Vis Comput (2015) 31:765–774
DOI 10.1007/s00371-015-1091-1

ORIGINAL ARTICLE

Learning best views of 3D shapes from sketch contour

Long Zhao[1] · **Shuang Liang**[1] · **Jinyuan Jia**[1] · **Yichen Wei**[2]

Published online: 28 April 2015

Abstract In this paper, we introduce a novel learning-based approach to automatically select the best views of 3D shapes using a new prior. We think that a viewpoint of the 3D shape is reasonable if a human usually draws the shape from it. Hand-drawn sketches collected from relevant datasets are used to model this concept. We reveal the connection between sketches and viewpoints by taking context information of their contours into account. Furthermore, a learning framework is proposed to generalize this connection which aims to learn an automatic best view selector for different kinds of 3D shapes. Experiments on the Princeton Shape Benchmark dataset are conducted to demonstrate the superiority of our approach. The results show that compared with other state-of-the-art methods, our approach is not only robust but also efficient when applied to shape retrieval tasks.

Keywords Best view selection · Sketch-based modeling · Context similarity · Bag-of-features

1 Introduction

The recent years witnessed tremendous advances in the technologies of 3D graphics, and they are widely accessible in our daily life. Especially, the 3D shape as the basic element of 3D graphics always plays a vital role. This has resulted in the demand for various accurate 3D shape modeling and analyzing methods for real-world applications. Automatically selecting the best views for a given 3D shape is one of the most important preprocessing tasks in 3D graphics. It has been applied in many 3D graphics applications, including virtual reality, shape retrieval [4,9,25], computer-aided design (CAD) [18], 3D multimedia [32], and so on. The problem of best view selection is to seek a few viewpoints that follow human visual preference.

Many research works have been conducted to solve this problem. Previous works, such as mesh saliency [13] and viewpoint entropy [29], focus on discovering the relationship between geometric characteristics (e.g. structure of mesh strips or vertexes) and human visual perception. Their goal is to answer the question: Which part of a 3D shape catches human interest? However, this is very difficult since accurate shape analyzing has already been a challenging task. Recently, Liu et al. [17] gives a new insight into this field. Rather than answering the above question, it makes use of things which carry the information of human visual preference to estimate the viewpoint of 3D shapes. The web image, a medium that contains view information about how people choose their favorite views in photographing, is employed and has achieved good performance.

Other than photographing, painting is another prior that reflects human visual preference. For example, a sketch can depict the painter's favorite view of the object. Moreover, sketches depicting the same object but drawn by different painters can show different preferences. Professional painters always prefer perspective drawing, while amateurs choose the easiest angle (front or side) to avoid a poor drawing. This kind of diversity makes it possible for us to discover all candidate best views for a certain object.

The main contribution of this paper can be summarized as follows: Firstly, we reveal an another interesting prior for the best view selection problem of 3D shapes. We think a viewpoint of the 3D shape is good if people usually draw the shape from it. In addition, hand-drawn sketches are used

✉ Shuang Liang
shuangliang@tongji.edu.cn

[1] Tongji University, Shanghai, China

[2] Microsoft Research, Beijing, China

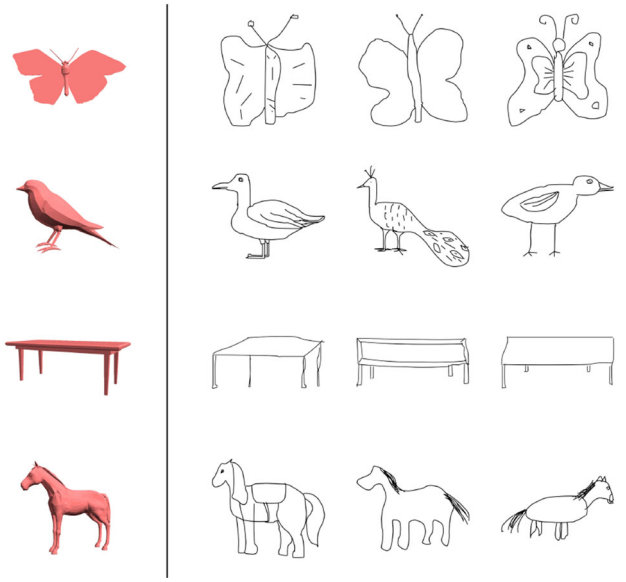

Fig. 1 Sketches reflect favorite views when people draw objects. *Left* different 3D shapes. *Right* relevant sketches of the shapes

to model this concept. Thanks to the rapid development of sketch-based shape retrieval technology, lots of large-scale 3D shape datasets providing relevant sketches are available from different sources [4,14,15]. As illustrated in Fig. 1, these datasets make it possible for us to model our concept, for the meaningful relationship between sketches and 3D shapes is established by them.

Secondly, compared with images, sketches are line drawings full of deformation and noises, while color and texture messages are lost. Therefore, mapping sketches to accurate viewpoints of 3D shapes is challenging. To deal with this condition, we make use of context information of contours from input sketches. A contour is a set of edges forming lines or curves. Furthermore, the context of a contour is regarded as how a contour and its surrounding ones are connected to each other, which can be used to represent a meaningful part of an object (e.g. silhouettes of a foot or a tail). We notice that context information of sketch contours always encodes rich information and is utilized to measure the similarity. In practice, this measurement achieves stable performance and can reduce the influence of deformation and noises from sketches.

Thirdly, contour context also carries common features which reflect human preference of drawing an object. For example, we always draw feet of horses or desks at the bottom of a sketch, while we draw tails of animals on the horizontal sides. Motivated by this observation, we propose a learning framework to learn a generic best view classifier via context features and use it to select the best views for different kinds of 3D shapes. Experiments show that our approach is very efficient compared to other state-of-the-art methods, especially when applied in shape retrieval tasks.

