

# Sketch-based 3D model retrieval by incorporating 2D-3D alignment

Bo Li · Henry Johan

© Springer Science+Business Media, LLC 2012

**Abstract** Sketch-based 3D model retrieval is very important for applications such as 3D modeling and recognition. In this paper, a sketch-based retrieval algorithm is proposed based on a 3D model feature named View Context and 2D relative shape context matching. To enhance the accuracy of 2D sketch-3D model correspondence as well as the retrieval performance, we propose to align a 3D model with a query 2D sketch before measuring their distance. First, we efficiently select some candidate views from a set of densely sampled views of the 3D model to align the sketch and the model based on their View Context similarities. Then, we compute the more accurate relative shape context distance between the sketch and every candidate view, and regard the minimum one as the sketch-model distance. To speed up retrieval, we precompute the View Context and relative shape context features of the sample views of all the 3D models in the database. Comparative and evaluative experiments based on hand-drawn and standard line drawing sketches demonstrate the effectiveness and robustness of our approach and it significantly outperforms several latest sketch-based retrieval algorithms.

**Keywords** Sketch-based 3D model retrieval · 2D-3D alignment · View context

## 1 Introduction

Sketch-based 3D model retrieval is to retrieve 3D models using a 2D sketch as input. This scheme is intuitive and convenient for users to search for relevant 3D models and also important for several applications including sketch-based modeling [27] and

---

B. Li (✉) · H. Johan  
School of Computer Engineering, Nanyang Technological University,  
50 Nanyang Avenue, Singapore 639798, Singapore  
e-mail: libo0002@ntu.edu.sg

H. Johan  
e-mail: henryjohan@ntu.edu.sg

sketch-based recognition [40]. One example of integrating a sketch-based retrieval algorithm into a sketch-based modeling process is proposed by Fonseca et al. [10].

Currently, there exist many sketch-based 3D model retrieval algorithms such as [11, 16, 18, 25, 32, 38, 39]. However, to the best of our knowledge, all the available approaches compare the query 2D sketch with a very limited number of sample views of the 3D model. For example, Funkhouser et al. [11] only sampled 13 views rendered from 4 top corners, 6 edge midpoints and 3 adjacent face centers of a cube; Kanai [16], Yoon et al. [39] and Saavedra et al. [32] sampled only 14 views comprising 6 orthographic and 8 isometric views by sampling viewpoints on a cube or a sphere. In fact, this sparse view sampling approach is subject to inaccurate 2D sketch-3D model correspondence because the pose of the query sketch, that is, the viewpoint of the viewer when drawing the sketch, may have big difference with any of the sample views. Thus, the 2D-3D correspondence is not robust based on only several sample views generated using predefined fixed sample locations.

When retrieving 3D models using a 2D query sketch, we need to compute the distance between the 2D sketch and the 3D model. Ideally, it is good if we compare the 2D sketch with the most similar view or the optimal corresponding view of the 3D model. However, if we sparsely sample a limited number (e.g. 3~24 in previous work) of views, the chance that the optimal view is among the selected sample views will be low. However, due to the high computational cost, we also cannot exhaustively compare with a large amount of sample views of a 3D model.

Motivated by the above findings and in order to improve the retrieval performance, we propose a novel sketch-based 3D model retrieval algorithm which first performs a 2D sketch-3D model alignment before 2D-3D matching. Our proposed 2D sketch-3D model alignment utilizes a 3D model feature named View Context [19] to rapidly select some candidate views from a set of densely sampled views. View Context is utilized because we have found a new property of it: View Contexts of different views of the same model are often distinctively different. This property facilitates us to distinguish different views during the candidate views selection for 2D-3D correspondence. Our sketch-based retrieval algorithm is composed of two stages which are precomputation and retrieval. The retrieval stage comprises two steps which are 2D-3D alignment and 2D-3D matching. The effectiveness as well as the robustness of our approach are demonstrated by comparative and evaluative experiments, using both hand-drawn sketches and standard line drawings as queries and a standard 3D model dataset as target database. Moreover, we have achieved a better performance than several latest sketch-based retrieval algorithms.

The rest of this paper is organized as follows. In Section 2, we review the related work in sketch-based 3D model retrieval. Feature extraction and feature distance computation methods for 3D model and 2D sketch are presented in Sections 3 and 4, respectively. In Section 5, we present our sketch-based 3D model retrieval algorithm. Experiments are conducted in Section 6. Section 7 concludes the paper and lists several future research directions.

## 2 Related work

Funkhouser et al. [11] developed a search engine that supports 2D/3D sketch queries. To measure the distance between a 2D sketch and a 3D model, they applied the

3D spherical harmonics [17] method to the 2D sketch in an analogous way to extract a rotation-invariant amplitude-related feature vector and then compared it with those of the 13 sample views. Similarly, Pu and Ramani [29, 30] extended 3D spherical harmonics [17] and shape distribution [28] from 3D models to 2D drawings and proposed a 2.5D spherical harmonics and 2D shape histogram respectively for the retrieval of CAD drawings. Lee et al. [18] matched a sketch with 24 possible orthogonal contour views, based on 6 standard view directions and 4 axis-aligned up-vectors. Squared distance transform is then applied and a sum of squared distances-based similarity metric is adopted to measure the sketch-model distance. Hou and Ramani [12, 13] used a multi-classifier to estimate the probability of the sketch belonging to each class and adopted a classifier combination scheme to find relevant classes. Cao et al. [3] proposed a different retrieval framework by reconstructing a 3D query model using Bezier surface representation based on user drawn sketches. It constructs an accurate enough 3D query model where users need to draw enough curves to specify the features, which means it may take more time for users to perform retrieval.

Kanai [16] proposed a sketch-based retrieval interface by employing two rotation-invariant features, which are generic Fourier descriptor (GFD) [41] and a variation of local binary pattern (LBP) initially proposed by Ojala et al. [26], to measure the distance between a 2D sketch and a rendered view of a 3D model. Wang et al. [38] proposed a sketch-based CAD model retrieval interface using three sketches and a skeleton image as input. To measure the similarity of a 2D outline sketch and the outlines of a 3D model, they adopted angular radial partitioning (ARP) [4]. It decomposes an outline sketch into a set of angular radial sectors, then applies Fourier transform to the statistics of the feature points' distribution, and finally uses the rotation-invariant magnitude vector to represent the 2D sketch. However, they compared the sketch with only the 3 standard outline views of a normalized 3D model. This is feasible for CAD model retrieval but not appropriate for general 3D model retrieval, for which the pose of the query sketch is often not one of the principle views.

Recently, Napoléon and Sahbi [24, 25] proposed another sketch-based retrieval algorithm. They utilized a multi-scale convexity/concavity (MCC) shape representation [1] to represent the contours of a set of (3~9) sampled views. To speed up the retrieval, a pruning strategy and a dynamic programming approach are adopted to match the MCC features of the sketch and the contours. Yoon et al. [39] proposed a sketch-based retrieval algorithm by matching the sketch with 14 rendered suggestive contours [7] feature views of a model based on the diffusion tensor fields feature representation for the sketch and sampled views. Using the same view sampling method and feature views as Yoon et al. [39], Saavedra et al. [32] proposed a sketch-based 3D model retrieval algorithm using a structure-based local approach (STELA) and achieved a better performance than an improved histogram of edge local orientations-based global approach (HELO) proposed by Saavedra and Bustos [33].

To summarize, previous sketch-based 3D model retrieval algorithms adopted a coarse 2D sketch-3D model matching framework. They measure the distance between a sketch and a 3D model by directly comparing the distances between the sketch and only a small number of predefined sample views and then choosing the minimum one. However, this framework has a shortcoming in terms of accuracy in the 2D-3D correspondence, which motivates us to develop a sketch-based retrieval

algorithm by integrating a 2D-3D alignment step to correspond the query sketch and the 3D model before the actual 2D-3D matching. Experimental results in Section 6 show that considering more views of a target 3D model to correspond with a query sketch for a more accurate 2D-3D correspondence improves the retrieval performance.

