Eurographics Workshop on 3D Object Retrieval (2013)
S. Biasotti, I. Pratikakis, U. Castellani, T. Schreck, A. Godil, and R. Veltkamp (Editors)

Sketch-Based 3D Model Retrieval by Viewpoint Entropy-Based Adaptive View Clustering

Bo Li[1], Yijuan Lu[1], Henry Johan[2]

[1] Department of Computer Science, Texas State University, San Marcos, USA [2] Fraunhofer IDM@NTU, Singapore

Abstract

Searching for relevant 3D models based on hand-drawn sketches is both intuitive and important for many applications, such as sketch-based 3D modeling and recognition. We propose a sketch-based 3D model retrieval algorithm by utilizing viewpoint entropy-based adaptive view clustering and shape context matching. Different models have different visual complexities, thus there is no need to keep the same number of representative views for each model. Motivated by this, we propose to measure the visual complexity of a 3D model by utilizing viewpoint entropy distribution of a set of sample views and based on the complexity value, we can adaptively decide the number of representative views. Finally, we perform Fuzzy C-Means based view clustering on the sample views based on their viewpoint entropy values. We test our algorithm on two latest sketch-based 3D model retrieval benchmarks and compare it with other four state-of-the-art approaches. The results demonstrate the superior performance and advantages of our algorithm.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Computer Graphics]: Information Storage and Retrieval—Information Search and Retrieval

1. Introduction

Retrieving 3D models using human-drawn sketch(es) as input is an intuitive and easy way for users to search for relevant 3D models. It is also useful for related applications such as sketch-based 3D modeling and recognition. For example, sketch-based 3D model retrieval in the context of a 2D sketch image of a scene, such as a 2D storyboard, is usually a fundamental step of 3D animation production guided by 2D storyboards [TWLB09]. This is mainly because sketches are more human-friendly and people are more accustomed to "sketching" their ideas using a set of sketches. In 3D animation production, usually 2D storyboards are first drawn as an input. Then proper 3D models are retrieved from available 3D databases to build a 3D scene [TWLB09] while keeping the context information in the original 2D scene consistent.

Recently, quite a few sketch-based 3D model retrieval algorithms [FMK*03, Kan08, YSSK10, SXY*11, EHBA11, LSG*12] have been proposed. However, most of the available algorithms compare the 2D sketch query with a fixed number of predefined sample views of each 3D model, no matter how complex the model is. For example, Funkhouser et al. [FMK*03] compared a sketch with 13 sample views for each model by setting the cameras on the 4 top corners, 6 edge midpoints and 3 adjacent face centers of a cube; Both Kanai [Kan08] and Yoon et al. [YSSK10] adopted the 14 sample views approach which includes 6 orthographic and 8 isometric views. However, these sampling strategies cannot guarantee that the extracted sample views are representative enough to depict a 3D model since they do not consider the complexities of different models. In fact, there is no need to sample and compare 13 or 14 views for a simple model, such as a sphere or a cube, while more views should be sampled for a complicated model. That is, we need an adaptive sampling strategy.

Motivated by the above findings, we propose to sample different number of representative views according to the visual complexity (a measurement of shape complexity based on visual information) of a 3D model. A novel 3D visual complexity metric is proposed based on the viewpoint entropy distribution of a set of uniformly sampled views of the 3D model. It is further utilized to assign the number of cluster centers (representative views) during the Fuzzy C-Means (FCM) [Bez81] clustering of the sampled views based on their viewpoint entropy values. After that, shape context matching algorithm is used for the 2D-3D matching between the 2D sketch and the representative views of each model. The effectiveness as well as the advantages of our approach are demonstrated by comparative and evaluative experiments on two latest sketch-based 3D model retrieval benchmarks.

The main contributions introduced in this paper are summarized as follows:

- We quantitatively study and formulate the visual complexity analysis of a 3D model problem. Based on this, we propose a reasonable and effective 3D visual complexity metric by measuring the information theory-related viewpoint entropy.
- Our proposed approach shows promising application potential for sketch-based 3D model retrieval. The 3D visual complexity metric has been successfully applied into the sketch-based 3D model retrieval to adaptively decide the number of representative views for each 3D model. By a set of comparative experiments on two latest sketch-based 3D model retrieval benchmarks, the outperforming performance of our approach have been demonstrated.
- Our work will explicitly guide the research in visual complexity estimation of a 3D model and provide a path for view clustering-based 3D model retrieval.

2. Related Work

2.1. Sketch-Based 3D Model Retrieval

According to different view sampling strategies, sketch-based 3D model retrieval techniques can be categorized into two groups: (1) matching sketches with the predefined sample views rendered from certain fixed viewpoints; (2) matching sketches with the clustered views generated by view clustering.