2 Related work

The web image-driven approach proposed by [17] is one which is most related to our work. It directly explores human perception on observing 3D shapes from the relevant web images. Area similarity, silhouette similarity and saliency disparity are utilized to compute the corresponding view from an image. Final views are then judged by voting on all input images.

Our method mainly differs from [17] in the following two aspects. Firstly, we explore human preference on observing 3D shapes by estimating where human tends to draw it. Hand-drawn sketches other than photos are used to learn this bias. And we also propose a different similarity measurement to map sketches to viewpoints, which can handle deformation of sketches and produces stable results. Secondly, selecting views by images disables [17] to best view computation when the class of an input 3D shape is unknown. Instead, we generalize the specific relation between sketches and viewpoints in the dataset by learning a generic classifier, which can compute best views for 3D shapes without precise classification.

Other works compute best views directly from the geometric features of 3D shapes, such as mesh strips or vertexes. Saliency of a 3D shape is firstly addressed in [13] and the best view is selected as the one observing the largest amount of mesh saliency among a set of sampled views. Recently, a new saliency measurement is introduced in [27], which is efficient for large point sets. Viewpoint entropy [29] is another important geometric model to help to select best viewpoints of a 3D shape. It employs the projected area of all the visible triangles as entropy to measure the best view with maximum relative projective area. Moreover, Page et al. [23] improves this method by using information theory. However, this kind of methods usually generates unreasonable results when the 3D shape is complex, since the connection between geometric structure of 3D shapes and human perception cannot be easily modeled.

Learning guided methods are also introduced in recent works. Laga and Nakajima [12] uses boosting to learn best views of 3D shapes based on the assumption that models belonging to the same class of shapes share the same salient features. Laga [10] presents another data-driven approach. It formulates the best view selection problem as a feature selection and classification task. This approach is robust to intra-class variations and is consistent within the models of the same class of shapes, but its performance highly depends on the training datasets. Eitz et al. [4] just uses the silhouette length, projected area and smoothness of depth distribution over the shape as the features to learn a perceptual classifier. Although it shows capable results for simple shapes, it fails easily when the shape is complex.

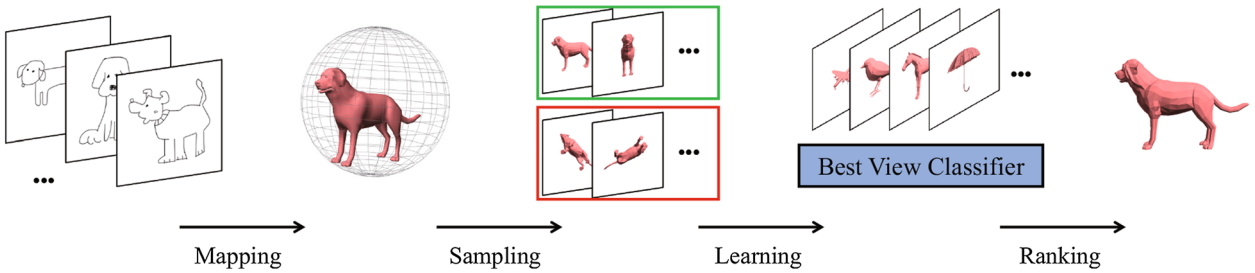

Fig. 2 Overview of our algorithm. The four stages of our algorithm: mapping sketches to the viewpoints of corresponding shapes, sampling training data, learning a best view classifier and ranking final viewpoints

3 Approach overview

The main idea of our algorithm is to learn perceptual view information from relevant hand-drawn sketches. Formally, for a 3D shape m_i, a series of corresponding sketches drawing m_i are provided. We aim to learn a best view classifier using context information of contours from these data. From the uniform view space $\mathcal{V}_i$ around m_i, a set of best views is selected by this classifier as the ones reflecting where human is possibly to draw the shape. The work flow of our algorithm is shown in Fig. 2, which includes four stages: mapping sketches to the viewpoints of corresponding shapes, sampling training data, learning a best view classifier and ranking final viewpoints.

The rest of our paper is organized as follows. Section 4 describes an efficient similarity measurement between sketches and viewpoints of the relevant 3D shape. We use this measurement to map sketches to viewpoints for further sampling the training data and learning the best view classifier. Section 5.1 describes how we map sketches to suitable viewpoints and how we sample positive and negative examples. Then our learning algorithm of view selection is introduced in Sect. 5.2. In Sect. 6, a greedy ranking algorithm is proposed to rank the classification results which takes view diversity into account. At last, all experimental results are shown in Sect. 7.

4 Similarity measure

Before revealing the connection between a hand-drawn sketch and a viewpoint of the 3D shape, we need to convert the projection of the viewpoint into a sketch-like view map at first for further comparison. Recent works [18,30] present a hybrid line rendering method to generate 2D views for the 3D shape and obtain good performance. In this paper, we adopt this method and combine exterior silhouettes, occluding contours, suggestive contours [3] and shape boundaries to generate the final view map. An example of line rendering view map is shown in Fig. 3b.