### 3 Feature extraction

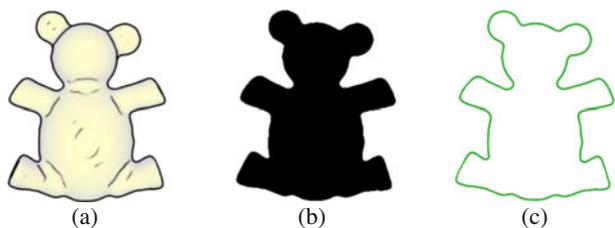
To measure the difference between a 2D sketch and the views of a 3D model effectively and efficiently, we need to extract similar yet simple features. In our algorithm, to represent different features of a view, we extract **silhouette** and **outline** feature views for both 2D sketches and 3D models. Silhouette feature view is selected because of its robustness for the 2D sketch-3D model alignment, while outline view is chosen because of its better accuracy in selecting the relevant models during the 2D-3D matching in the retrieval stage. Silhouette and outline feature views are simple in essence and often coexist in both the sketches and the sample views of an object and thus form a simple and similar feature set. Compared to the features in the related work section, such as 3D spherical harmonics, generic Fourier descriptor (GFD), local binary pattern (LBP), multi-scale convexity/concavity (MCC) as well as diffusion tensor fields feature representations of suggestive contours, the features we selected have the virtues of simple and low computational complexity.

#### 3.1 3D model feature extraction

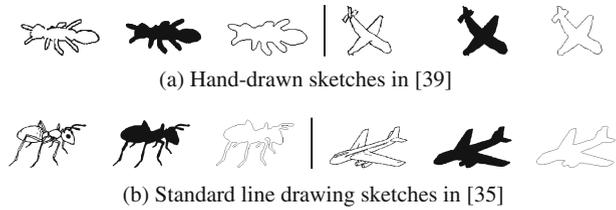
We render the silhouette and outline views based on orthographic projection. Due to the orthographic projection, there is a symmetrical property in rendering both feature views: two views rendered from two opposite camera locations are identical. One example showing the two feature views of a teddy model is shown in Fig. 1. Silhouette view depicts the region information of the view while outline view represents its contour information.

Silhouette and outline views are used to extract the View Context features of a 3D model and a 2D sketch. Additionally, outline view is also used to compute a more accurate relative shape context matching [2] cost between the sketch and each candidate view resulting from the 2D-3D alignment.

**Fig. 1** The feature views of a 3D teddy model. **a** The teddy model; **b** silhouette view; **c** outline view



**Fig. 2** Four sets of examples of sketch feature extraction for both hand-drawn sketches in [39] and standard line drawings in [35]. For every set, from *left to right*: sketch, silhouette view and outline view



### 3.2 2D sketch feature extraction

A sketch is composed of a set of curves. It can be: (1) a hand-drawn sketch drawn by non-artist people, such as the sketches built by Yoon et al. [39]; (2) a sketch drawn by artists, for example, the sketch dataset created by Cole et al. [6]; or (3) a standard line drawing such as the 260 Snogross and Vanderwart's standardized 2D object images [35].

We need to extract the silhouette and outline feature views for a 2D sketch to correspond with a 3D model. We generate a silhouette view based on the following steps: binarization, Canny edge detection, morphological closing operations (repeat until the result no longer changes), gap connection and region filling. After that, we apply the Canny edge detector on the binary silhouette image to extract the outline of the sketch. Figure 2 illustrates two groups of examples of hand-drawn sketches in Yoon et al. [39] and standard line drawings in Snogross and Vanderwart [35].

## 4 Feature distance

To compute the distance between two feature views, we need to extract appropriate shape descriptors to balance the efficiency and accuracy in different stages of our retrieval algorithm. For the View Context feature extraction used in the precomputation stage and the 2D-3D alignment step in the retrieval stage, we adopt a computationally efficient integrated image descriptor. For the 2D-3D matching during the retrieval stage, we utilize the more accurate relative shape context descriptor.

### 4.1 Integrated image distance

We compare two sets of feature views based on an integrated image descriptor, motivated by the Light Field descriptor proposed by Chen et al. [5] and their source code. To represent the region and contour information of the feature views, we adopt the shape descriptor proposed by Zhang and Luo [42] and use 35 Zernike moments  $Z$  to represent the silhouette view and 10 Fourier descriptors  $F$  to represent the outline view. Zernike moments  $Z$  are normalized into the range of (0,1) by dividing them by the area of the 2D shape. Fourier descriptors  $F$  are normalized by the constant component  $F(0)$ , which is also named the direct current (DC) component of the Fourier descriptor series. Thus,  $F \in (0, 1)$ . In addition, to depict the geometric information of the outline view, we extract its eccentricity and circularity features. Eccentricity is to measure how much a shape deviates from a circle. For a 2D

shape defined by  $n$  points  $\{(x_i, y_i) | i = 1, \dots, n\}$ , we adopt the following definition to compute its eccentricity,

$$E = \left[ \sum_{i=1}^n (x_i - c_x)^2 - \sum_{i=1}^n (y_i - c_y)^2 \right]^2 + 4 \cdot \frac{[\sum_{i=1}^n (x_i - c_x)(y_i - c_y)]^2}{[\sum_{i=1}^n (x_i - c_x)^2 + \sum_{i=1}^n (y_i - c_y)^2]^2}, \quad (1)$$

where  $(c_x, c_y)$  is the center of the bounding box of the 2D shape. For our case, the 2D shape is a closed curve and the range of its eccentricity is  $[0,1)$ . Circularity is to measure the compactness of the shape. It is defined as the quotient of the area of the shape and the area of a circle with the same perimeter:  $C = 4 * \pi * A / P^2$ , where  $C$  is the circularity,  $A$  and  $P$  are the area and perimeter of the shape, respectively. We use the city block (L1) distance metric to measure the distances of Zernike moments ( $d_Z$ ), Fourier descriptors ( $d_F$ ), Eccentricity descriptor ( $d_E$ ) and Circularity descriptor ( $d_C$ ).

$$d_Z = \sum_{p=1}^{35} |Z_1(p) - Z_2(p)|, \quad (2)$$

where  $Z_1$  and  $Z_2$  are the Zernike moments features of two silhouette views.

$$d_F = \sum_{q=1}^{10} |F_1(q) - F_2(q)|, \quad (3)$$

where  $F_1$  and  $F_2$  are the Fourier descriptors features of two outline views.

$$d_E = |E_1 - E_2|, \quad (4)$$

where  $E_1$  and  $E_2$  are the eccentricity features of two outline views,  $d_E \in [0, 1)$ .

$$d_C = |C_1 - C_2|, \quad (5)$$

where  $C_1$  and  $C_2$  are the circularity features of two outline views,  $d_C \in [0, 1)$ .

The integrated image distance  $d$  between two sets of feature views is the combination of the above four component distances,

$$d = d_Z + d_F + d_E + d_C. \quad (6)$$

The four features  $Z, F, E, C$  depict a feature view from different aspects and they have the same contribution in the computation of the integrated image distance. Therefore, we linearly combine them and assign the same weight for each feature.

#### 4.2 Relative shape context matching distance

We use the relative shape context matching [2] to compute a more accurate distance to measure the difference between the sketch and each candidate view resulting from the alignment step during the retrieval stage. Relative shape context is defined to achieve rotation invariance property and it is necessary for our sketch-based retrieval scenario, for which sample views should be independent of camera up-vectors during rendering and the orientation of the sketch.

To compute the difference between two outline feature views, we first sample a set of feature points for each image and then use the relative shape context matching algorithm described in [2] to measure their distance.

- (1) *Feature points sampling.* We sample 100 points for every outline feature view based on the following steps: contour extraction, cubic B-Spline interpolation and uniform sampling.
- (2) *Relative shape context matching* We first extract the relative shape context feature [2] for every feature point in an outline view and then adopt Jonker's LAP algorithm [15] to correspond the feature points of two outline views and finally use the minimum matching cost to measure their distance. To compute the relative shape context, we compute the tangent vector to define the local relative  $x$  axis for each sample point. This can be easily achieved considering that we use a cubic B-spline to interpolate the contour during the above feature points sampling process and the derivative curve of a cubic B-spline curve is a quadric B-spline curve [31].

## 5 Our sketch-based 3D model retrieval algorithm

As described in Sections 1 and 2, many previous sketch-based 3D model retrieval algorithms (e.g. [18, 25, 38]) sample only a limited number (e.g. 3~24) of views to match a 3D model with a query 2D sketch. Apparently, as mentioned in Section 1, this sparse view sampling approach will limit the accuracy of the 2D-3D correspondence. This is because if the pose of the query sketch is apparently different from those of the limited number of predefined sampling views, the 2D-3D correspondence is not accurate. Thus, the 2D-3D matching distance cannot represent the real difference between the 2D sketch and the 3D model. Motivated by the above findings, we propose to first perform a 2D sketch-3D model alignment step to find a set of candidate views for the 2D-3D correspondence and then compute the 2D-3D matching distance based on the candidate views.