Using Predefined Views. As mentioned before, most existing sketch-based 3D model retrieval algorithms compare sketches with views resulting from a set of predefined sample orientations. Recently, Yoon et al. [YSSK10] developed a sketch-based retrieval algorithm based on the diffusion tensor fields feature representation and matched a sketch with *14* suggestive contours feature views of a model. Shao et al. [SXY*11] proposed to perform the sketch-based retrieval based on a direct and robust contour-based shape matching method. For each model, they rendered and compared with *7* views which comprise 3 canonical views (front, side, top) and 4 corner views sampled on a cube. Eitz et al. [EHBA10] adopted a Bag-of-Features (BoF) framework and extracted Histogram of Gradient (HOG) local features for the subdivided patches of both sketch and views. They sampled *50* views and tested on the Princeton Shape Benchmark (PSB) [SMKF04a] using several sketches but did not provide the overall performance. In [EHBA11, LSG*12], they further sampled *102* views and performed experiments on a database level. Li and Johan [LJ12, LSG*12] proposed an algorithm by first approximately aligning a 2D sketch with a 3D model, in terms of shortlisting a set of candidate views to correspond with the 2D sketch, based on a 3D model feature named “View Context” [LJ10] before 2D-3D matching. Their 2D-3D matching algorithm is also based on relative shape context matching [BMP02].

This strategy has a shortcoming of ignoring the representativeness regarding the selected views and this also motivates us to develop a sketch-based retrieval algorithm by adaptively clustering the sample views.

Using Clustered Views. Compared to the approaches based on predefined views, much less research work has been done for the strategy based on view clustering. Mokhtarian and Abbasi [MA05] proposed a view clustering method by matching the rendered views and discarding the similar views whose matching costs fall in a predefined threshold. Ansary et al. [AVD07] proposed an image-based 3D model retrieval algorithm by clustering *320* sample views into a set of characteristic views based on the Bayesian probabilistic approach. They also developed a method to optimize the number (varying from 1 to 40) of characteristic views based on the X-means clustering method. Zernike moments are adopted to represent the views or 2D image queries. Unfortunately, only one demo result for sketch queries was given and no overall performance was evaluated.

2.2. Shape Complexity

Geometrical shape complexity approaches have been reviewed by Rossignac [Ros05] from five perspectives: algebraic, topological, morphological, combinational, and representational. Recently, a new tendency is to measure the visual complexity of a 3D model. This also has its foundation in computer vision and 3D human perception: a 3D object can be viewed as a collection of 2D views. It is also consistent with the human perception theory to utilize information theory to measure the shape complexity of 3D models. Saleem et al. [SBWS11] measured the visual complexity of a 3D model based on the feature distance analysis of its sample views. Page et al. [PKS*03] defined a 2D/3D shape complexity based on the entropy of curvatures and Rigau et al. [RFS05] measured the inner and outer shape complexities based on mutual information.

Utilizing information theory related measurement to characterize the visual information features of a sample view of a 3D model has been recognized as an effective way, thus useful for 3D shape complexity measurement as well. Vázquez et al. [VFSH03] proposed viewpoint entropy to depict the amount of information a view contains and based on this, they developed a method to automatically find a set of best views with top view entropy values.

3. Viewpoint Entropy Distribution-Based View Clustering

In this paper, we propose a 3D visual complexity metric based on the viewpoint entropy distribution of a set of sample views of a 3D model. After that, we apply the 3D visual complexity metric into our sketch-based 3D model retrieval algorithm to decide the number of representative views (cluster centers) to represent a 3D model. Finally, based on the viewpoint entropy values of the sample views, a Fuzzy C-Means (FCM) algorithm is employed to select the assigned number of representative views for each model.

Viewpoint Entropy Distribution. We subdivide a regular icosahedron (denoted as L_0) n times based on the Loop subdivision algorithm and name the resulting shape as L_n. For each model, we sample a set of viewpoints by setting the cameras on the vertices of L_n. All the 3D models are first scaled into a unit sphere and orthogonal projection is applied during 3D rendering. We adopt the viewpoint entropy computation method in [TFTN05]. For a 3D model with m faces, the viewpoint entropy of a view is defined as follows,

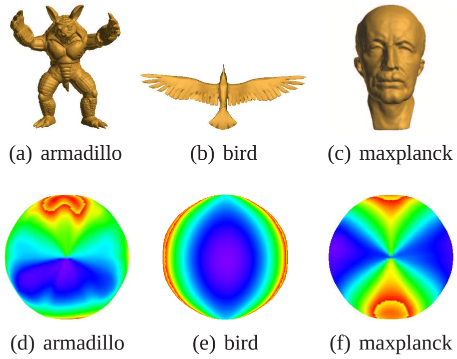
$$E = -\sum_{j=0}^{m} \frac{A_j}{S} log_2 \frac{A_j}{S} \tag{1}$$

where, A_j is the visible projection area of the j^{th} (j=1, 2, $\cdots$, m) face of a 3D model and A_0 is the background area. S is the total area of the window where the model is rendered: S=A_0+$\sum_{j=1}^{m} A_j$.

Figure 1 shows the viewpoint entropy distributions of three models using L_3 for view sampling and mapping their entropy values as colors on the surface of the spheres. We can see models with different complexity degrees have different types of entropy distribution. A visually complex model (e.g. armadillo and maxplanck) usually correspondingly has a more complicated pattern, and vice versa (e.g. bird). Motivated by this finding, we propose to measure the 3D visual complexity of a 3D model based on its viewpoint entropy distribution.