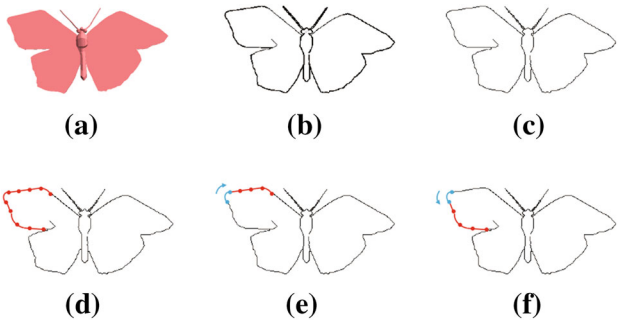

Fig. 3 An illustration of contour grouping. **a** An input 3D shape; **b** the line rendering view map generated by hybrid line rendering method; **c** the result after edge thinning; **d** the final result of contour grouping (to avoid clutter, we only show contour groups on the *upper left* of the image); **e**, **f** two differnt walks (shown in *red*) starting at the same contour group (shown in *blue*)

4.1 Contour grouping

A contour is defined as a set of edges forming a coherent boundary, curve or line [33]. Our goal is to use context information of contours to measure the similarity between two contour maps.[1] Therefore, all pixels in a contour map should be grouped into meaningful contour groups in advance. Here we present a simple contour grouping strategy.

Given a contour map, we initially perform the edge thinning [31] operation to it. After that, we compute the gradient orientation for each pixel on lines using the Sobel operator [7]. The result is a very sparse line drawing map, and each line is one pixel width with each pixel p having an edge orientation θ_p. This preprocessing operation can reduce most of the noise on the initial map, while simultaneously being very efficient.

Then pixel seeds are uniformly sampled from all lines in the contour map. We form the initial contour groups using a simple greedy approach that combines 8-connected contours from each seed until the sum of their orientation differences is above a threshold ($\pi/2$). Then for an initial contour group

[1] In this paper, we use "contour map" to refer either to an input sketch or to a view map generated from the viewpoint of 3D shapes.

g_i, we denote its mean position as x_i and its mean orientation as θ_i. According to [33], the affinity between two groups is defined as

$$a(g_i, g_j) = |\cos(\theta_i - \theta_{ij}) \cdot \cos(\theta_j - \theta_{ij})|^2 \quad (1)$$

where θ_{ij} is the angle between x_i and x_j. We use Eq. 1 to greedily merge contour groups until the maximum value of each pair of $a(g_i, g_j)$ is less than a certain threshold (0.8). Intuitively, it can further merge two groups if the angle between the groups' means is similar to the groups' orientations. We get the final grouping results as illustrated in Fig. 3d.

This contour grouping method is computationally trivial. In practice, we find that the results are robust when applied in contour context computation.

4.2 Context-based measure

Using context messages to measure the similarity of different parts of objects is proven very efficient in many 3D shape analyzing tasks [8,11,21]. Context of a contour is regraded as how the contour and its surrounding ones are connected to each other, which provides rich information. In this paper, we think two contours are similar if their appearance and context are similar. The graph model presented by [11] is used to formulate our approach.

We begin to model this method by defining the appearance similarity of two contour groups g_i and g_j

$$d_{app}(g_i, g_j) = \exp\left(-\frac{d_{spa}(g_i, g_j)^2}{2\sigma_{spa}^2}\right) \cdot \cos(\theta_i, \theta_j) \quad (2)$$

where θ_x is the orientation of g_x, and $d_{spa}(g_i, g_j)$ is the Euclidean distance between their normalized mean positions in the map (with $\sigma_{spa} = 0.2$). Intuitively, two contour groups are similar if they are in the same position and with same orientation.

In order to take context messages into account, we construct a dual graph $G = (V, E)$ for each contour map. Each node in V represents a contour group. Two nodes are connected with an edge $e \in E$ if their contour groups are spatially adjacent in the contour map. Let W_x^n denote a walk of length n starting at the node (i.e. contour group) g_x, and we define the similarity of two walks as

$$d_{walk}^n\left(W_i^n, W_j^n\right) = \frac{1}{n+1}\sum_{k=1}^{n+1} d_{app}\left(w_i^k, w_j^k\right) \quad (3)$$

where w_x^k is the k-th node on the walk of W_x^n. A walk W_x^n is an ordered node sequence from node g_x which can capture the local contextual structure of g_x. As a result, we can measure the context similarity of two nodes g_i and g_j by compare all the walks of them and retain the best match respectively. This process can be formulated as

$$d_{con}^n(g_i, g_j) = \frac{1}{|P_i^n|}\sum_{\{a|a\in P_i^n\}} \max_{\{b|b\in P_j^n\}} d_{walk}^n(a, b) \quad (4)$$

where P_x^n is a collect of all walks of length n starting from node g_x and $|P_x^n|$ is the number of walks in P_x^n. For P_x^n is able to capture the context information of g_x, we call it the context descriptor of g_x. Note that when $n = 0$, Eq. 4 reduces to the appearance similarity given in Eq. 2.

Given the similarity measurement defined by Eq. 4, computing part-wise correspondences between two contour maps becomes straightforward. Each contour map is represented with its structural graph. For each contour group on a contour map, we compute its similarity to the other parts on the target map using the measurement defined in Eq. 4 and obtain the best match. Therefore, the context similarity of two contour maps c_i and c_j is

$$S_{con}^n(c_i, c_j) = \frac{1}{|c_i|}\sum_{\{g_i^x|g_i^x\in c_i\}} \max_{\{g_j^y|g_j^y\in c_j\}} d_{con}^n\left(g_i^x, g_j^y\right) \quad (5)$$

where g_i^x is the contour group in c_i and $|c_i|$ is the number of groups in c_i. Obviously, when $n = 0$, Eq. 5 reduces to an equation that only takes appearance into account:

$$S_{app}(c_i, c_j) = \frac{1}{|c_i|}\sum_{\{g_i^x|g_i^x\in c_i\}} \max_{\{g_j^y|g_j^y\in c_j\}} d_{app}\left(g_i^x, g_j^y\right) \quad (6)$$

In Sect. 7.1, we demonstrate that $S_{con}^n(c_i, c_j)$ outperforms $S_{app}(c_i, c_j)$, and taking context information into consideration can evidently improve the performance.