It should be noted that our 2D sketch-3D model alignment is different from the common 2D image-3D model registration techniques [9, 14] which optimize the rotation angles and the translation and scaling parameters to register a 3D model with a 2D image. Firstly, their 2D image and 3D model depict the same object. However, for our case they are not and some differences are often existent. Secondly, previous 2D image-3D model registration techniques used 2D real image which has brightness (shading) information and developed an as accurate as possible 2D-3D alignment. For our case, we use sketch which only has line information and since the 2D sketch and the 3D model are not completely the same, an approximate alignment is sufficient.

In this section, we present a sketch-based 3D model retrieval algorithm utilizing the 3D model feature View Context [19] and relative shape context matching [2]. It includes two stages: precomputation and retrieval. During the retrieval stage, we first select a set of candidate views to align a 2D sketch with a 3D model based on the precomputed View Context features of the 3D model before measuring their more accurate distances, in terms of relative shape context matching cost. The 2D-3D alignment step avoids brute-force direct matching between the sketch and many sample views, that is reducing the search space to only a set of candidate views, by

utilizing the features of all the densely sampled views to efficiently shortlist several good candidate views for a more accurate 2D-3D correspondence.

The idea of View Context was originally proposed in [19] for 3D model retrieval using 3D model queries. In [20] we found that View Context can be utilized to align a 2D sketch with a similar 3D model and performed preliminary tests on some models. Based on [19] and [20], we found a new property of View Context: for different views of the same model, their View Contexts are often different. Therefore, View Context can be utilized to distinguish different sample views of the model, thus useful for candidate views selection for the 2D-3D alignment. Based on this, we develop our main idea to align a 3D model with a 2D sketch as follows: we replace each sample view of the 3D model with the sketch and compute its View Context and if the obtained new View Context is very similar to its original one, then this sample view is considered as a candidate view for the 2D-3D alignment.

### 5.1 View context

To meet the requirements of the 2D-3D alignment step in our sketch-based retrieval algorithm, we modify the View Context proposed in [19] and [20] by choosing a fixed set of base views described as follows and an integrated image descriptor presented in Section 4.1 for feature distance computation. For a 3D model centered at the origin, we select a series of views as base view set  $\mathbf{V}^b$ ,

$$\mathbf{V}^b = \langle V_1^b, V_2^b, \dots, V_m^b \rangle, \quad (7)$$

where  $m$  is the number of base views. For a view  $V$ , its View Context is defined as the visual information differences between  $V$  and each view in the base view set  $\mathbf{V}^b$ ,

$$\{d(V, V_j^b) | V_j^b \in \mathbf{V}^b, 1 \leq j \leq m\}, \quad (8)$$

where  $d(V, V_j^b)$  is the integrated image distance (Section 4.1) between  $V$  and  $V_j^b$ . Thus, View Context measures the shape appearance deviation feature of a 3D model with respect to a set of base views.

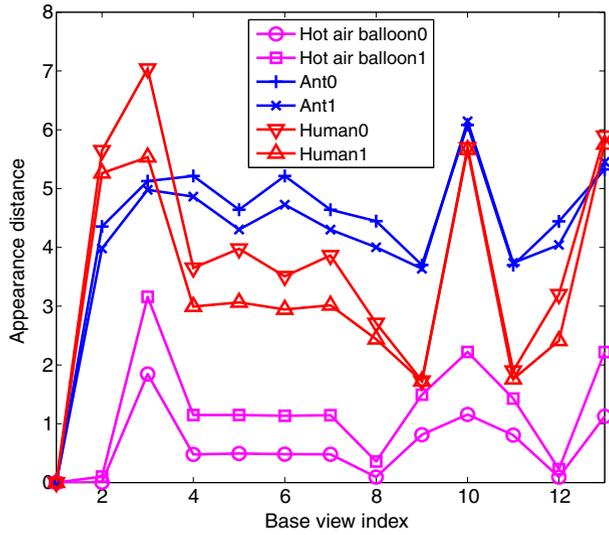
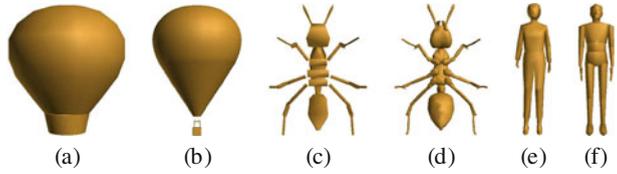
Figure 3 shows the View Contexts of several models: Fig. 3g plots the View Context features of the initial poses of the six models in Fig. 3a–f. In these examples, for demonstration, the base view set  $\mathbf{V}^b$  consists of 13 views rendered from the 4 top corners, 6 edge midpoints and 3 adjacent face centers of a cube centered at the origin.

We can see that for similar models their View Context features are similar and different models are distinctively different in their View Contexts. Moreover, we found that View Contexts of different views of the same model are also often different, as shown in Fig. 4. This newly found property is important for sketch-based retrieval framework to distinguish different sample views of a 3D model for the 2D-3D alignment.

### 5.2 Precomputation stage

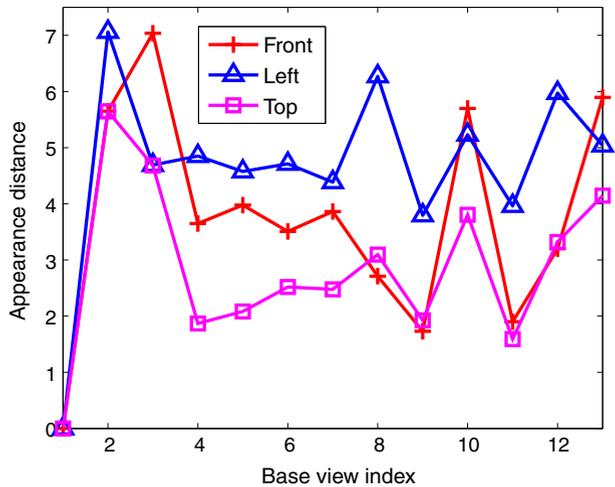
To speed up retrieval, we precompute the View Context and relative shape context features for a set of sample views of each target 3D model in database.

**Fig. 3** View contexts of six models: **a** hot air balloon0; **b** hot air balloon1; **c** Ant0; **d** Ant1; **e** Human0; **f** Human1



(g) Views context plots

**Fig. 4** An example indicating that view contexts of different views of the same model are often different. The view contexts of the front, left and top views of the model Human0 in Fig. 3e are shown



### 5.2.1 View context precomputation

The View Context feature computation for every 3D model is detailed as follows.

- (1) *Base and sample views definitions.* We define the viewpoints for the base and sample views by subdividing an icosahedron based on the Loop subdivision rule [22]. Figure 5 shows the view sampling by subdividing the icosahedron ( $L_0$ ) once ( $L_1$ ) and twice ( $L_2$ ) and we set the cameras at the vertices of the subdivided icosahedron for the base and sample view sequence generation. Considering the symmetrical property in rendering the feature views (Section 3.1), we select half- $L_1$  (select one from pair symmetric vertices, 21 views) for the base views and half- $L_2$  (81 views) for the sample views. We denote the sample view set  $\mathbf{V}^s$  as follows,

$$\mathbf{V}^s = \langle V_1^s, V_2^s, \dots, V_n^s \rangle, \quad (9)$$

where  $n$  is the number of sample views. Thus,  $n = 81$ ,  $m = 21$ .

- (2) *View context feature extraction.* We compute the integrated image distance (Section 4.1) between each sample view in  $\mathbf{V}^s$  and each base view in  $\mathbf{V}^b$ . Assume that  $d_{ij}$  ( $i = 1, \dots, n$ ;  $j = 1, \dots, m$ ) is the distance between the sample view  $V_i^s$  and the base view  $V_j^b$ , then for each model we form an  $n \times m$  view distance matrix  $D^s = \{d_{ij}\}_{n \times m}$ . The  $i^{\text{th}}$  row represents the View Context feature of the sample view  $V_i^s$ , that is,  $D_i^s = \langle d_{i1}, d_{i2}, \dots, d_{im} \rangle$ .

