Multimed Tools Appl manuscript No.
(will be inserted by the editor)

Sketch-based 3D model retrieval utilizing adaptive view clustering and semantic information

Bo Li · Yijuan Lu · Henry Johan · Ribel Fares

Received: date / Accepted: date

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, human computer interaction, 3D animation, game design, and etc. In this paper, our target is to significantly improve the current sketch-based 3D retrieval performance in terms of both accuracy and efficiency. We propose a new sketch-based 3D model retrieval framework by utilizing adaptive view clustering and semantic information. It first utilizes a proposed viewpoint entropy-based 3D information complexity measurement to guide adaptive view clustering of a 3D model to shortlist a set of representative sample views for 2D-3D comparison. To bridge the gap between the query sketches and the target models, we then incorporate a novel semantic sketch-based search approach to further improve the retrieval performance. Experimental results on several latest benchmarks have evidently demonstrated our significant improvement in retrieval performance.

Keywords Sketch-based 3D model retrieval · Semantic information · Adaptive view sampling · Viewpoint entropy · Sketch recognition · Machine learning · Shape context matching

B. Li
School of Computing, University of Southern Mississippi, 730 E Beach Blvd, Long Beach, USA 39560
e-mail: bo.li@usm.edu

Y. Lu, R. Fares
Department of Computer Science, Texas State University, 601 University Dr, San Marcos, USA 78666
e-mail: lu@txstate.edu, ribelfares@gmail.com

H. Johan
Fraunhofer IDM@NTU, Nanyang Technological University, 50 Nanyang Ave, Singapore 639798
e-mail: hjohan@fraunhofer.sg

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. Sketch-based 3D model retrieval attracts a lot of attention due to its promising application potentials in sketch-based 3D modeling and recognition, human computer interaction, 3D animation, game design, etc.

Recently, quite a few sketch-based 3D model retrieval algorithms [36] [6] [21] 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, independent to the complexity of the model. For example, Yoon et al. [36] compared a sketch with 14 sample views for each model. However, these sampling strategies cannot guarantee that the extracted sample views are appropriate and representative enough to depict any 3D model since they do not consider the complexities of different models. In fact, there is no need to sample and compare 13 or even more views for a simple model, such as a book or a wheel, while more views should be sampled for a complicated model. That is, we need an adaptive sampling strategy.

Motivated by the above finding, we propose to sample different number of representative views for a 3D model to compare with a 2D sketch according to the 3D information complexity of the 3D model [18] (Section 3), which is a novel 3D complexity metric proposed by us based on the viewpoint entropy distribution of a set of uniformly sampled views of a 3D model. The metric is further utilized to assign the number of cluster centers (representative views) during the Fuzzy C-Means (FCM) [4] clustering of the sampled views based on their viewpoint entropy values. After that, an efficient shape context matching [3] algorithm is used for the parallel 2D-3D matching between the 2D sketch and the representative views of all the 3D models. An overview of our 2D-3D comparison algorithm is shown in Fig. 1 (a). We have found that on the one hand our adaptive view clustering approach improves the effectiveness of sample views for a 3D model, and hence the retrieval accuracy, on the other hand it also significantly reduces the computational complexity of retrieval process due to less sample views to compare, thus making it more applicable for large-scale 3D model retrieval applications.

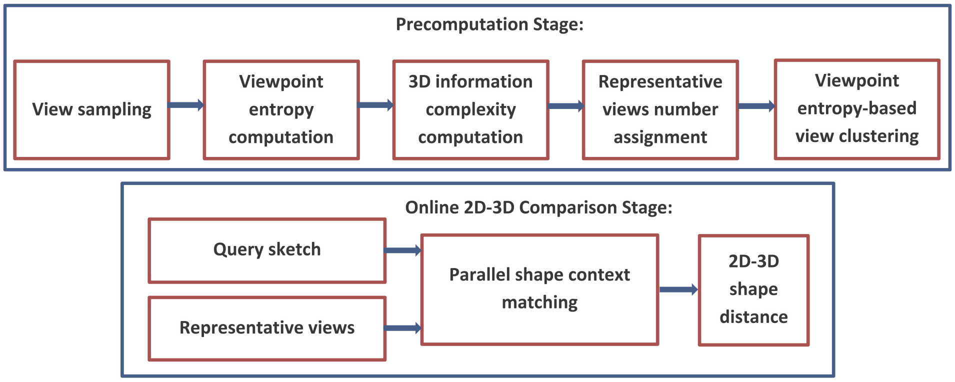
We found that it is still a very challenging task for existing algorithms to achieve outstanding performance, in terms of both effectiveness and efficiency, especially when applied on a large-scale sketch-based retrieval scenario. This is because there exists a big semantic gap between user sketches and 3D models in the database: iconic representations of a sketch and accurate representation of a 3D model for the same object. That is, human sketches always have arbitrary styles, iconic representations in 2D space, high-level abstraction, and drastic simplification, which bring a lot of difficulties in sketch description and representation. The 3D model of an object is generally an accurate representation of its geometry information. Such big semantic gap makes the search based on a direct 2D-3D comparison suffer low accuracy and high computational cost if we sample views densely. However, all the existing methods fail to bridge the semantic gap without considering the semantic information of either the sketch queries or the target 3D models.

