

This stage is to evaluate similarity of contour feature between query sketch and the remaining 3D models in the database to select an initial retrieval result. As mentioned above, contour feature is represented as IDSC histograms in our retrieval system. Since our sample points can locate on the any edge lines inside of the object, dynamic programming matching method adopted in Ling and Jacobs(2007) can not be applied in our retrieval system directly. Inspired by Eitz et al.(2012), we convert contour feature into a vocabulary of visual words and use a bag-of-words method to compute dissimilarity value between the query sketch and the 2D view of a 3D model.

We assume \( C_q \) and \( C_v \) as the extracted contour features from the query sketch and a 2D view of a 3D model respectively. Let \( h_i \) and \( h_j \) denote two different IDSC histograms, we first define a dissimilarity measure between these two histograms as:

\[
h_d(h_i, h_j) = \frac{K}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}
\]

where \( h_i(k) \) and \( h_j(k) \) represent the k-th bin of the IDSC histograms \( h_i \) and \( h_j \) respectively, and \( K \) is the number of histogram bins. Based on the dissimilarity metric Eq.(5), we can cluster all of the IDSC histograms into visual words to create a visual vocabulary and each cluster represents a visual word in the vocabulary. Eitz et al.(2012) and Wang et al.(2014) used the K-means clustering algorithm to generate visual vocabulary where the cluster number \( N \) need to be predefined. Hence, they have to try different cluster number to determine an optimal value. Instead, we use the X-means clustering algorithm(Pelleg and Moore, 2000) to cluster the IDSC histograms into visual words, in which the optimal cluster number can be determined automatically.

Suppose that a visual vocabulary \( W = \{w_1, \ldots, w_{i-1}, \ldots, w_m\} \) with the number of \( m \) visual words has been generated, we can quantize the IDSC histogram from each 2D view of the remaining 3D models by using this visual vocabulary. If an IDSC histogram falls into the cluster of word \( w_i \), then it can be represented as the index of this visual word. Thus, the contour feature of each 2D view can be represented by a \( m \)-vector: \( C_v = \{(w_{i1}, t_{i1}), \ldots, (w_{im}, t_{im})\} \), where \( t_i \) denotes the importance of a visual word \( W_i \) in the visual vocabulary \( V \). The weight \( t_i \) can be computed by using the
standard weighting scheme tf-idf(term frequency-inverse document frequency), which is defined as:

\[
t_i = \frac{n_i}{n} \log \frac{N}{n_i}
\]

where \(n_i\) is the number of word \(w_i\) occurring in a 2D view, \(n\) is the total number of words in this view, \(N\) the total number of 2D views in the remaining 3D models, \(n_i\) is the number of word \(w_i\) occurring in all of the 2D views in the remaining 3D models. By performing the same quantization step on the query sketch, the contour feature of the query sketch can also be represented as: \(C_q = \{(w_i, t_i^1), \ldots, (w_i, t_i^n), \ldots, (w_m, t_m^n)\}\). Thus, the dissimilarity of contour feature between a query sketch and a 2D view of a 3D model can be defined as:

\[
D_{\text{con}}(C_q, C_v) = 1 - \frac{\sum_{i=1}^n t_i \times t_i^j}{\sqrt{\sum_{i=1}^n (t_i)^2} \sqrt{\sum_{i=1}^n (t_i^j)^2}}
\]

According to contour feature dissimilarity, we can rank the remaining 3D models from the database in an increasing order and select the top ranked 50 models as an initial retrieval result.

3.5. Rank Improving

For an ideal sketch-based 3D model retrieval system, the top-ranking retrieved 3D models and the sketching object should belong to the same class. Since most existing retrieval approaches consider only measuring the similarity between the query sketch and the 3D models from the database, some particular 3D models which do not belong to the same class of the query sketch may rank higher than the others belong to the same category as the query sketch. Due to the hand-drawn sketch is ambiguous and incorrect, it often shows little resemblance to the corresponding models. To tackle this problem, we exploit relationships among the initial retrieved 3D models to improve the ranking order based on the graph-transduction method (Bai et al., 2010). The basic idea is to utilize the similarity between 3D models in the initial retrieved set as context for providing better similarity measure between the query sketch and 3D models.

Assume that \(v_1\) is a query sketch, \(v_1, v_2, \ldots, v_n\) is a collection of 2D views of 3D models in the initial rank list. Firstly, we define two affinity matrices \(W_1 = (w_{i,j}^1)_{n \times n}\) and \(W_2 = (w_{i,j}^2)_{n \times n}\) by using dissimilarity measure of skeleton graph \(D_{ij}(v_i, v_j)\) and contour feature \(D_{\text{con}}(v_i, v_j)\):

\[
w_{i,j}^1 = \exp\left(\frac{(D_{\text{sk}}(v_i, v_j))^2}{\sigma_1^2}\right) \quad w_{i,j}^2 = \exp\left(\frac{(D_{\text{con}}(v_i, v_j))^2}{\sigma_2^2}\right)
\]