The context similarity given by Eq. 5 requires setting the maximum length n of the walks. Setting n to 0 is equivalent to comparing contour groups by their appearance similarity. Larger values of n capture more structures, while small values of n capture less. Experimentally, we found that values between 3 and 5 provide good and stable results. And we set $n = 4$ in all results shown below.

4.3 Efficient computation via key points

The similarity measurement of contour maps is the core algorithm of our approach which is further applied in the sampling and training stages. It should be simple and fast enough to be continuously computed. However, Eq. 5 requires a large search space for each pair-wise match of contour groups, and thus inefficient. We notice that the most context information in a contour map is carried by the corner of contours, while

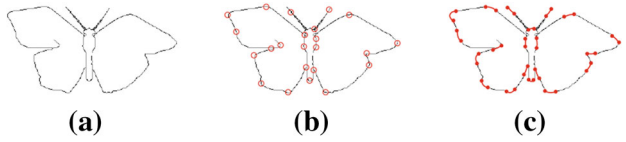

Fig. 4 Illustration of keypoint-based detectors. **a** An input contour map; **b** key points detected by Harris corners [22]; **c** key contour groups (shown in *red*) computed in the contour map

single lines or curves is less important. Therefore, it is unnecessary to conduct all pair-wise matches between two maps. Here we speed up the computation via only perform matches at the key points of contour maps.

Recent approaches [16,20] show that keypoint-based detectors, such as difference of Gaussian (DoG) [19], Hessian operator [1] and Harris-Laplace detector [22], are suitable for line drawing feature extraction. Harris corner detector is employed in our method, for it achieves better performance than other keypoint-based methods [20]. Given a contour map c_x, we compute the Harris key points of it at first. A set of contour groups $\mathcal{K}_x$ are constructed as shown in Fig. 4, which only contains groups having key points on them. Then the key context similarity between two maps can be computed as

$$S_{key}(c_i, c_j) = \frac{1}{|\mathcal{K}_i|} \sum_{\{g_i^x | g_i^x \in \mathcal{K}_i\}} \max_{\{g_j^y | g_j^y \in \mathcal{K}_j\}} d_{con}^4 \left(g_i^x, g_j^y\right) \quad (7)$$

where we fix the length of walks to 4 here. Moreover, we define all P_i^n (Eq. 4) starting from contour groups which are contained in $\mathcal{K}_x$ as the key context descriptors of a given contour map.

5 Learning the best views

5.1 The training data

During the training process, a collection of 3D shapes $\mathcal{M}$ and their corresponding sketches $\mathcal{S}_i$ to each shape $m_i \in \mathcal{M}$ are assumed to be provided. In the experiments, 3D shape collection of the Princeton Shape Benchmark (PSB) [26], and relevant sketch data collected by [4] are used. The PSB defines a split into training and test dataset, which both have 907 shapes with different classification. Then for each shape, a set of sketches belonging to the same category are collected by [4]. Each sketch consists of hand-drawn silhouettes and closed boundary curves, which meets our requirements. In the remainder of this paper, we use the training dataset of the PSB to learn the classifier, while we evaluate our approach on the test dataset.

Like many other learning problems, in order to learn a classifier which helps to score a given viewpoint, positive and negative examples are required. However, the association between the sketch and the viewpoint of a shape is not established on the PSB dataset. Here we present an approach to map a sketch to the shape's viewpoint via the key context descriptor described in Sect. 4.3, and further we use this kind of relation to collect the best and worst viewpoints as training samples.

Firstly, for each shape $m_i \in \mathcal{M}$, we uniformly sample K viewpoints on its bounding sphere and compute the contour map $c_i^k \in \mathcal{C}_i$ for the viewpoint $v_i^k \in \mathcal{V}_i$. Experimentally, we set $K = 300$ to achieve stable performance. Then, for each sketch $s_i^j \in \mathcal{S}_i$ belonging to shape m_i, the similarity measure defined by Eq. 7 is utilized to computed the similarity between every pair of s_i^j and c_i^k. We formulate the possibility that a sketch s_i^j is drawn from the viewpoint v_i^k as: $\forall v_i^m \in \mathcal{V}_i$,

$$p(s_i^j, v_i^k) = \frac{S_{key}\left(s_i^j, c_i^k\right) - \min S_{key}\left(s_i^j, c_i^m\right)}{\max S_{key}\left(s_i^j, c_i^m\right)} \quad (8)$$

where c_i^x indicates the contour map computed from v_i^x. Obviously, we map the sketch s_i^j to the viewpoint v_i^k when $p(s_i^j, v_i^k) = 1$, and we treat all the viewpoints that meet this condition as the positive samples. To collect the negative samples, the average of $p(s_i^j, v_i^k)$ towards all $\mathcal{S}_i$ is computed, then the viewpoint v_i^k with the average possibility lower than a threshold ξ is regarded as a negative sample. This negative sampling strategy has the intuitive interpretation that a viewpoint is worse if people seldom draw the 3D shape from it. Finally, we collect the samples for each shapes in the PSB dataset and gather positive and negative examples according to this strategy respectively.