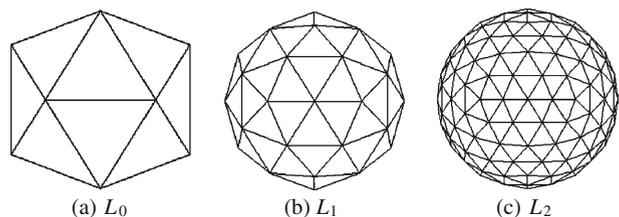
### 5.2.2 Relative shape context precomputation

We also precompute the relative shape context features (Section 4.2) for the sample views of each 3D model. They will be used in the retrieval stage. To improve the storage efficiency, we adopt a sparse matrix representation to denote the relative shape context features and only keep the feature values that are non-zeros (e.g.  $> 1e-5$ ) and save them into a series of three-dimensional vector  $\langle \theta, r, \text{value} \rangle$ , where  $(\theta, r)$  denotes one bin ( $\theta$ : orientation,  $r$ : distance) of the relative shape context partition, for which we use the default  $5 \times 12$  partition. During the retrieval stage, we thus only need to extract the relative shape context features for the query sketch.

## 5.3 Retrieval stage

Based on the precomputed View Context and relative shape context features of the 81 sample views for each target 3D model, we develop a retrieval algorithm comprising two steps: 2D-3D alignment and 2D-3D matching. The details are as follows.

**Fig. 5** Viewpoints sampling method.  $L_0$ : icosahedron;  $L_1$ ,  $L_2$ : subdivide the icosahedron once and twice



### 5.3.1 Step 1. 2D-3D alignment

- (1) *2D sketch feature extraction.* First, we extract the silhouette and outline views of the query 2D sketch based on the method in Section 3.2 and then compute its Zernike moments, Fourier descriptors, eccentricity and circularity features (Section 4.1).
- (2) *Sketch's view context feature extraction.* Similar to the View Context precomputation (Section 5.2.1) for a 3D model, we compute the integrated image distances (Section 4.1) between the sketch and all the base views of the target model and name the resulting distance vector  $D^k = \langle d_1, d_2, \dots, d_m \rangle$  sketch's View Context.
- (3) *2D-3D alignment.* To align the 2D sketch and a 3D model before 2D-3D matching, we choose some candidate views by keeping a certain percentage  $T$  (e.g. 20%, 10% or 5%, that is, 16, 8 or 4 sample views in our experiments) of the sample views with top View Context similarities as the sketch, in terms of correlation similarity  $S_i$ ,

$$S_i = \frac{D_i^s \cdot D^k}{\|D_i^s\| \|D^k\|}. \quad (10)$$

where,  $D_i^s$  (defined in Section 5.2.1) and  $D^k$  are the View Contexts of the sample view  $V_i^s$  of the 3D model and the 2D sketch, respectively.

### 5.3.2 Step 2. 2D-3D matching

- (1) *Sketch-model distance computation.* To more accurately measure the similarity between the sketch and the model as well as to encompass the orientation differences between the sketch and the sample views, we compare the sketch with every candidate outline view using the relative shape context matching (Section 4.2) and regard the minimum relative shape context distance obtained as the sketch-model distance.
- (2) *Ranking and output.* We sort all the sketch-model distances between the sketch and the models in an ascending order and list the retrieved models accordingly.

## 6 Experiments and discussion

To evaluate our sketch-based retrieval algorithm using a 2D-3D alignment, we perform comparative and evaluative experiments based on both hand-drawn and standard line drawing query sketches, as well as a standard 3D model database. We would like to mention that our 2D sketch-3D model alignment is different from the previous 2D image-3D model registration techniques, where the 2D image contains the view of the same object as the 3D model. Thus, they can use objective metrics to measure the alignment accuracy. However, for us the object in the 2D sketch is not completely the same as the 3D model and thus it is not one of its complete views. Therefore, there is no one exact pose to perfectly align the 2D sketch with 3D models. As a result, we mainly evaluate the alignment accuracy by comparing the robustness (change in performance) of our retrieval algorithm while reducing the number of candidate views during the alignment.

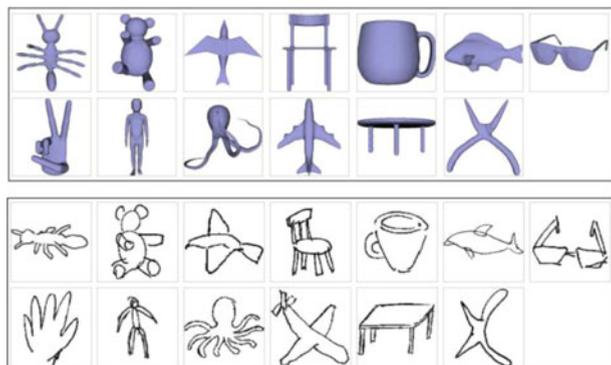
## 6.1 Hand-drawn sketches

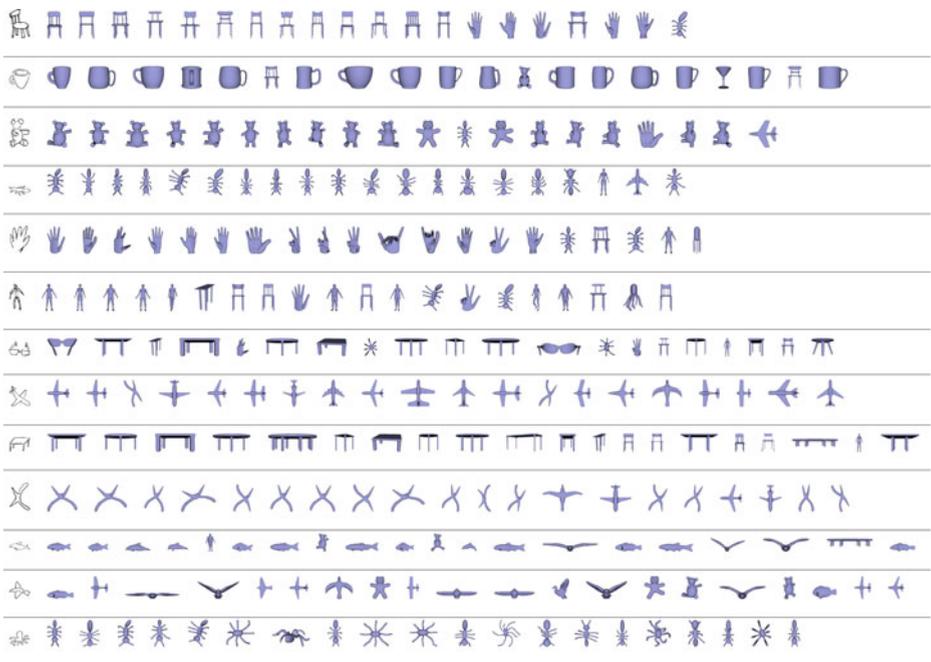
We first test and evaluate our sketch-based retrieval algorithm by performing a similar experiment as the one described in a 2010 paper by Yoon et al. [39]. They built a benchmark database by using the first 260 models (13 classes, 20 each) of AIM@Shape Watertight Model Benchmark (WMB) dataset [36] as target 3D model dataset and 250 hand-drawn sketches as the query sets. For each class, one typical 3D model and sketch are shown in Fig. 6. We need to mention that: (1) to compare with the available retrieval results obtained by Yoon et al. in this section, we select the same sketches as those in their paper; (2) the hand-drawn sketches were drawn by non-artists and some of them are very simple sketches, e.g., using only 4 line segments to represent an ant. We will give the overall performance of our approach later.

For the precomputation (Section 5.2), on average it takes 97 sec to process a model using a computer with an Intel Xeon CPU E5520@2.27 GHz and 12.0 GB of RAM: 8.8 sec for the View Context precomputation and 88.2 sec for the relative shape context precomputation, for all the 81 sample views of the 3D model. During the retrieval stage (Section 5.3), we set the default value for the percentage  $T$  for candidate views selection (Section 5.3) to be 20%, that is, keeping top 16 candidate views. We use the sketches in Fig. 6 as queries and the top-20 retrieved models are listed respectively in Fig. 7. Compared to the retrieval results obtained by Yoon et al. [39], as shown in Fig. 8, our retrieval lists are better for the bear, ant and hand queries and comparable for the chair and cup queries. For the human and glasses sketches, Yoon et al. achieved better results (in Section 6.3, we will show that our approach achieves better performance on class level). For the seven queries, the average accuracy (the percentage of the relevant models) in the top-8 retrieval results of our algorithm and Yoon et al.'s are 80.4 and 76.8%, respectively. Thus, we have achieved a better performance.