Viewpoint Entropy-Based 3D Visual Complexity. To assign the same number of representative views for each distinct class, we perform viewpoint entropy distribution analysis on a class-level and propose an entropy-based metric to measure the 3D visual complexity of a model.

(1) Class distribution analysis based on viewpoint entropy. This is to uncover the properties of entropy distribution based on class-level experiments on a dataset. As an example, we select the target 3D model dataset of Yoon et al.'s [YSSK10, LSG*12] sketch-based retrieval benchmark. It comprises 13 selected classes (260 models, 20 each) of AIM@ Shape Watertight Model Benchmark (WMB) dataset [VtH07]. One sample from each class is shown in Figure 2 (a). For each model, we adopt L_2 (162 views) for the viewpoint sampling and then compute the viewpoint entropy at each viewpoint. After that, we compute the mean and the standard deviation entropy values m and s among all the 162 views of the model. Finally, we average all of them over the 20 models for each class. Figure 2 (b) shows the entropy distributions of different classes and the analysis of our clustering results. As shown in the figure, viewpoint entropy reasonably reflects the semantic distances among different classes. For example, "bird", "plane" and "fish" all have "wings" and are also visually similar. "Human", "ant" and "chair" share the characteristic of elongate shapes. "Hand" and "teddy", "cup" and "table" have the following same properties: the areas of certain (usually the front, top and side) views are apparently larger than other views; most views of these classes have bigger projection areas than those of other classes, thus both their mean and standard deviation entropy values are larger. Both types are flat but have differences in thickness and thus we denote these classes as "thin & flat" and "thick & flat" types, respectively.


(a) armadillo (b) bird (c) maxplanck

(d) armadillo (e) bird (f) maxplanck

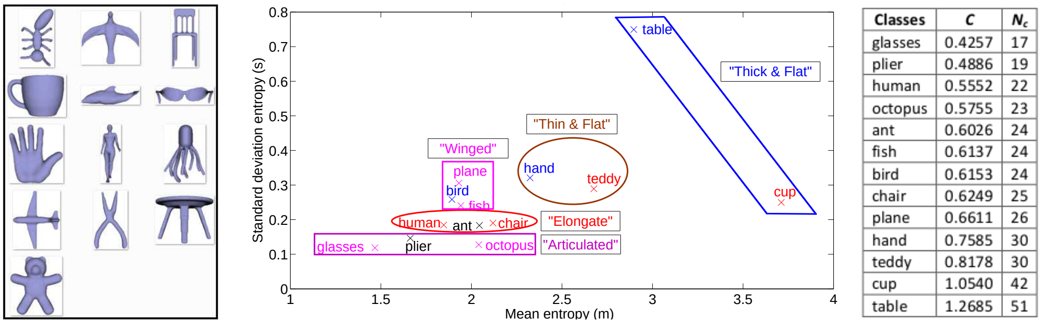
Figure 1: *Viewpoint entropy distribution examples: First row shows the models (in the original poses); Second row demonstrates the viewpoint entropy distributions of the models seen from the original poses. Viewpoint entropy is coded using HSV color model and smooth shading. Red: small entropy; green: mid-size entropy; blue: large entropy.*

(2) Entropy-based 3D visual complexity measures. Based on the above finding and analysis, to measure the 3D visual complexity C using m and s, we can adopt the following three approaches: angle $C = \frac{s}{m}$, area $C = s * m$, and Euclidean (D_2) distance,

$$C = \sqrt{\hat{s}^2 + \hat{m}^2} \tag{2}$$

where, $\hat{s}$ and $\hat{m}$ are the normalized s and m by their respective maximums over all the classes. For the angle and area metrics, normalization or not will have no impact on the ranking results. Different metrics may have different performance when applied on related applications. According to our experiments, for 3D model retrieval, Euclidean metric has the best performance in terms of both accuracy and robustness and we also adopt it in our retrieval algorithm. Figure 2 (c) sorts and lists the 3D visual complexity values of the 13 classes according to the D_2 distance metric (Eq. 2).

Viewpoint Entropy-Based Adaptive Views Clustering.



Classes	C	N_c
glasses	0.4257	17
plier	0.4886	19
human	0.5552	22
octopus	0.5755	23
ant	0.6026	24
fish	0.6137	24
bird	0.6153	24
chair	0.6249	25
plane	0.6611	26
hand	0.7585	30
teddy	0.8178	30
cup	1.0540	42
table	1.2685	51

(a) A model example per class

(b) Entropy distributions w.r.t classes and our annotation

(c) Visual complexity values and view numbers

Figure 2: *Viewpoint entropy distributions and numbers of representative views of different classes in the WMB 3D benchmark*

Utilizing the visual complexity value C (Eq. 2) of a model, we adaptively assign its number of representative feature views (outline views presented in Section 4) and perform view clustering to obtain the representative views, as follows.