Given a viewpoint v_i^k of the shape m_i, our above sampling strategy can be summarized as the following discriminant function

$$\Theta(v_i^k) = \begin{cases} 1, & \text{if } \exists s_i^m \in \mathcal{S}_i, \quad p\left(s_i^m, v_i^k\right) = 1; \\ 0, & \text{if } \forall s_i^m \in \mathcal{S}_i, \quad \frac{1}{m}\sum_m p\left(s_i^m, v_i^k\right) < \xi; \\ null, & \text{otherwise} \end{cases} \quad (9)$$

where $\Theta(v_i^k) = 1$ indicates that v_i^k is treated as a positive sample, while 0 is a negative one. We set ξ to 0.05 experimentally. Note that all the sampling process is performed in advance to save computational efficiency.

5.2 Learning algorithm

The feature vector of each viewpoint of the 3D shape is built upon a bag-of-features (BoF) model, which has been widely used to extract visual features in various computer vision tasks [28]. The basic idea of this approach is to compare

the difference among viewpoints based on a histogram of features.

In the training dataset, we randomly sample one million key context descriptors presented in Sect. 4.3 from the contour maps of both positive and negative samples, in order to cover a wide variety of possible descriptors. Then the contextual vocabulary is generated via a fast version of k-medoids clustering algorithm [24]. The set of resulting cluster centroids $\mathcal{W} = \{\mathbf{w}_i\}$ forms the contextual vocabulary where each entry $\mathbf{w}_i$ (contextual word) represents the contextual features in the ith cluster. We represent each viewpoints as the histogram of contextual word frequency from the relevant contour map. Since the size of the contextual vocabulary $|\mathcal{W}|$ is an important parameter that strongly influences performance, we determine its value ($|\mathcal{W}| = 800$) according to the optimization framework proposed by [4].

Let $\mathbf{h}_i^k$ denote the feature vector of the viewpoint v_i^k. We aim to learn a scoring function $Score(\mathbf{h}_i^k) \in [0, 1]$ to predict the possibility that human tends to draw from the viewpoint. This supervised learning problem can be easily solved by the Support Vector Machines (SVM), which have been successfully applied to classification tasks in computer vision, computer graphics, and geometry processing due to its stable performance. The scoring function is defined as:

$$Score(\mathbf{h}_i^k) = \mathbf{t} \cdot \mathbf{h}_i^k - b \tag{10}$$

where $\mathbf{t}$ and b are learned coefficient and bias terms. Since most of the feature vectors are sparse, we use the LIBLINEAR library provided by [6] to train the classifier because of its fast computation speed. Note that in order to balance the number of positive and negative examples, we equally sample five thousand viewpoints from both of them during the training process.

6 Viewpoint ranking

Each candidate viewpoint $v_i \in \mathcal{V}$ of a 3D shape m is scored by Eq. 10 we learned from 2D sketch samples, which naturally reflects the possibility that human tends to draw the 3D shape from v_i. And the best viewpoints can be selected as the ones with the highest scores.

Since each v_i is densely sampled from the bounding sphere of the shape, nearby viewpoints always have close scores due to their similar contour maps. Therefore, if we select the top N best v_i by ranking the highest scores directly, results will be collected just in one side of the 3D shape, which is useless. In order to discover all possible viewpoints, diversity should be encouraged when they are ranked. Below, we detail our ranking algorithm, which encourages top-ranked viewpoints to correspond to different sides of a 3D shape.

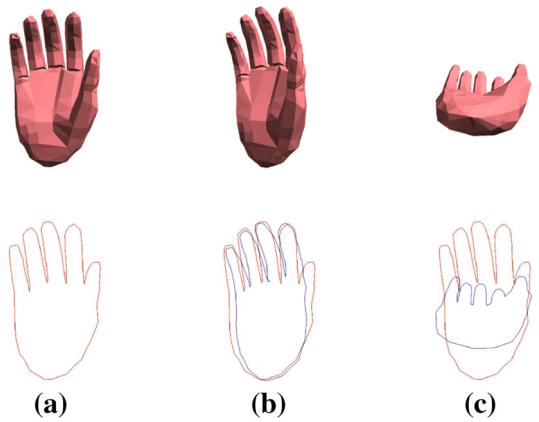

Fig. 5 We use the Intersection over Union (IoU) to computed the similarity between two viewpoints. *Top* different views of a 3D shape. *Bottom* relevant projection areas of different views. The IoU of viewpoint (**a**) and (**b**) is 0.87, while the IoU of viewpoint (**a**) and (**c**) is 0.43

Let s_i be the initial score of v_i computed by the classifier in Eq. 10. Inspired by the ranking strategy proposed in [5], we introduce a new scoring function $\tilde{s}_i$ that helps to rank viewpoints, which is formulated as

$$\tilde{s}_i = s_i + \alpha(\Phi(v_i)) \tag{11}$$

where $\Phi(v_i)$ is a penalty term to suppress similar viewpoints, and $\alpha(\cdot)$ is a monotonically decreasing function that controls the effect of $\Phi(v_i)$. We found that the specific choice of $\alpha(\cdot)$ is not very important, as long as it falls to zero for a moderate value. Experimentally, we use $\alpha(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$, with $\sigma = 0.2$. The similarity penalty term $\Phi(v_i)$ is computed as

$$\Phi(v_i) = \max_{\{v_j | v_j \in \tilde{\mathcal{V}}\}} IoU(v_i, v_j) \tag{12}$$

where $\tilde{\mathcal{V}}$ is a set of viewpoints including all the ones rank higher than v_i. The Intersection over Union (IoU) is employed to measure the similarity of two viewpoints. IoU is defined as the intersection of two viewpoints' projection areas divided by their union. Figure 5 shows examples of viewpoints with different IoU scores. The term $\Phi(v_i)$ penalizes viewpoints with high similarity to previously ranked viewpoints and leads to lower $\tilde{s}_i$. As a result, the function $\tilde{s}_i$ forces we to select new viewpoint from different sides of the 3D shape, which ensures the diversity. Note that without loss of generality, we define $\tilde{s}_i = s_i$ when $\tilde{\mathcal{V}}$ is empty.