where \(i, j = \{1, \ldots, n\}\), \(\sigma_1\) and \(\sigma_2\) can be chosen as the maximal dissimilarity value of skeleton graph and contour feature in the collection. Then, we build two fully connected graphs \(G_1 = (N_1, E_1)\) and \(G_2 = (N_2, E_2)\), which share the same set of nodes \(N_1 = N_2 = \{v_1, v_2, \ldots, v_n\}\) and edges \(E_1 = E_2 = \{e(i, j)|e(i, j) = (v_i, v_j), i \neq j\}\), \(w_{i,j}^1\) and \(w_{i,j}^2\) are the edge weight of the edge \(e(i, j)\) in the graph \(G_1\) and \(G_2\) respectively. So a Markov chain can be constructed on these two graphs \(G_1\) and \(G_2\) with the probabilistic transition matrices \(P_1 = (p_{i,j}^1)_{n \times n}\) and \(P_2 = (p_{i,j}^2)_{n \times n}\):

\[
p_{i,j}^1 = \frac{w_{i,j}^1}{\sum_{k=1}^n w_{i,k}^1} \quad p_{i,j}^2 = \frac{w_{i,j}^2}{\sum_{k=1}^n w_{i,k}^2}
\]

where \(p_{i,j}^1\) and \(p_{i,j}^2\) represents the probability of transition in one time step from node \(v_i\) to node \(v_j\) in the graph \(G_1\) and \(G_2\) respectively. We denote \(S_1^t(i)\) and \(S_2^t(i)\) as the new similarity measures between the query sketch \(v_1\) and a 2D view \(v_i\) at the \(t\)-th iteration step during transition, which can be computed based on the probabilistic transition matrices \(P_1 = (p_{i,j}^1)_{n \times n}\) and \(P_2 = (p_{i,j}^2)_{n \times n}\):

\[
S_1^t(i) = \sum_{j=1}^n p_{i,j}^1 S_1^{t-1}(j) \quad S_2^t(i) = \sum_{j=1}^n p_{i,j}^2 S_2^{t-1}(j)
\]

when \(t = 1\), \(S_1^1(i) = w_{i,j}^1\) and \(S_2^1(i) = w_{i,j}^2\). Assume that the number of iterations \(t = T\), then \(S_1^T(i)\) and \(S_2^T(i)\) can be represented as the final improved similarity measures of skeleton graph and contour feature between the query sketch \(v_1\) and a 2D view \(v_i\). We combine the improved similarity measures of skeleton graph and contour feature by using linear interpolation as an integrated similarity measures:

\[
sim(v_1, v_i) = \lambda \times S_1^T(i) + (1 - \lambda) \times S_2^T(i)
\]

where \(\lambda\) is a weight factor. In our experiment, we set the iteration time \(T = 5\) and use purity selection method to determine the weight factor \(\lambda\). We find that the retrieval system can achieve relatively optimal performance when \(\lambda = 0.3\). Then, we can re-rank the 3D models in the initial retrieval list according to this integrated similarity measure.
4. Experiment

In this section, we will describe the experimental environment setup, the data set used for evaluation purpose and comparative experiments of the proposed algorithm with other retrieval algorithms.

4.1. Experimental setup

Fig. 7 The user interface of our retrieval system.

All experiments are performed on a PC with Windows 8.1/64bits, Intel Core 2 Duo i3-4130 CPU@3.40GHz, 8.0 GB memory. Additionally, the PC is installed a Wacom CTH-670 touch drawing tablet as a sketch pad for the users. The retrieval interface of our system is given as Fig.7. The left side of the interface are three canvases for sketching the model. Our retrieval system can support a user submit at most three freeform sketches as seen from three different viewing directions to describe a target 3D model. The user also can erase and modify if they do not satisfy with the drawn sketch. Moreover, we provide a convenient way that user can directly save and upload a finished hand-drawn sketch to the canvas. The right side is displaying page for the retrieved 3D models with a relevant JPEG image which is a thumbnail view of the searched model. The user can click the image to download the corresponding 3D model.

To test the retrieval effectiveness of our approach, we have chosen Princeton Shape Benchmark (PSB)(Shilane, et al., 2004) as the test 3D model database, which is the most well-known and frequently used 3D model retrieval benchmark. This benchmark contains 1814 models and is split into a train set and a test set. For the input query sketch, we use two standard sketch benchmarks to test our retrieval system: the Sketch-PSB(Eitz, et al., 2012) and SHREC 2013 Large Scale Sketch-Based 3D Shape Retrieval(Li, et al., 2013). The Sketch-PSB contains 1814 sketches, each sketch is associated with one category from the PSB benchmark. SHREC 2013 contains 7200 sketches, which are partitioned into 90 classes and 80 sketches per each class. We employ these two sketch dataset as query inputs for the comparison with other leading sketch-based 3D model retrieval algorithms.

4.2. Experiment Results and Comparisons

Figure.8 illustrates some examples of sketch-based retrieval results using our approach. The left side of the figure is the query sketch, and the right side shows the top 16 retrieved 3D models which are ordered according to the similarity to the query sketch. The average response time per query for our retrieval time is 1.62s.