Many professional or generic 3D model databases, such as Engineering Shape Benchmark (ESB) [11], Bonn University Architecture Databases Benchmark [35], Princeton Shape Benchmark (PSB) [30], Konstanz 3D Model Benchmark CCCC [33] and Shape Retrieval Contest (SHREC) datasets [1], as well as other datasets mentioned in [19] are already classified. In fact, generally, most, though not all, available 3D model databases have class information, which can be utilized to improve the related practical retrieval performance. In other words, it is common to encounter such practical retrieval scenario or application where the database already has contained class information. One such example is the latest largest shape repository ShapeNet [29]. Therefore, by leveraging the class information, we should be able to achieve a lot better retrieval. This practical retrieval scenario serves as one of our goals, as well.

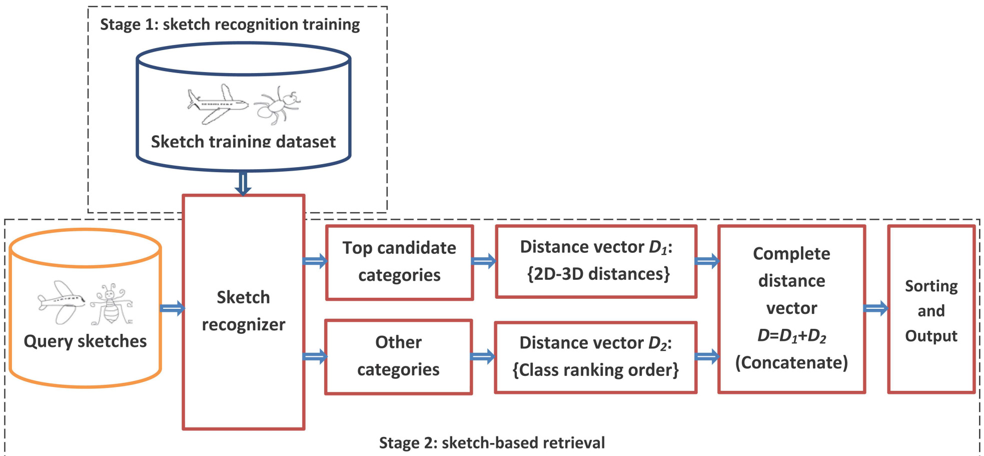
Considering the above findings and motivation, in this paper, we further propose a novel semantic sketch-based 3D model search approach [15] (Section 4) to bridge the semantic gap. Our approach tries to understand the semantic meaning users are expressing through their sketches before searching the corresponding 3D models. By building an intelligent sketch recognizer, our approach can first predict the potential semantics of the user sketch. Then by searching the 3D models in the predicted semantic categories (same as the "classes" in a 3D model dataset; we use them interchangeably in the paper), the best matchings can be found. We design the approach of semantic sketch-based retrieval as shown in Fig. 1 (b). The experimental results demonstrate that the semantic approach achieves significant improvements in both search accuracy and efficiency.

Our paper is an extended version based on our prior work in [15] and [18]. The main contributions introduced in this paper are summarized below:

- We devise a new sketch-based 3D model retrieval framework which integrates two components. One component is an adaptive view clustering method to deal with appropriate and meaningful 3D view sampling based on the viewpoint entropy measurement. The other component is a semantic sketch-based retrieval approach to bridge the gap between the iconic representations of query sketches and the accurate representations of 3D models.
- We quantitatively study and formulate the 3D complexity analysis of a 3D model problem. Based on this, we propose a reasonable and effective 3D information complexity metric by measuring the information theory-related viewpoint entropy. It has been successfully applied into the 2D-3D comparison algorithm to adaptively decide the number of representative views for each 3D model. In addition, our work provides a feasible direction for the 3D complexity research and also validates a practicable path for view clustering-based 3D retrieval.
- We propose a novel semantic sketch-based 3D model search algorithm to bridge the semantic gap between user sketches and 3D models. It also opens broad research possibility for semantic sketch-based 3D model retrieval and annotation. In addition, we develop an intelligent sketch recognizer through supervised learning to appropriately capture the semantic meanings of users' sketches.
- We perform a set of comparative experiments on four latest sketch-based 3D model retrieval benchmarks and demonstrate the outperforming performance of our framework and its promising application potential for sketch-based 3D model retrieval.