To select the top N best viewpoints of the 3D shape, we iteratively rank all viewpoints by $\tilde{s}_i$ and select the viewpoint $\tilde{v}$ with the highest value as the candidate best one. Note that we perform mean-shift [2] from $\tilde{v}$ with s_i to find the local maximum to conduct a more stable best viewpoint v. Finally, we can obtain the best view set for the given 3D shape. Algorithm 1 summarizes our whole ranking process.

Algorithm 1 Top N best viewpoints ranking.

Input: initial scores s_i of each viewpoint $v_i \in \mathcal{V}$
Output: $\tilde{\mathcal{V}}$, the set of top N best viewpoints

$\tilde{\mathcal{V}} \leftarrow \{\}$
repeat
  for all $v_i \in \mathcal{V}$ **do**
    $\tilde{s}_i = s_i + \alpha(\Phi(v_i))$
  end for
  select $\tilde{v}$ with the highest $\tilde{s}_i$
  use mean-shift to find v from $\tilde{v}$ with s_i
  $\mathcal{V} \leftarrow \mathcal{V} - v$
  $\tilde{\mathcal{V}} \leftarrow \tilde{\mathcal{V}} + v$
until $\tilde{\mathcal{V}}$ is not changed
return $\tilde{\mathcal{V}}$

7 Experiments

In this section, we conduct extensive experiments to evaluate the performance of our view selection algorithm. We demonstrate that our approach achieves competitive results as compared with other state-of-the-art methods. As mentioned in Sect. 5.1, the PSB dataset [4,26] is employed to train and evaluate our algorithm.

7.1 Evaluation of context similarity

The way to measure the similarity between two contour maps is very important to our approach, which influences the accuracy of our sampling and training stages. Thus we conduct an experiment that compares the similarity measurement proposed in Sect. 4 with human intuition to demonstrate its efficiency.

Firstly, we randomly sample one hundred shapes and one relevant sketch for each 3D shape in the test dataset. Given a shape m_i and its relevant sketch s_i, we invite ten users to select the proper viewpoint according to the sketch. To avoid strong bias and ensure universality, all participants are chosen from college students without professional painting background. And each user is asked to select the most likely view from all candidate positions (uniformly sampled as described in Sect. 5.1) of the shape if they draw the given sketch.

We treat these user-labeled data as the ground truth. Then we map the sketch to a viewpoint v_i according to Sect. 5.1 by different measurements, and computed the accuracy as

$$Accuracy(v_i) = \frac{1}{n}\sum_{k=1}^{n} IoU(v_i, \bar{v}_i^k) \tag{13}$$

where $\bar{v}_i^k$ is the viewpoint selected by the k-th user for shape m_i, and n is the number of users. Note that we use projection IoU defined in Sect. 6 to compute the similarity between two viewpoints. Finally, we regard the

Table 1 The average accuracy and runtime (seconds per shape) of different similarity measurements proposed in Sect. 4

Measure	Accuracy (%)	Time (s)
S_{app} (Eq. 6)	64.33	0.134
S^4_{con} (Eq. 5)	85.62	4.596
S_{key} (Eq. 7)	84.49	0.858

The results are tested on a desktop computer with an Intel 3.39 GHz Quad-core CPU

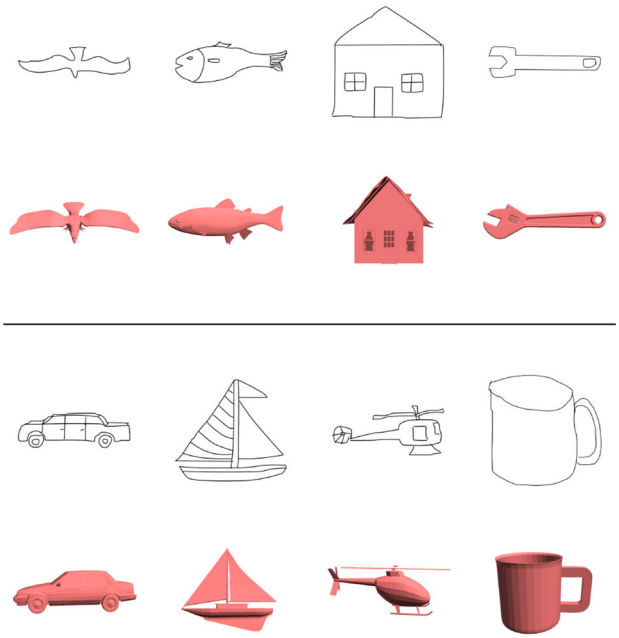

Fig. 6 Visual examples of mapping sketches to viewpoints of 3D shapes. We use the key context similarity measurement of contours (Eq. 7) to find the suitable viewpoint for a given sketch

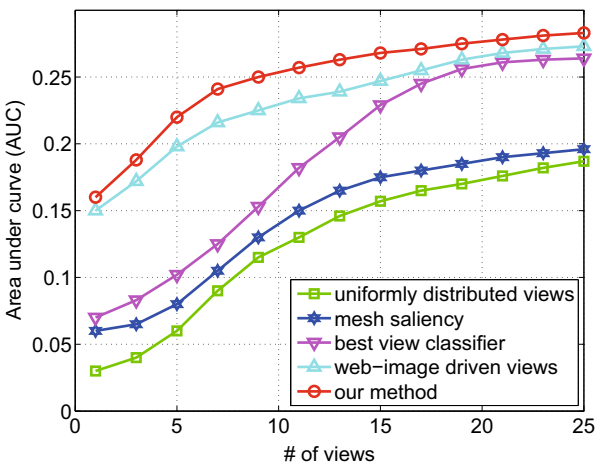

Fig. 7 Comparison of our best view selection approach to various state-of-the-art algorithms when applied to shape retrieval tasks, including uniformly distributed views, web image-driven method [17], best view classifier [4], and mesh saliency [13]