To measure the retrieval accuracy of our algorithm, we adopt the performance metric of First Tier (FT). It defines how much percentage of a class has been retrieved among the top  $C$  list, where  $C$  is the cardinality of the relevant class of the query sketch. For our case,  $C = 20$ . We test the same queries as in Fig. 7 with different percentages ( $T = 20\%$ ,  $10\%$  and  $5\%$ ) for candidate views selection. Table 1 compares their FT scores.

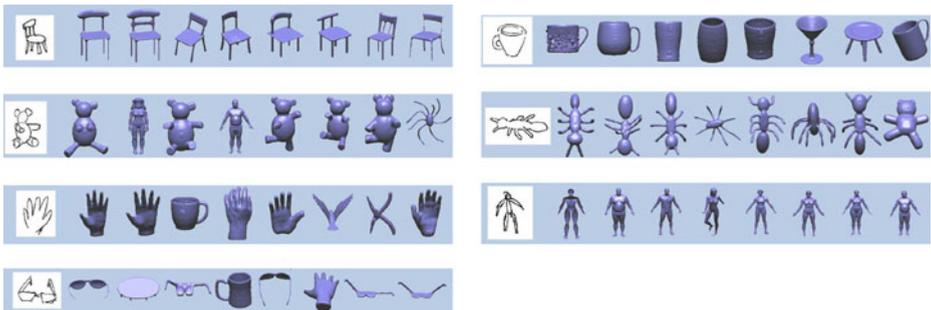
**Fig. 6** Typical 3D model and 2D sketch for each class of Yoon et al.'s [39] benchmark





**Fig. 7** Hand-drawn sketch-based retrieval examples on WMB database using hand-drawn sketches in [39]. The first 20 models are listed.

We can see that when we reduce the number of candidate views to be half of the default value ( $T = 20\%$ , 16 views), that is, 8 views, the average FT score decreases only 3.6%. Even after reducing it further to be only a quarter of the default value, that is, only 4 candidate views, the FT score drops only 9.3% averagely. This indicates the robustness of our sketch-based retrieval algorithm with respect to the number of candidate views. The relatively high FT scores also demonstrate the accuracy of our retrieval algorithm. We note that for some classes, such as human and octopus, when  $T$  becomes higher, FT may decrease somehow. Our explanation is as follows.



**Fig. 8** Hand-drawn sketch-based retrieval results in [39]

**Table 1** First Tier performance comparison using different percentage  $T$  values and the thirteen query sketches in Fig. 7

$T$ (%)	Chair (%)	Cup (%)	Teddy (%)	Ant (%)	Hand (%)	Human (%)	Glasses (%)	Plane (%)	Table (%)	Plier (%)	Fish (%)	Bird (%)	Octopus (%)
20	70	85	85	90	75	45	10	85	75	80	65	45	35
10	55	85	80	85	70	50	10	75	70	80	65	45	35
5	50	80	80	85	70	45	10	70	55	75	55	35	40

When  $T$  is increased, more candidate views are considered to compute the sketch-model distance, that is to say, a longer sequence (e.g. 8 views when  $T = 10\%$  and 16 views when  $T = 20\%$ ) of sketch-view distances will be computed for each model. The sketch-model distance computed based on more candidate views may be smaller than that computed based on less candidate views. Therefore, when more candidate views are considered, the sketch-model distances between the sketch and some irrelevant models may become smaller and thus these irrelevant models will be pushed forward in the retrieval lists and this decreases the First Tier performance.

## 6.2 Standard line drawings

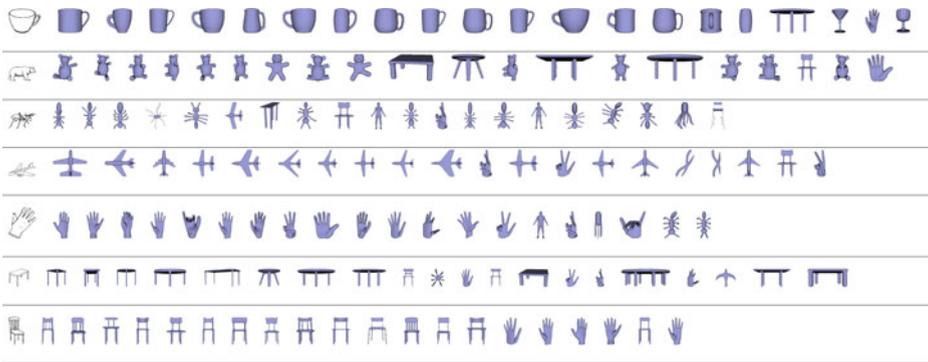
We perform a similar experiment as described in Section 6.1 using line drawing queries. We still use the same WMB database but utilize Snoggrass and Vanderwart's standard line drawings dataset [35] as queries. Figure 9 shows several line drawings examples that have relevant classes in WMB.

Similarly, we set the percentage  $T$  for candidate views selection to be 20%. We use the sketches in Fig. 9 as queries and their top-20 retrieval results are shown in Fig. 10. Table 2 shows the changes of the FT performance when using different percentage values for candidate views selection. The robustness of our sketch-based retrieval algorithm is verified again by the standard line drawing sketch queries. The decreases in the FT performance by changing  $T$  from 20 to 10% and from 20 to 5% are 3.1 and 7.1% on average.

## 6.3 Overall performance comparison

To assess the overall performance of our algorithm on a database level and perform a comparative evaluation with other approaches, we test our retrieval algorithm on the complete query set (250 sketches) of Yoon et al.'s [39] benchmark and compare the performance with a 2011 paper by Saavedra et al. [32]. They tested their proposed STELA approach on the same benchmark database and compared with the global shape descriptor-based approach HELO [33]. Table 3 and Fig. 11 compare the First Tier performances of our approach ( $T = 20\%$ ) and these two methods (STELA and

**Fig. 9** Typical sketches in Snoggrass dataset [35]



**Fig. 10** Standard sketch-based retrieval examples on WMB database using line drawings in [35]. The first 20 models are listed

HELO) on each class. For the performances of STELA and HELO, we refer to [32]. The average First Tier performances over all the classes are as follows: HELO: 13.9%, STELA: 16.5%, Ours: 41.5%. Apparently, we have achieved much better results in terms of respective classes and overall performance.

In addition, we want to compare our approach with the algorithm in Yoon et al. [39], in terms of the overall performance. Though we cannot find the complete overall performance data in the paper, according to our knowledge (personal communication with one of the author of the paper Yoon et al. [39]: Dr. Sang Min Yoon), the performance of Yoon et al. [39] is comparable to STELA, in terms of the overall First Tier performance as well as the First Tier performance for each class. Thus, our alignment-based retrieval approach also outperforms Yoon et al. [39].

To have a comprehensive evaluation of our algorithm, we further provide the results for other performance metrics including Precision-Recall plot, Nearest-Neighbor (NN), Second Tier (ST), E-measure (E), Discounted Cumulative Gain (DCG) and Average Precision (AP), as shown in Fig. 12 and Table 4 respectively. The meaning of the above performance metrics is as follows [21, 34]. Precision indicates how much percentage of the top  $K$  models belongs to the same class as the query model while recall means how much percentage of a class has been retrieved among the top  $K$  retrieval list. NN measures the percentage of the closest matches that are relevant models. ST is the recall of the top  $2(C-1)$  list, where  $C$  is the cardinality of the relevant class of the query model. E is used to measure the performance of the retrieval results with a fixed length, e.g. the first 32 models. It combines both the precision  $P$  and recall performance  $R$ :  $E = 2/(\frac{1}{P} + \frac{1}{R})$ . DCG is defined as the summed weighted value related to the positions of the relevant models. AP is to measure the overall performance and it combines precision, recall as well

**Table 2** First Tier performance comparison using different percentage  $T$  values and the seven query sketches in Fig. 10

$T$ (%)	Cup (%)	Bear (%)	Ant (%)	Plane (%)	Hand (%)	Table (%)	Chair (%)
20	90	70	55	70	80	60	75
10	85	65	40	70	75	55	75
5	90	55	30	70	70	55	80