(1) Sample views generation. We still adopt L_2 (162 viewpoints) for the feature views sampling. But considering the symmetry property of outline feature views rendered from two opposite viewpoints, we select half of them (within a semi-sphere, 81 views) to generate the sample views.

(2) Assign the number of representative views. We set the number of representative views N_c to be proportional to its visual complexity C.

$$N_c = \alpha \cdot C \cdot N_0 \tag{3}$$

where, N_0 is the total number of sample views in the sample view space and α is a constant. In our algorithm, N_0=81. Since we only consider half of the view space, correspondingly we set $\alpha = \frac{1}{2}$. And the corresponding numbers of representative views N_c for the 13 classes are listed in Figure 2 (c).

(3) Representative view clustering using Fuzzy C-Means [Bez81] clustering. For each sampled viewpoint, we use the viewpoint entropy value e of the rendered view together with its 3D coordinate (x, y, z) as its feature $E = (x, y, z, e)$. Then, based on a Fuzzy C-Means clustering algorithm, we cluster all the N_0 feature vectors into N_c clusters, each having a member function measuring the possibilities of the N_0 feature vectors belonging to the cluster. After that, we label each viewpoint to the cluster with the maximum member function value. Finally, for each cluster, we regard the viewpoint that is closest to the cluster, in terms of D_2 distance, as the representative view of the cluster.

4. Sketch-Based Retrieval Algorithm

Feature Views Generation. Considering the factors of robustness, effectiveness, and efficiency, we extract outline feature views for both 2D sketches and 3D models in our algorithm. We illustrate one example for each in Figure 3. For 3D, we first render silhouette views and then extract the outlines. For 2D, we also first generate its silhouette view mainly through a series of morphological operations: binaryzation, Canny edge detection, morphological closing, dilation to fill the gaps between sketch curves, and region filling.

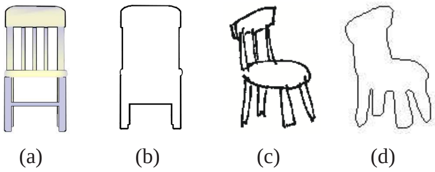


Figure 3: *2D/3D feature views generation examples. (a) curvature view of a chair model in PSB [SMKF04a]; (b) outline view; (c) a hand-drawn chair sketch in Yoon et al.'s [YSSK10] sketch benchmark dataset; (d) outline view of the chair sketch.*

Feature Extraction. Shape context matching [BMP02] is utilized to compute the distance between two outline feature views (one for sketch and one for model) during the retrieval stage. Shape context [BMP02] is a log-polar histogram and defines the relative distribution of other points with respect to a point. The default shape context definition partitions the surrounding area of a sample point of a 2D shape into 5 distance bins and 12 orientation bins, as shown in Figure 4 (c). Thus, the shape context is represented by a 5 × 12 matrix. In Figure 4, we show the shape context features of three points in two shapes. As shown in Figure 4 (d) and (e), different points have different shape context features in one shape and similar points in two similar shapes usually have similar shape context features, like Figure 4 (d) and Figure 4 (f). Shape context is scale and transformation-invariant but not rotation-invariant. To achieve the property of rotation invariance, in [BMP02] a relative frame is defined by adopting the local tangent vector at each point as the reference *x* axis for angle computation and we name it relative shape context. To encompass the differences in the camera up-vectors during the process of outline feature views generation, we extract the rotation-invariant relative shape context features for each feature view, as follows.

First, we uniformly sample 100 feature points for an outline in the feature view based on cubic B-Spline interpolation and uniform sampling. Then, we extract the relative shape context feature [BMP02] for each point. Finally, Jonker's LAP algorithm [JV87] is used to match the feature points of two outline feature views and the minimum matching cost is their distance.

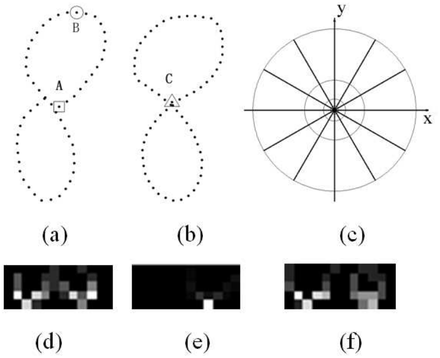


Figure 4: *Shape context examples. (d), (e), (f) are the shape context features of points A and B in (a) and point C in (b) respectively. The grayscale value of each element represents the percentage of other points in each bin. Darker means smaller.*

Online Retrieval. Given a query sketch and a 3D database, based on the representative views and their precomputed relative shape context features for each model, we develop the following online retrieval algorithm.

(1) Sketch feature extraction. We extract the outline feature view for the 2D sketch and compute its relative shape context features.

(2) Sketch-model distance vector computation. For each model, we perform shape context matching between the sketch and each representative view and regard the minimum matching cost as the sketch-model distance.

(3) Sketch-model distances sorting and output. We sort all the sketch-model distances in an ascending order and list the corresponding models accordingly.