Table 2 Results of different view selection methods when applied to shape retrieval tasks

Method	AUC	# of views
50 uniformly distributed views	0.235	50
Mesh saliency [13]	0.202	34
Best view classifier [4]	0.229	15
Web image-driven views [17]	0.239	13
Our method	0.241	7

Our method obtains the standard area under the curve (AUC) performance with the fewest average number of views

mean accuracy of a similarity measurement of all shapes as its performance. We evaluate different similarity measurements proposed in Sect. 4 respectively, including S_{app} (Eq. 6), S^4_{con} (Eq. 5) and S_{key} (Eq. 7). Results are shown in Table 1.

As illustrated in Table 1, taking the context messages into account can significantly improve the accuracy, while more time is needed for computation. With minor loss of accuracy, using keypoint-based similarity measurement can speed up the whole process. Therefore, the key context similarity measurement in Eq. 7 is the most efficient method after balancing time and accuracy. Figure 6 shows some visual results of our key context similarity measurement.

7.2 Shape retrieval performance

Selecting best views is a very intuitive task for observing 3D shapes, thus the performance of a view selection method cannot be directly quantized from simple datasets. Instead, we evaluate the performance of our method by applying it into shape retrieval tasks.

We follow [4] to set up the experiment. Different view selection methods are used to compute the candidate viewpoint of each shape in the dataset, and we get final retrieval results respectively. Area under the curve (AUC) computed from the Precision-Recall Curve of a retrieval result is used to evaluate the retrieval performance. Then for each view selection method, we vary the parameters to gain different candidate viewpoints, and plot its AUC-#Views Curve. We compare our approach with other state-of-the-art methods, including uniformly distributed views, web image-driven method [17], best view classifier [4], and mesh saliency [13]. As shown in Fig. 7, our method outperforms the other methods using similar number of candidate views and especially when the number of views is relatively small.

According to [4], the standard retrieval performance is obtained when 50 uniformly distributed views are used, and we treat it as the baseline. Obviously, a view selection method is better if it reaches the standard performance with fewer

Fig. 8 Visual examples of best views obtained by different approaches. From *top* to *bottom* our method, web image-driven method [17], perceptually best view classifier [4], and mesh saliency [13]

selected candidate views. Table 2 shows the average view number of each method when its AUC is close to the standard performance. Our method achieves the standard retrieval performance with the fewest selected viewpoints. Specifically, the number of viewpoints generated by our approach is six times fewer than the baseline and nearly twice fewer than other state-of-the-art methods. Figure 8 shows some visual results of our approach.

7.3 Limitation

Our approach favors best views from the front or side of a 3D shape, which reflects the bias when human draws an object. Common 3D shapes existing in our daily life with regularly grid structures, such as birds, cars, fishes, instruments, and horses are well handled by our approach. Compared to [17], our approach sometimes selects different viewpoint for the same object, since different bias (views in photographing and drawing an object) are employed respectively.

However, we find that the performance of our approach usually depends on the quality of training samples. Poorly drawn sketches are harmful to the accuracy of our sampling stage in Sect. 5.1 and influence the final performance. Secondly, our method fails when dealing with odd objects that seldom appear in our daily life, for no such kind of features can be learned from hand-drawn sketches. Thirdly, our method cannot convey all possible human visual preferences for an object. For example, in Fig. 8, photographing-based approach [17] shows that people tend to observe vehicles (e.g. cars and motorcycles) from oblique views, while this kind of information is not carried by sketches. That is the limitation of our method.

8 Conclusion

We present a novel approach to select the best views of a 3D shape by hand-drawn sketches. We take advantage of context information extracted from contours to reveal the connection between sketches and viewpoints. Furthermore, a learning framework using the bag-of-features model is presented to generalize this connection. Experiments on the Princeton Shape Benchmark (PSB) dataset demonstrate the superiority of our approach when applied to shape retrieval tasks.

Future work includes exploring other effective context descriptors for sketches. In order to describe human visual preference more sufficiently, how to combine our method with other algorithm using different visual prior knowledge (e.g. photography-based approach [17]) is another important problem to solve. Additionally, we also need to explore if our method can be applied to other data, such as contours derived by edge detection in colored paintings or photos.

Acknowledgments This research work was supported by the National Science Foundation of China (No. 61272276, 61305091), the National Twelfth Five-Year Plan Major Science and Technology Project of China (No. 2012BAC11B01-04-03), Special Research Fund of Higher Colleges Doctorate (No. 20130072110035), the Fundamental Research Funds for the Central Universities (No. 2100219038), and Shanghai Pujiang Program (No. 13PJ1408200).