**Table 3** First Tier performance comparison between our method and STELA [32], as well as HELO [33]

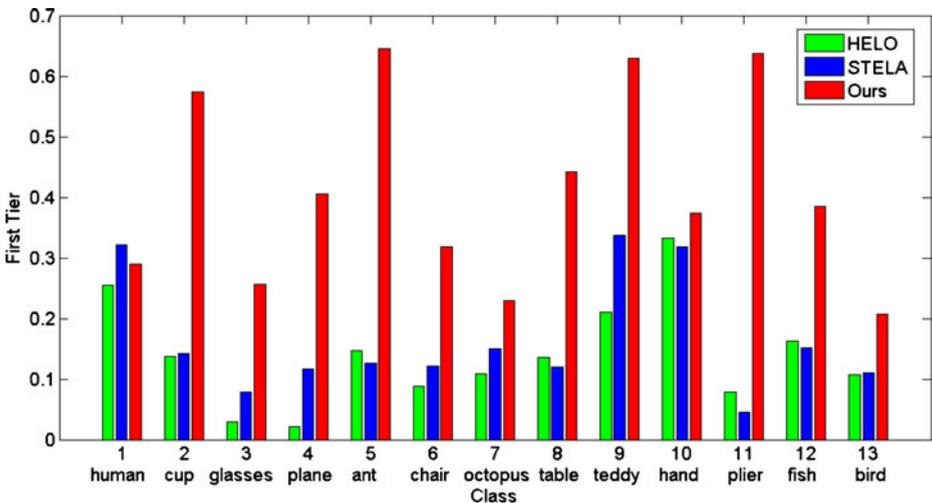
Methods	Chair (%)	Cup (%)	Teddy (%)	Ant (%)	Hand (%)	Human (%)	Glasses (%)	Plane (%)	Table (%)	Plier (%)	Fish (%)	Bird (%)	Octopus (%)
HELO	8.8	13.8	21.0	14.7	33.3	25.5	2.9	2.1	13.5	7.9	16.2	10.7	10.8
STELA	12.1	14.2	33.8	12.6	31.9	32.1	7.9	11.7	12.0	4.5	15.2	11.0	15.0
Ours	31.8	57.4	62.9	64.5	37.4	29.1	25.6	40.5	44.2	63.8	38.4	20.8	22.9

as ranking positions. A good AP needs both high recall and precision. AP can be computed by counting the total area under the Precision-Recall curve.

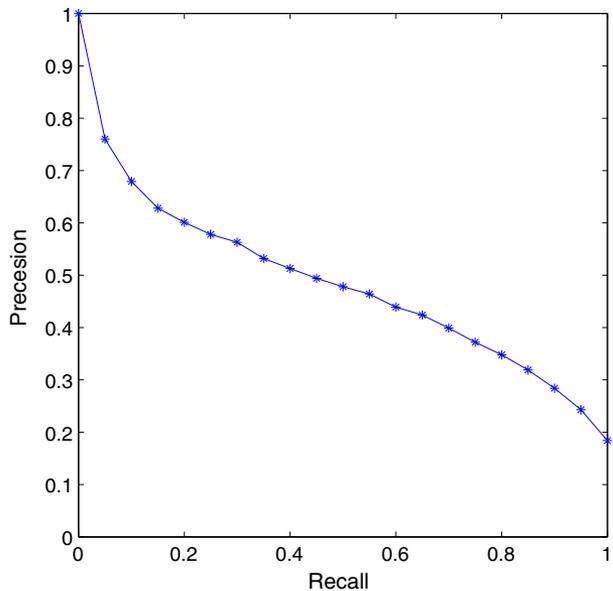
Similarly, we also perform the robustness experiment by changing the values of  $T$  and compare the results in Fig. 13. Their average First Tier performances over all the classes are as follows:  $T = 20\%$ : 41.5%;  $T = 10\%$ : 40.8%;  $T = 5\%$ : 38.9%. The conclusion is consistent with the previous ones, thus our retrieval algorithm is robust with respect to either respective models or classes.

#### 6.4 Extensibility to larger or other database

To test the extensibility of our SBR algorithm to a larger database, we use the complete 400 models in the WMB database. That is, we add 140 more models, classified into 7 classes, each 20 and regard them as noise. Example models for those 7 classes are shown in Fig. 14. We set  $T = 20\%$  and perform a similar experiment as the one in Section 6.3. Figure 15 compares the performance with that of the experiment done in Section 6.3 which uses Yoon's benchmark (260 models of the WMB database). We can see even when we added more models to the 3D dataset used in Yoon's benchmark, the performance is still stable and for most classes there are only trivial decrease. The average FT performance is 38.3% and it only drops

**Fig. 11** First Tier performance comparison between our method and STELA [32], as well as HELO [33]

**Fig. 12** Precision-Recall performance of our algorithm on the Yoon et al's [39] benchmark



3.2% compared to the performance achieved using Yoon's benchmark. We need to mention that the accuracy disparity in the "cup" class is due to the newly added "vase" class and some "vases" are quite similar to "cup", either in terms of the overall shape or their outlines. On the other hand, the outlines of some cup sketches are also similar to vases, which also shows a limitation of the outline feature representation.

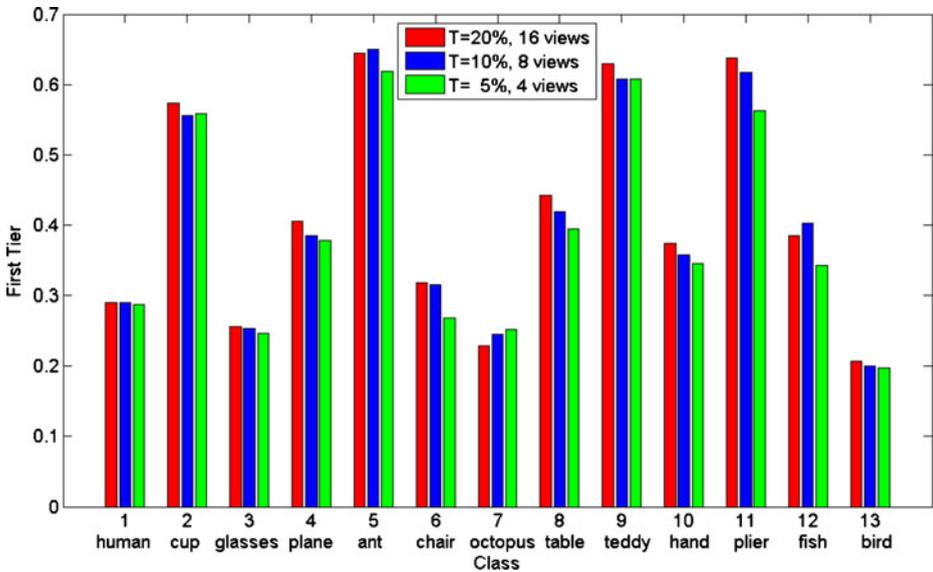
We further tested our algorithm on the NIST database [8] from which we select 260 models that have relevant classes in the Yoon et al.'s sketch dataset. 13 relevant classes were selected from NIST but we combined related classes according to the available sketch categories in Yoon et al.'s sketch database, in the end we got 8 classes. We set  $T = 20\%$  and the First Tier performances are: human: 23.6%, cup: 78.2%, glasses: 31.8%, plane: 60.8%, chair: 57.9%, table: 46.6%, fish: 43.7%, bird: 13.3%. The Average First Tier performance is 44.5%, which is comparable to the performance on Yoon et al.'s database.