5. Experimental Results and Discussion

In this section, we test our approach on two latest sketch-based 3D model retrieval benchmark datasets: Yoon et al.'s Benchmark [YSSK10, LSG*12] and SHREC'13 Sketch Track Benchmark [LLG*13]. To comprehensively evaluate the retrieval performance, we select six performance metrics: Precision-Recall (PR) diagram, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-Measure (E) and Discounted Cumulative Gain (DCG) [SMKF04a]. We denote our view clustering-based retrieval algorithm as "SBR-VC".

5.1. Yoon Et Al.'s Benchmark

In this section, we conduct evaluative and comparative experiments based on Yoon et al.'s sketch-based 3D model retrieval benchmark [YSSK10, LSG*12]. We have introduced its target dataset in Section 3 and Figure 2 (a). The query set contains 250 hand-drawn sketches for the 13 classes, each containing 17∼21 sketches. One example per class is shown in Figure 5.

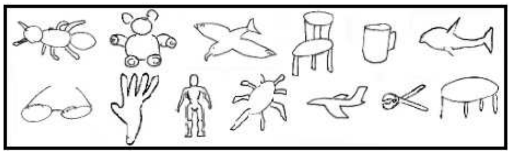


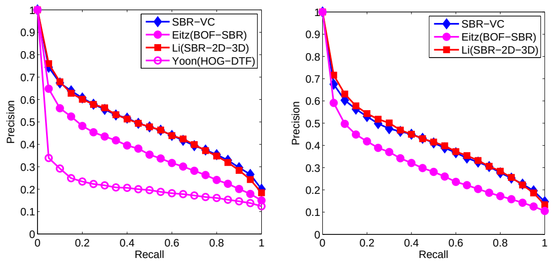
Figure 5: *Example 2D sketch per class in the query set [YSSK10, LSG*12].*

Overall Performance. To evaluate our algorithm and compare it with other approaches on a database level, we test our retrieval algorithm presented in Sections 3∼4 on the 250 sketches of Yoon et al.'s benchmark. Its average performance is compared with Yoon et al.'s [YSSK10, LSG*12] HOG-DTF approach and other two state-of-the-art algorithms reported in the SHREC'12: Sketch-Based 3D Shape Retrieval Track [LSG*12]: Li and Johan's SBR-2D-3D algorithm [LSG*12, LJ12] and Eitz et al.'s BOF-SBR approach [LSG*12, EHBA10, EHBA11]. Figure 6 (a) and Table 1 show their Precision-Recall diagram and other comparison results, respectively.

Table 1: *Other performance metrics comparison between our method and SBR-2D-3D, BOF-SBR and HOG-DTF.*

Method	NN	FT	ST	E	DCG
SBR-VC	0.664	0.427	0.587	0.413	0.730
BOF-SBR	0.532	0.339	0.497	0.338	0.662
SBR-2D-3D	0.688	0.415	0.581	0.411	0.731
HOG-DTF	0.220	0.167	0.286	0.182	0.513

As can be seen, our retrieval performance is apparently better than HOG-DTF and BOF-SBR. It also achieves similar performance as SBR-2D-3D algorithm (the top search algorithm reported in SHREC'12 Track [LSG*12]). However, SBR-2D-3D algorithm involves complicated, time-consuming "View Context" [LJ10] pre-computation and 2D



(a) Yoon et al.'s benchmark (b) Extended Yoon et al.'s benchmark

Figure 6: *Precision-Recall diagram performance comparisons on Yoon et al.'s and Extended Yoon et al.'s benchmarks between our method and other state-of-the-art algorithms.*

sketch to 3D model alignment is a necessary step. In contrast, our viewpoint entropy-based approach can achieve the same search accuracy without performing a 2D sketch-3D model alignment (as shown in Figure 6). Therefore, our view selection approach is effective, but more simple, efficient, and much faster in computation (at least *m*-1 times faster than the 2D-3D alignment method of SBR-2D-3D, where *m* is the number of base views and is set as 21 in Eq. 9 of [LJ12]). The major reason is summarized as follows.

To decide the candidate views for each 3D model, SBR-2D-3D needs to compute the View Context feature for each sample view, which is a 21×81 dimensional vector (21 is the number of base views and 81 is the number of sample views). Therefore, for each model we need to render 21x81 silhouette views and extract the ZFEC hybrid features [LJ12] while our SBR-VC algorithm only need render 81 color-coded sample views and computes their viewpoint entropy values. The total time needed for rendering a sample view of a model and computing its viewpoint entropy value is much less than the time needed for rendering and extracting the ZFEC hybrid features of a silhouette view. Thus, our candidate views selection approach is at least 20 times faster than the 2D-3D alignment method of SBR-2D-3D.

We also believe our approach is more promising to achieve even better performance by modifying the metric definition and incorporating the semantic distances among classes during the view number assignment.