References

1. Bay, H., Tuytelaars, T., Gool, L.: Surf: speeded up robust features. In: European Conference on Computer Vision (ECCV) (2006)
2. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell **24**(5), 603–619 (2002)
3. DeCarlo, D., Finkelstein, A., Rusinkiewicz, S., Santella, A.: Suggestive contours for conveying shape. In: SIGGRAPH, pp. 848–855 (2003)
4. Eitz, M., Richter, R., Boubekeur, T., Hildebrand, K., Alexa, M.: Sketch-based shape retrieval. ACM Trans. Graph. (Proc. SIGGRAPH) **31**(4), 31:1–31:10 (2012)
5. Endres, I., Hoiem, D.: Category-independent object proposals with diverse ranking. IEEE Trans. Pattern Anal. Mach. Intell. **36**(2), 222–234 (2014)
6. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
7. Feldman, J.A., Feldman, C.M., Falk, G., Grape, C., Pearlman, J., Sobel, I., Tenenbaum, J.M.: The stanford hand-eye project. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 521–526 (1969)
8. Fisher, M., Hanrahan, P.: Context-based search for 3D models. ACM Trans. Graph. **29**(6), 182 (2010)
9. Giorgi, D., Mortara, M., Spagnuolo, M.: 3D shape retrieval based on best view selection. In: ACM workshop on 3D object retrieval (2010)
10. Laga, H.: Data-driven approach for automatic orientation of 3D shapes. Vis. Comput. **27**(11), 977–989 (2011)
11. Laga, H., Mortara, M., Spagnuolo, M.: Geometry and context for semantic correspondences and functionality recognition in man-made 3D shapes. ACM Trans. Graph. **32**(5), 150-1–150-16 (2013)
12. Laga, H., Nakajima, M.: Supervised learning of salient 2D views of 3D models. J. Soc. Art Sci. **7**(4), 124–131 (2008)
13. Lee, C.H., Varshney, A., Jacobs, D.: Mesh Saliency. In: SIGGRAPH (2005)
14. Li, B., Lu, Y., Godil, A., et al.: SHREC'13 track: large scale sketch-based 3D shape retrieval. In: Eurographics Workshop on 3D Object Retrieval (3DOR), pp. 89–96 (2013)
15. Li, B., Lua, Y., Li, C., et al.: A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries. Comput. Vis. Image Underst. **131**, 1–27 (2015)
16. Liang, S., Zhao, L., Wei, Y., Jia, J.: Sketch-based retrieval using content-aware hashing. In: Pacific-Rim Conference on Multimedia (PCM), pp. 133–142 (2014)
17. Liu, H., Zhang, L., Huang, H.: Web-image driven best views of 3D shapes. Vis. Comput. **28**(3), 279–287 (2012)
18. Liu, Y.J., Luo, X., Joneja, A., Ma, C.X., Fu, X.L., Song, D.: User-adaptive sketch-based 3-D CAD model retrieval. IEEE Trans. Autom. Sci. Eng. **10**(3), 783–795 (2013)
19. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **42**(3), 145–175 (2001)
20. Ma, C., Yang, X., Zhang, C., Ruan, X., Yang, M.H.: Sketch retrieval via dense stroke features. In: British Machine Vision Conference (BMVC) (2013)

21. Malisiewicz, T., Efros, A.A.: Beyond categories: the visual memex model for reasoning about object relationships. In: Annual Conference on Neural Information Processing Systems (NIPS) (2009)
22. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Int. J. Comput. Vis. **60**(1), 63–86 (2004)
23. Page, D., Koschan, A., Sukumar, S., Roui-Abidi, B., Abidi, M.: Shape analysis algorithm based on information theory. In: International Conference on Image Processing (ICIP) (2003)
24. Park, H.S., Jun, C.H.: A simple and fast algorithm for K-medoids clustering. Expert Syst. Appl. **36**(2), 3336–3341 (2009)
25. Shao, T., Xu, W., Yin, K., Wang, J., Zhou, K., Guo, B.: Discriminative sketch-based 3D model retrieval via robust shape matching. In: Pacific Graphics (PG) (2011)
26. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The Princeton Shape Benchmark. In: Shape Modeling International (SMI) (2004)
27. Shtrom, E., Leifman, G., Tal, A.: Saliency detection in large point sets. In: International Conference on Computer Vision (ICCV) (2013)
28. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: International Conference on Computer Vision (ICCV), pp. 1470–1477 (2003)
29. Vázquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy. In: 6th International Fall Workshop Vision, Modeling and Visualization (2001)
30. Wang, F., Lin, L., Tang, M.: A new sketch-based 3D model retrieval approach by using global and local features. Graph. Models **76**(3), 128–139 (2014)
31. Zhang, T.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. Commun. ACM **27**(3), 236–239 (1984)
32. Zhao, S., Ooi, W.T., Carlier, A., Morin, G., Charvillat, V.: Bandwidth adaptation for 3D mesh preview streaming. ACM Trans. Multimed. Comput. Commun. Appl. **10**(1), 13-1–13-20 (2014)
33. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: European Conference on Computer Vision (ECCV) (2014)

Long Zhao is currently a master student in School of Software Engineering, Tongji University, China. His research interests include computer graphics, computer vision and pattern recognition.

Shuang Liang is currently an assistant professor in School of Software Engineering, Tongji University, China. She received her B.S. degree from Zhejiang University in 2003, and Ph.D. degree from Nanjing University in 2008. Her main research interests include computer graphics & vision, human-computer interaction and machine learning.

Jinyuan Jia is currently a professor in School of Software Engineering, Tongji University, China. He received his Ph.D. degree from Department of Computer Science and Technology, The Hong Kong University of Science and Technology in 2004. His research interests include computer graphics, Web3D and mobile VR.

Yichen Wei joined Visual Computing Group, Microsoft Research Asia in 2006. Before that, he obtained his Ph.D. degree from The Hong Kong University of Science and Technology in 2006, and B.S.E degree from Peking University in 2001. His research interests include detection, tracking and recognition of generic objects, structure from motion, and stereo matching.