## 6.5 Discussions

We found that a good pose to align a 3D model with a sketch often ranks high and for many of them it is among the top four. For example, Fig. 16 shows the top four candidate views for the cup, teddy and plane hand-drawn query sketches and cup, bear and plane line drawing query sketches. As seen in Fig. 16, in the top four candidate views for these sketches, usually we already can find certain views

**Table 4** Other performance metrics of our algorithm on the Yoon et al's [39] benchmark

NN	ST	E	DCG	AP
0.688	0.581	0.411	0.731	0.556



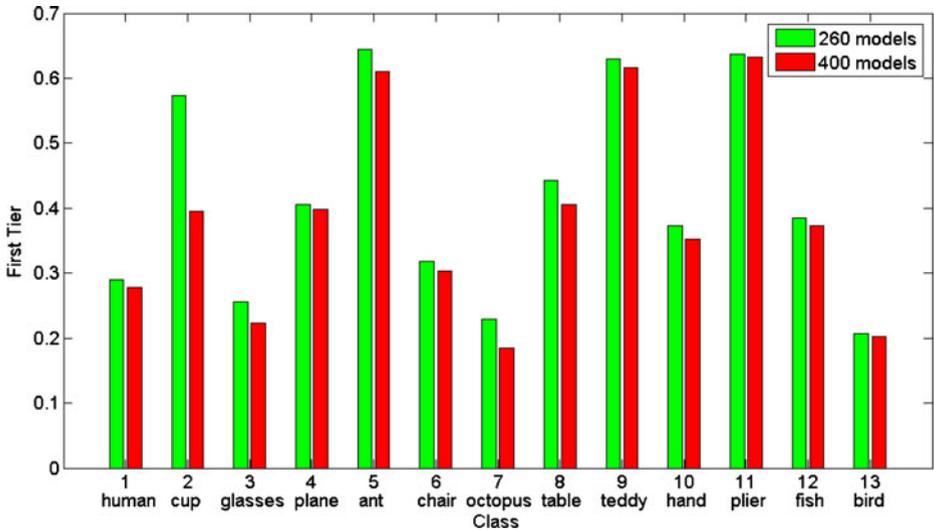
**Fig. 13** First Tier performance comparison using different percentage  $T$  values on the Yoon et al.'s [39] benchmark

of the 3D models that are accurate enough, in terms of retrieval, to correspond with the sketches. We also want to mention that the orientation differences between the sketch and candidate views in the examples, such as those existing in the four candidate views for the plane line drawing query (alignment results in the third row of Fig. 16), are not an issue for the retrieval since we utilize the relative shape context to encompass the variations in camera up-vectors during rendering.

To find out the contribution of 2D-3D alignment, we compared the performances of using the fixed sampling method and our alignment approach based on the same number of sample/candidate views. For the fixed one, we tested with Yoon et al.'s sampling method [39]: 6 orthographic and 8 isometric views. Because of the symmetrical property in rendering our feature views as described in Section 3.1, only half of the 14 sample views, that is 3 orthographic and 4 isometric views, are selected after aligning 3D models with Continuous Principal Component Analysis (CPCA) [37] method. For our algorithm, we keep the top 7 candidate views. We test them on the Yoon et al.'s database. Table 5 compares their First Tier scores with respect to each class and their overall First Tier performances are as follows: Fixed: 32.6%, Ours: 39.8%, which demonstrates an apparent improvement of using the 2D-3D alignment step to shortlist several candidate views. As can be seen from Table 5,



**Fig. 14** Typical 3D model for each of the added 7 classes

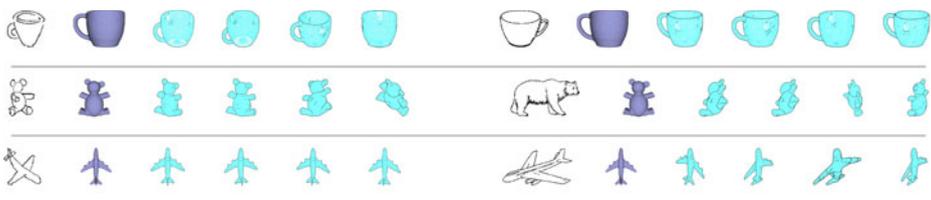


**Fig. 15** First Tier performance comparison using the whole WMB database (400 models) and only the relevant 260 models as the target 3D model database

compared to the fixed sampling approach, our method also achieves a more balanced performance especially on the chair, hand, human, glasses, table and plier classes.

Here, we also want to mention that the relative shape context matching part is also important to achieve a better performance. The clue can be also found from the fact that without alignment, that is using the fixed sampling approach, the relative shape context matching-based retrieval algorithm already achieves a First Tier performance of 32.6%, which already surpasses HELO (around 13.9%) and STELA (around 16.5%), referred to Section 6.3. However, if incorporating our 2D sketch-3D model alignment step to shortlist better candidate views, we further improve the retrieval performance to 39.8%. Therefore, both the View Context-based 2D sketch-3D model alignment and the relative shape context matching on the outline feature views, have important contributions to our apparently better performance than HELO and STELA.

To conclude, our 2D-3D matching considers a large number of sample views compared to previously sparse view sampling strategies, thus it is more robust to



**Fig. 16** 2D-3D alignment examples. Each row shows two sets of alignment results for a hand-drawn sketch and a line drawing sketch. For each result, from left to right: a 2D sketch, a 3D model (in initial pose) and the top 4 candidate views to align the 2D sketch and the 3D model

**Table 5** First Tier performance comparison between fixed sampling and our method

Methods	Chair (%)	Cup (%)	Teddy (%)	Ant (%)	Hand (%)	Human (%)	Glasses (%)	Plane (%)	Table (%)	Plier (%)	Fish (%)	Bird (%)	Octopus (%)
Fixed	16.3	51.6	57.1	52.4	16.6	16.2	10.6	40.5	32.1	38.5	46.6	20.5	24.7
Ours	28.1	55.3	59.5	64.2	34.2	28.1	25.6	38.3	41.3	60.3	38.4	20.0	24.7

different poses of the sketches. It can efficiently find several good candidate poses of a 3D model to align the model with a sketch. The above two types of experiments on the hand-drawn sketch queries and standard line drawing queries have demonstrated the effectiveness of our retrieval algorithm, which shows better performance than Yoon et al. [39] and Saavedra et al. [32, 33]. The robustness of our retrieval algorithm is also verified in our experiments.

## 6.6 Limitations

As shown in the above experiments, our approach has a good accuracy in terms of sketch-based retrieval. Nevertheless, it has some limitations. Firstly, the performances for some sketches (e.g. glasses, octopus and bird) are not as good as others and still have room for further improvements. Secondly, relative shape context matching part dominates the most part of the retrieval time: on average, it takes 0.86 sec to extract the features (Zernike moments, Fourier descriptors, eccentricity, circularity and relative shape contexts) for a sketch; only 0.37 msec for the 2D-3D alignment for a model; 17.5 msec for the 2D-3D matching based on relative shape context for a pair of sketch-candidate view. The average time for a complete retrieval on the Yoon et al.'s database is 19.5, 37.3 and 72.3 sec when keeping 4, 8 and 16 candidate views respectively. The retrieval time  $t$  (sec) is proportional to both the number of candidate views  $M$  and the number of the 3D models in the database, denoted by  $N$ . We denote  $\tau$  as the matching time for one candidate view, then the retrieval time  $t$  (sec) can be approximately formulated as follows:  $t = M * N * \tau$ . In our experiments,  $\tau = 0.0175$  sec.

According to the robustness analysis of our algorithm in Section 6.3 (Fig. 13), there is no much performance decrease when we keep fewer candidate views. Thus, we further tested our algorithm by keeping only 2 and 1 candidate view, and still got the average First Tier performances of 37.4% and 35.9% respectively, compared to the 41.5% when keeping 16 candidate views. The retrieval time is 10 and 5.4 sec respectively. Thus, our suggestion is that users can make decision for the tradeoff between the accuracy and efficiency based on the requirements of their respective applications and available resources.

## 7 Conclusions and future work

In this paper, we have presented a sketch-based 3D model retrieval algorithm based on the idea of first aligning a 3D model with a query 2D sketch before computing their matching distance. The algorithm comprises precomputation and retrieval stages. During the precomputation stage, we compute the View Context and relative shape context features of a set of densely captured sample views for each target

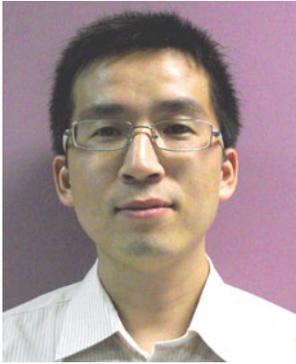
model. Based on the precomputed View Context features of a target model, in the retrieval stage we can efficiently and effectively align the model with the 2D sketch. Experiments based on hand-drawn and standard line drawings sketches demonstrate the superior performance and robustness of our approach. Thus, it has a potential to be used in applications, such as sketch-based 3D model recognition and modeling, as well as 3D scene reconstruction based on 2D sketches.

Several facets of the algorithm can be further explored. First, during the retrieval stage, we can use representative relative shape context [23] to speed up the matching process between the sketch and the candidate views since we can reject the candidate views that are obviously different from the sketches earlier. In addition, if using other faster correspondence algorithms to replace our adopted LAP algorithm or adopting some 2D image descriptors which are comparable in terms of effectiveness but more computationally efficient, we may improve the retrieval performance further. Second, we want to further test our sketch-based retrieval on other types of 3D model databases and sketches. Third, extending our algorithms to other types of queries is another interesting direction. For example, query by a 2D image or even a sketch of a 3D scene comprising several objects.