Scalability Robustness. To evaluate the scalability property of our algorithm, we perform the same experiment as that in Li et al. [LSG*12] which uses the "Extended" version of the target dataset: adding the remained 140 models (7 classes, each with 20 models) of the WMB database. These irrelevant models are regarded as noise to increase the challenge of retrieval. Figure 6 (b) and Table 2 compare their performance again. We can see that our algorithm also shows a comparable robustness as SBR-2D-3D with respect to scalability and the performance gaps among different approaches are similar as those on the Yoon et al.'s benchmark. The performance decreases of our SBR-VC algorithm in terms of FT and DCG are 12.9% and 6.6%, compared to the 18.0% and 7.3% drops for Eitz et al.'s BOF-SBR, and 10.6% and 5.3% decreases for Li and Johan's SBR-2D-3D. Thus, SBR-VC is comparable to SBR-2D-3D in terms of both overall performance and scalability.

Table 2: *Other performance metrics for the performance comparison on the Hand-drawn sketch queries and Extended version of target dataset.*

Method	NN	FT	ST	E	DCG
SBR-VC	0.576	0.372	0.519	0.360	0.682
BOF-SBR	0.460	0.278	0.412	0.281	0.614
SBR-2D-3D	0.628	0.371	0.520	0.364	0.692

In addition, unlike SBR-2D-SD, which need compute, save, or load the relative shape context features of all the sample views of the models in the target dataset, SBR-VC has less computation and memory cost. It only need calculate the shape context features of the selected candidate views for each model. Therefore SBR-VC has superior scalability than SBR-2D-3D, thus it can be easily scaled to a large scale sketch-based 3D model retrieval application.

5.2. SHREC'13 Sketch Track Benchmark

There is one limitation in Yoon et al.'s benchmark: it has a rather small number of either 2D sketches (250) or 3D models (260 or at most 400). To evaluate and solicit latest and state-of-the-art sketch-based retrieval algorithms on a large scale benchmark, Li et al. [LLG*13] built a new benchmark for the Shape Retrieval Contest (SHREC) 2013 Track on Large Scale Sketch-Based Retrieval. It is based on a large sketch recognition dataset built by Eitz et al. [EHA12] and the Princeton Shape Benchmark (PSB) benchmark [SMKF04b]. The SHREC'13 Sketch Track Benchmark has 7200 hand-drawn sketches in total, equally divided into 90 classes, each with 80 sketches. 1258 relevant 3D models are selected from the PSB benchmark to form the target 3D dataset. Some example sketches and their relevant 3D model examples are shown in Figure 7. To accustom to machine learning-based retrieval algorithms, they also build training and testing datasets by randomly selecting 50 sketches per class for training and the remained 30 sketches for testing. The complete target model dataset is used as a whole for both training and testing purpose.

We have tested our SBR-VC algorithm on the "Training", "Testing", and "Complete" benchmark datasets and compared it with other participating approaches in the SHRSC'13 Sketch Track, which include Li (SBR-2D-3D), Saavedra (FDC), and Aono (EFSD). For SBR-VC, we tested different number of sampling points, denoted by NUM, such as 50 or 100. To accustom to the large-scale retrieval for efficiency consideration, we keep less representative views by

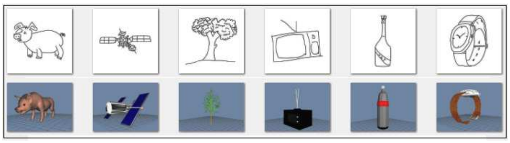


Figure 7: *Example 2D sketches (1st row) and relevant 3D models (2nd row) in the SHREC'13 Sketch Track Benchmark.*

setting $\alpha = \frac{1}{6}$. We add the parameter settings into the algorithm name, such that SBR-VC-NUM-50 means the NUM is set to 50, and so on so forth.

Figure 8 and Table 3 compare their performance in terms of 6 performance metrics. It can be observed that both SBR-VC and SBR-2D-3D achieve much better accuracy than FDC and EFSD. SBR-VC-NUM-100 keeps more sampling points but less representative views because of the change in N_c. It is comparable to SBR-2D-3D-NUM-50 in terms of retrieval efficiency. But it has a superior retrieval performance than SBR-2D-3D. For example, considering the retrieval on the complete benchmark, the performance increases are 21.1%, 22.8%, 19.2%, 14.9% and 6.1% in terms of NN, FT, ST, E, and DCG respectively. It should be pointed out that all the retrieval performance values are not high is mainly because the benchmark is really challenging. For the queries, a lot of variations exist in the sketches for each class, while for the target models, many classes only contain few models. Accurately retrieving those classes of models is difficult and the performance evaluation values on these classes are usually much lower, especially on NN, FT, and ST values. Even though, our algorithm still shows its superior performance compared to other state-of-the-art approaches, such as SBR-2D-3D, FDC, and EFSD. This also demonstrates the robustness of our approach.

6. Conclusions and Future Work

We have presented a sketch-based 3D model retrieval algorithm by first clustering the sampled views of a 3D model into adaptively specified number of representative feature views and then employing shape context matching to compare the sketch with each representative feature view. It is more reasonable to sample an appropriate number of representative views for a 3D model according to its visual complexity. A 3D visual complexity metric is first proposed based on the viewpoint entropy distribution of a set of sample views. Based on the proposed 3D visual complexity metric, the number of representative feature views can be adaptively assigned during our view clustering process.