From the experiment results, we can find that most of our retrieval results can successfully return relevant 3D models of the query sketch. Because we adopt the graph transduction method to improve the rank order, the retrieved 3D models in the high ranking position and the query sketch belong to the same class. Particularly, we can still gain a satisfactory retrieval result even when the query sketch contains complex internal structures, such as the car sketch(Fig.8(b)) and fish sketch(Fig.8(c)). Nevertheless, there still exist some failure cases, for instance a sword, a hammer and a round table are retrieved in the human sketch(Fig.8(e)), a shoe and a bottle are retrieved in the fish sketch(Fig.8(c)) and a lamp post and a satellite dish are retrieved in the table lamp sketch(Fig.8(f)). The main reason is that the skeleton graphs of these negative
Fig. 8 Examples of sketch-based 3D model retrieval results using our retrieval algorithm. The left side is the query sketch, the right side is the top 16 retrieved 3D models.

retrieval results are highly similar to the query sketch, which is difficult to eliminate them during the filtering stage.

To further verify our combined approach of skeleton graph and contour feature, we separately applied skeleton graph and contour feature to retrieve 3D models. Moreover, we also combined skeleton graph with contour feature but without adding rank improving module to test the retrieval performance. We compared the performance of them with the whole proposed composite approach. Additionally, we compared our approach with other four leading sketch-based 3D model retrieval algorithms, including Spherical Harmonics (Funkhouser, et al., 2003), Light Field (Chen, et al., 2003), STELa (Saavedra, et al., 2011) and GALIF (Eitz, et al., 2012). To have a comprehensive evaluation on these approaches, we employed seven evaluation metrics which have been commonly used in information retrieval (Shilane, et al., 2004), including Precision-Recall plot (PR), Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-Measures (E), Discounted Cumulated Gain (DCG) and Average Precision (AP). The comparison result is shown in Fig. 9.

In the figure, SPH refers to spherical harmonics method, LF refers to light field method, SK refers to only applying our skeleton graph method for retrieving models, CON refers to only applying our extracted contour feature for retrieving models, SK+CON indicates that we combine skeleton graph and contour feature these two feature descriptors but without adding graph-transduction method for ranking improving, SK+CON+GD indicates that we integrate skeleton graph, con-
tour feature and graph-transduction method these three components together for retrieving models. From the experiment result, we can find that our integrated method (SK+CON+GD) outperforms other leading sketch-based 3D model retrieval approaches. The combined skeleton graph and contour feature method (SK+CON) takes the second place and the GALIF method takes the third place. The contour feature method (CON) performs worse than the GALIF method which takes the fourth place. The performance of skeleton graph method (SK) is worse than the STELA method and better than the spherical harmonics and light field method which takes the sixth place. The spherical harmonics and light field method only exploit boundary information which cannot handle the input sketch with internal structures. The STELA and GALIF method can capture locality information of the query sketch, but their methods are highly sensitive to the stylistic variation of the sketches because of their proposed feature descriptors are lack of global structure information of the sketch and only computing the similarity between the query sketch and 3D models. In contrast, our feature descriptors combining skeleton graph and contour feature can provide a more differentiate capability. Furthermore, our rank improving method can guarantee most of the top ranked retrieved 3D models belong to the same class of the query sketch.

5. Conclusion

In this paper, we proposed a novel sketch-based 3D model retrieval method, which integrates skeleton graph and contour feature from query sketch and 2D views of 3D models. We first preprocess each 3D model in the database by performing normalization and selecting the most representative viewpoints. Then, a hierarchical coarse-to-fine searching strategy is adopted to retrieve 3D models. Skeleton graph is extracted to describe structural information of query sketch as well as the 2D views of 3D models, which is used to eliminate a large amount of dissimilar 3D models. Contour feature is extracted to capture detailed information of the sketch, which is used to select initial searched results. Finally, the graph transduction method is exploited to re-rank the retrieval results. The experiment results demonstrate that our method can
efficiently retrieve relevant 3D models and achieve more accurate retrieval results than other previously proposed methods.

However, there still exist some limitations in our method. First, the average retrieval time of our approach is greater than other methods because of integrating skeleton graph and contour feature and adding a rank improving module in our retrieval system. Second, if the skeleton graph of an irrelevant 3D model from the database is highly similar to the query sketch, it is difficult to exclude this model in the filtering stage. Despite these drawbacks, our approach can still achieve better retrieval accuracy than other approaches.

The future of our work is to develop a user interaction feedback mechanism. After the users submit the query sketch, the system first provide a list of retrieved 3D models. Then, the users can refine the retrieval results by selecting which 3D models they think are the good results. This feedback mechanism can not only provide more desirable retrieved 3D models to the user, but also enhance user interaction just by making some easy choices. It will largely improve the effectiveness of our retrieval system.

Acknowledgement

This research is supported by Natural Science Foundation of China (NSFC) (No. 61320106008), NSFC-Guangdong Joint Fund (No. U1201252, U1135005), Guifeng Zheng is the corresponding author.

References