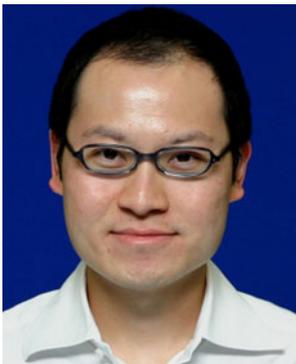
## References

1. Adamek T, O'Connor NE (2004) A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Trans Circuits Syst Video Technol* 14(5):742–753
2. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. *IEEE Trans Pattern Anal Mach Intell* 24(4):509–522
3. Cao L, Liu J, Tang X (2006) 3D object retrieval using 2D line drawing and graph based relevance reedback. In: Proc. of the 14th annual ACM international conference on Multimedia, MULTIMEDIA '06, pp 105–108
4. Chalechale A, Mertins A, Naghdy G (2004) Edge image description using angular radial partitioning. *IEE Proc, Vis Image Signal Process* 151(2):28–41
5. Chen DY, Tian XP, Shen YT, Ouhyoung M (2003) On visual similarity based 3D model retrieval. *Comput Graph Forum* 22(3):223–232
6. Cole F, Golovinskiy A, Limpaecher A, Barros HS, Finkelstein A, Funkhouser T, Rusinkiewicz S (2008) Where do people draw lines? *ACM Trans Graph* 27(3):1–11
7. Doug D, Adam F, Szymon R, Anthony S (2003) Suggestive contours for conveying shape. *ACM Trans Graph (Proc. SIGGRAPH 2003)* 22(3):848–855
8. Fang R, Godil A, Li X, Wagan A (2008) A new shape benchmark for 3D object retrieval. In: Proc. of the 4th international symposium on advances in visual computing (ISVC), pp 381–392
9. Feldmar J, Ayache N, Betting F (1997) 3D-2D projective registration of free-form curves and surfaces. *Comput Vis Image Underst* 65(3):403–424
10. Fonseca MJ, Ferreira A, Jorge JA (2004) Towards 3D modeling using sketches and retrieval. In: Sketch-based interfaces and modeling, pp 127–136
11. Funkhouser TA, Min P, Kazhdan MM, Chen J, Halderman JA, Dobkin DP, Jacobs DP (2003) A search engine for 3D models. *ACM Trans Graph* 22(1):83–105
12. Hou S, Ramani K (2006) Sketch-based 3D engineering part class browsing and retrieval. In: Proc. of EUROGRAPHICS workshop on sketch-based interfaces and modeling, pp 131–138
13. Hou S, Ramani K (2007) Classifier combination for sketch-based 3D part retrieval. *Comput Graph* 31(4):598–609
14. Huttenlocher D, Ullman S (1987) Object recognition using alignment. In: Proc. of IEEE international conference on computer vision (ICCV), pp 102–111
15. Jonker R, Volgenant A (1987) A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing* 38(4):325–340
16. Kanai S (2008) Content-based 3D mesh model retrieval from hand-written sketch. *Int J Interact Des Manuf* 2(2):87–98

17. Kazhdan M, Funkhouser T, Rusinkiewicz S (2003) Rotation invariant spherical harmonic representation of 3D shape descriptors. In: Proc. of the symposium on geometry processing, pp 156–164
18. Lee J, Funkhouser T (2008) Sketch-based search and composition of 3D models. In: Proc. of EUROGRAPHICS workshop on sketch-based interfaces and modeling, pp 97–104
19. Li B, Johan H (2010) View context: a 3D model feature for retrieval. In: Boll S et al (eds) MMM 2010. LNCS, vol 5916. Springer, Heidelberg, pp 185–195
20. Li B, Johan H (2011) View context based 2D sketch-3D model alignment. In: 2011 IEEE workshop on applications of computer vision (WACV), pp 45–50
21. Li B, Johan H (2011) 3D model retrieval using hybrid features and class information. Multimedia tools and applications, pp 1–26 (online first version)
22. Loop C (1987) Smooth subdivision surfaces based on triangles. Master thesis, University of Utah
23. Mori G, Belongie S, Malik J (2005) Efficient shape matching using shape contexts. IEEE Trans Pattern Anal Mach Intell 27(11):1832–1837
24. Napoléon T, Sahbi H (2010) From 2D silhouettes to 3D object retrieval: contributions and benchmarking. EURASIP J Image Video Process 2010:1–17
25. Napoléon T, Sahbi H (2010) Sketch-driven mental 3D object retrieval. SPIE, p 75260L
26. Ojala T, Pietikäinen M, Harwood D (1996) A comparative study of texture measures with classification based on featured distributions. Pattern Recogn 29(1):51–59
27. Olsen L, Samavati FF, Sousa MC, Jorge JA (2009) Sketch-based modeling: a survey. Comput Graph 33(1):85–103
28. Osada R, Funkhouser T, Chazelle B, Dobkin D (2001) Matching 3D models with shape distributions. In: Proc. of shape modeling and applications, pp 154–166
29. Pu J, Ramani K (2006) On visual similarity based 2D drawing retrieval. Comput-aided Des 38(3):249–259
30. Pu J, Ramani K (2006) ShapeLab: a unified framework for 2D & 3D shape retrieval. In: Proc. of the 3rd international symposium on 3D data processing, visualization and transmission (3DPVT 2006), pp 1072–1079
31. Buss SR (2003) 3D computer graphics: a mathematical introduction with OpenGL. Cambridge University Press, Cambridge, MA
32. Saavedra J, Bustos B, Scherer M, Schreck T (2011) STELA: sketch-based 3D model retrieval using a structure-based local approach. In: Proc. of the first ACM international conference on multimedia retrieval (ICMR'11), p 26
33. Saavedra JM, Bustos B (2010) An improved histogram of edge local orientations for sketch-based image retrieval. In: Gosele M, Roth S, Kuijper A, Schiele B, Schindler K (eds) DAGM-symposium. Lecture notes in computer science, vol 6376. Springer, pp 432–441
34. Shilane P, Min P, Kazhdan M, Funkhouser T (2004) The Princeton shape benchmark. In: Proc. of shape modeling and applications, pp 167–178
35. Snodgrass JG, Vanderwart M (1980) A standardized set of 260 pictures: norms for name agreement, image agreement, familiarity, and visual complexity. J Exp Psychol Hum Learn Mem 6(2):174–215
36. Veltkamp RC, ter Haar FB (2007) SHREC 2007 3D retrieval contest. Technical report UU-CS-2007-015, Department of Information and Computing Sciences, Utrecht University
37. Vranic DV (2004) 3D model retrieval. PhD thesis, University of Leipzig
38. Wang J, He Y, Tian H, Cai H (2008) Retrieving 3D CAD model by freehand sketches for design reuse. Adv Eng Inf 22(3):385–392
39. Yoon SM, Scherer M, Schreck T, Kuijper A (2010) Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours. In: ACM multimedia, pp 193–200
40. Zamora S, Sherwood T (2010) Sketch-based recognition system for general articulated skeletal figures. In: Proc. of EUROGRAPHICS workshop on sketch-based interfaces and modeling, pp 119–126
41. Zhang D, Lu G (2002) Shape-based image retrieval using generic Fourier descriptor. Signal Process: Image Commun 17(10):825–848
42. Zhang D, Luo G (2002) An integrated approach to shape based image retrieval. In: Proc. of the 5th Asian conference on computer vision (ACCV 2002), pp 652–657



**Bo Li** received the BS and MS degrees in computer science and technology from Xi'an Polytechnic University (China) in 2002 and Xi'an Jiaotong University (China) in 2005, respectively. He was a teaching assistant in the School of Science at Xi'an Polytechnic University from 2002 to 2007. From 2005 to 2007, he was a senior software testing engineer in Zhong Xing Telecommunication Equipment Company Limited. Since 2007, he has been a PhD student in the School of Computer Engineering at Nanyang Technological University (Singapore). His research interests include 3D model retrieval, shape matching and 3D modeling.



**Henry Johan** received his BS, MS and PhD degrees in computer science from the University of Tokyo (Japan) in 1999, 2001 and 2004, respectively. From 2004 to 2006, he was a post-doctoral fellow in the Department of Complexity Science and Engineering at the University of Tokyo. Since 2006, he has been an assistant professor in the School of Computer Engineering at Nanyang Technological University (Singapore). His research interests in computer graphics include rendering, animation and shape analysis. He is a member of ACM SIGGRAPH.