Experiments on both small scale and large scale benchmarks have demonstrated the effectiveness of our retrieval algorithm, which shows better performance than Eitz et al.'s BOF-SBR [LSG*12, EHBA10, EHBA11], Yoon et al.'s HOG-DTF [LSG*12, YSSK10], Saavedra's FDC, and Aono et al.'s EFSD approaches and it is comparable to Li et al.'s SBR-2D-3D [LJ12] in terms of both overall performance and scalability robustness while our viewpoint entropy-based approach is simpler and more efficient. Thus, the proposed approach has achieved similar accuracy as the 2D sketch-3D model alignment by adaptive view clustering based on the viewpoint entropy distribution and 3D visual complexity analysis.

Table 3: *Other performance metrics for the performance comparison on the SHREC'13 Sketch Track Benchmark.*

Method	NN	FT	ST	E	DCG
Training dataset					
SBR-VC-NUM-100	0.160	0.097	0.149	0.085	0.349
SBR-VC-NUM-50	0.131	0.082	0.130	0.076	0.333
Li (SBR-2D-3D-NUM-50)	0.133	0.080	0.126	0.075	0.330
Saavedra (FDC)	0.051	0.039	0.069	0.041	0.279
Aono (EFSD)	0.024	0.019	0.038	0.020	0.241
Testing dataset					
SBR-VC-NUM-100	0.164	0.097	0.149	0.085	0.348
SBR-VC-NUM-50	0.132	0.082	0.131	0.075	0.331
Li (SBR-2D-3D-NUM-50)	0.132	0.077	0.124	0.074	0.327
Saavedra (FDC)	0.053	0.038	0.068	0.041	0.279
Aono (EFSD)	0.023	0.019	0.036	0.019	0.240
Complete benchmark					
SBR-VC-NUM-100	0.161	0.097	0.149	0.085	0.349
SBR-VC-NUM-50	0.131	0.082	0.130	0.076	0.332
Li (SBR-2D-3D-NUM-50)	0.133	0.079	0.125	0.074	0.329
Saavedra (FDC)	0.052	0.039	0.069	0.041	0.279
Aono (EFSD)	0.023	0.019	0.037	0.019	0.241

In the future work, we plan to test the performance of our algorithm by assigning different number of representative views to the models even within one class based on their complexity values. We also plan to apply our 3D visual complexity metric into other related applications.

Acknowledgments

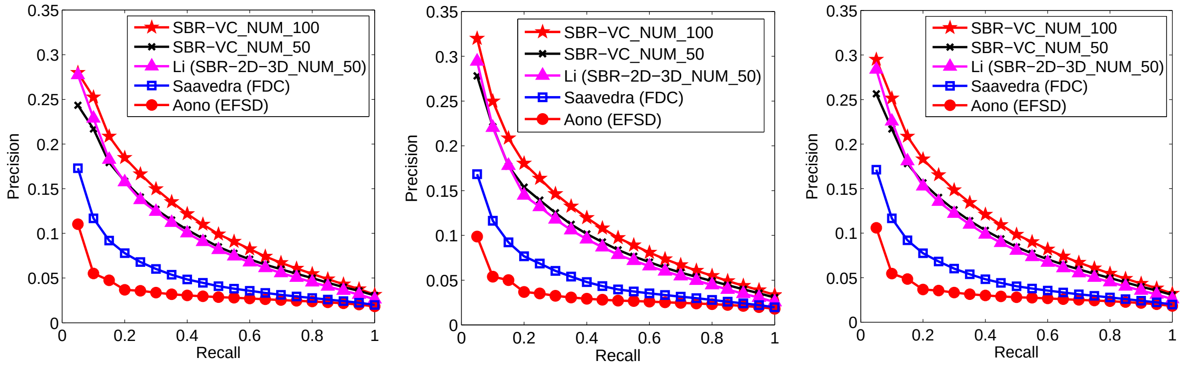
This work by Bo Li and Yijuan Lu is supported by the Texas State University Research Enhancement Program (REP), Army Research Office grant W911NF-12-1-0057, and NSF CRI 1058724 to Dr. Yijuan Lu.

References

[AVD07] Ansary T. F., Vandeborre J.-P., Daoudi M.: 3D-model search engine from photos. In *CIVR* (2007), pp. 89–92. 2

[Bez81] Bezdek J. C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981. 1, 4

[BMP02] Belongie S., Malik J., Puzicha J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 4 (2002), 509–522. 2, 4, 5



(a) Training dataset (b) Testing dataset (c) Complete dataset

Figure 8: *Precision-Recall diagram performance comparisons on different datasets of the SHREC'13 Sketch Track Benchmark between our method SBR-VC and other state-of-the-art algorithms.*

[EHA12] Eitz M., Hays J., Alexa M.: How do humans sketch objects? *ACM Trans. Graph. 31*, 4 (2012), 44. 6

[EHBA10] Eitz M., Hildebrand K., Boubekeur T., Alexa M.: Sketch-based 3D shape retrieval. In *SIGGRAPH Talks* (2010). 2, 5, 7

[EHBA11] Eitz M., Hildebrand K., Boubekeur T., Alexa M.: Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE Trans. Vis. Comput. Graph. 17*, 11 (2011), 1624–1636. 1, 2, 5, 7

[FMK*03] Funkhouser T. A., Min P., Kazhdan M. M., Chen J., Halderman J. A., Dobkin D. P., Jacobs D. P.: A search engine for 3D models. *ACM Trans. Graph. 22*, 1 (2003), 83–105. 1

[JV87] Jonker R., Volgenant A.: A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing 38*, 4 (1987), 325–340. 5

[Kan08] Kanai S.: Content-based 3D mesh model retrieval from hand-written sketch. *Interact. Des. and Manuf. 2*, 2 (2008), 87–98. 1

[LJ10] Li B., Johan H.: View context: A 3D model feature for retrieval. *In: S. Boll et al. (eds.): MMM 2010, LNCS, Springer, Heidelberg 5916* (2010), 185–195. 2, 5

[LJ12] Li B., Johan H.: Sketch-based 3D model retrieval by incorporating 2D-3D alignment. *Multimedia Tools and Applications* (2012), 1–23 (online first version). 2, 5, 6, 7

[LLG*13] Li B., Lu Y., Godil A., Schreck T., Aono M., Johan H., Saavedra J. M., Tashiro S.: SHREC'13 track: Large scale sketch-based 3D shape retrieval. In *3DOR* (2013), pp. 1–9. URL: `http://www.itl.nist.gov/iad/vug/sharp/contest/2013/SBR/`. 5, 6

[LSG*12] Li B., Schreck T., Godil A., Alexa M., Boubekeur T., Bustos B., Chen J., Eitz M., Furuya T., Hildebrand K., Huang S., Johan H., Kuijper A., Ohbuchi R., Richter R., Saavedra J. M., Scherer M., Yanagimachi T., Yoon G.-J., Yoon S. M.: SHREC'12 track: Sketch-based 3D shape retrieval. In *3DOR* (2012), pp. 109–118. 1, 2, 3, 5, 6, 7

[MA05] Mokhtarian F., Abbasi S.: Robust automatic selection of optimal views in multi-view free-form object recognition. *Pattern Recognition 38*, 7 (2005), 1021–1031. 2

[PKS*03] Page D. L., Koschan A., Sukumar S. R., Roui-Abidi B., Abidi M. A.: Shape analysis algorithm based on information theory. In *ICIP (1)* (2003), pp. 229–232. 2

[RFS05] Rigau J., Feixas M., Sbert M.: Shape complexity based on mutual information. *SMA 0* (2005), 357–362. 2

[Ros05] Rossignac J.: Shape complexity. *The Visual Computer 21*, 12 (2005), 985–996. 2

[SBWS11] Saleem W., Belyaev A. G., Wang D., Seidel H.-P.: On visual complexity of 3D shapes. *Computers & Graphics 35*, 3 (2011), 580–585. 2

[SMKF04a] Shilane P., Min P., Kazhdan M., Funkhouser T.: The Princeton shape benchmark. In *SMA* (2004), pp. 167–178. 2, 4, 5

[SMKF04b] Shilane P., Min P., Kazhdan M. M., Funkhouser T. A.: The princeton shape benchmark. In *SMI* (2004), IEEE Computer Society, pp. 167–178. 6

[SXY*11] Shao T., Xu W., Yin K., Wang J., Zhou K., Guo B.: Discriminative sketch-based 3D model retrieval via robust shape matching. *Comput. Graph. Forum 30*, 7 (2011), 2011–2020. 1, 2

[TFTN05] Takahashi S., Fujishiro I., Takeshima Y., Nishita T.: A feature-driven approach to locating optimal viewpoints for volume visualization. In *IEEE Visualization* (2005), p. 63. 3

[TWLB09] Ta A.-P., Wolf C., Lavoue G., Baskurt A.: 3D object detection and viewpoint selection in sketch images using local patch-based Zernike moments. In *Proceedings of the 2009 Seventh International Workshop on Content-Based Multimedia Indexing* (2009), pp. 189–194. 1

[VFSH03] Vázquez P.-P., Feixas M., Sbert M., Heidrich W.: Automatic view selection using viewpoint entropy and its applications to image-based modelling. *Comput. Graph. Forum 22*, 4 (2003), 689–700. 2

[VtH07] Veltkamp R. C., ter Haar F. B.: *SHREC 2007 3D Retrieval Contest.* Technical Report UU-CS-2007-015, Department of Information and Computing Sciences, Utrecht University, 2007. 3

[YSSK10] Yoon S. M., Scherer M., Schreck T., Kuijper A.: Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours. In *ACM Multimedia* (2010), pp. 193–200. 1, 2, 3, 4, 5, 7