(a) Our 2D-3D comparison method utilizing adaptive view clustering



(b) Our semantic sketch-based retrieval approach

Fig. 1 Our sketch-based 3D model retrieval framework which utilizes adaptive view clustering and semantic information. In (b), D_1 is computed based on the method presented in (a).

The organization of the paper is as follows. We review related work in Section 2. The view clustering-based 2D-3D comparison method and the semantic retrieval approach are respectively presented in Sections 3 and 4. Section 5 performs comparative experiments on four benchmarks while Section 6 draws the conclusions and lists several future work.

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.

2.1.1 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. [36] 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. Eitz et al. [6] [21] adopted a Bag-of-Features (BoF) framework and extracted Histogram of Gradient (HOG) local features for the subdivided patches of both sketches and *102* sample views of each model. In [7], Eitz et al. further proposed a feature named Gabor local line-based feature (GALIF) to deal with sketch-based 3D retrieval. Li and Johan [13] [21] proposed a sketch-based 3D model retrieval algorithm named SBR-2D-3D 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" [13] before 2D-3D matching. Their 2D-

3D matching algorithm is based on relative shape context matching [3].

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.

2.1.2 Using clustered views

Compared with the approaches based on predefined views, much less research work has been done for the strategy based on view clustering. Mokhtarian and Abbasi [22] 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. [2] 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 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 3D complexity

Geometrical shape complexity approaches have been reviewed by Rossignac [27] 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. [28] measured the visual complexity of a 3D model based on the feature distance analysis of its sample views. Page et al. [25] defined a 2D/3D shape complexity based on the entropy of curvatures.

Utilizing information theory related measurement to characterize the information that a sample view of a 3D model contains has been recognized as an effective way, thus useful for 3D complexity measurement as well. Vázquez et al. [32] 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 viewpoint entropy values. Our proposed 3D information complexity measurement is just based on viewpoint entropy. Compared with the 3D shape complexity metrics mentioned above, it is more efficient to compute and direct to understand, and also has its solid root in information theory.

2.3 Semantic 3D model retrieval

Without considering the semantic information of either the sketch queries or the target 3D models, all the sketch-based 3D model retrieval methods mentioned before fail to bridge the semantic gap between the sketches and 3D models. Our proposed semantic sketch-based 3D model retrieval approach makes the first attempt to use semantic classification information to reduce the semantic gap, as well as to adequately utilize the better-performing global feature matching to improve the retrieval efficiency.

According to our knowledge, in the field of 3D model retrieval, semantic 3D model retrieval techniques have been utilized only for the Query-by-Model scenario where there is no semantic gap between the queries and targets. However, in our semantic sketch-based retrieval approach, as an example of Query-by-Sketch scenario, it is utilized to bridge the semantic gap between the input sketches and output 3D models.

For example, Ohbuchi et al. [24] proposed a dimension reduction approach based on semi-supervised learning approach and they tried to incorporate the semantic class information into Query-by-Model retrieval applications. Recently, Gong et al. [9] proposed a semantic 3D signature based on semantic attributes, such as symmetry, circularity, and rectilinearity. Hou et al. [10] proposed a 3D model retrieval algorithm for 3D model queries based on semantic labeling. First, they performed SVM-based clustering on the target 3D model database and then classified the query into a labeled class. Finally, they ranked all the models in the relevant class by employing a feature vector selection approach. Nie et al. [23] proposed an image search algorithm based on a probabilistic network to model the relatedness between images by considering three layers of relationships: semantic-level, cross-modality level and visual-level.

In summary, semantic information has been used in 3D model retrieval but purely for the performance improvement in a retrieval scenario where there is no semantic gap between the query and targets and thus it is optional to consider semantic information. However, it can be considered as a necessity in order to bridge the existing big gap between the 2D sketches and 3D models in the scenario of sketch-based 3D model retrieval.

3 2D-3D comparison method using adaptive view clustering

Different models have different complexities, thus there is no need to keep the same number of representative views for each model to compare with a sketch. In this section, we propose a 3D information complexity metric based on the viewpoint entropy distribution of a set of sample views of a 3D model. After that, we apply the 3D information complexity metric into our 2D-3D comparison method 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.

3.1 Viewpoint entropy distribution-based view clustering

3.1.1 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 [31]. For a 3D model with p faces, the viewpoint entropy e of a view is defined as follows,

$$e = -\sum_{j=0}^{p} \frac{A_j}{A} log_2 \frac{A_j}{A} \tag{1}$$

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

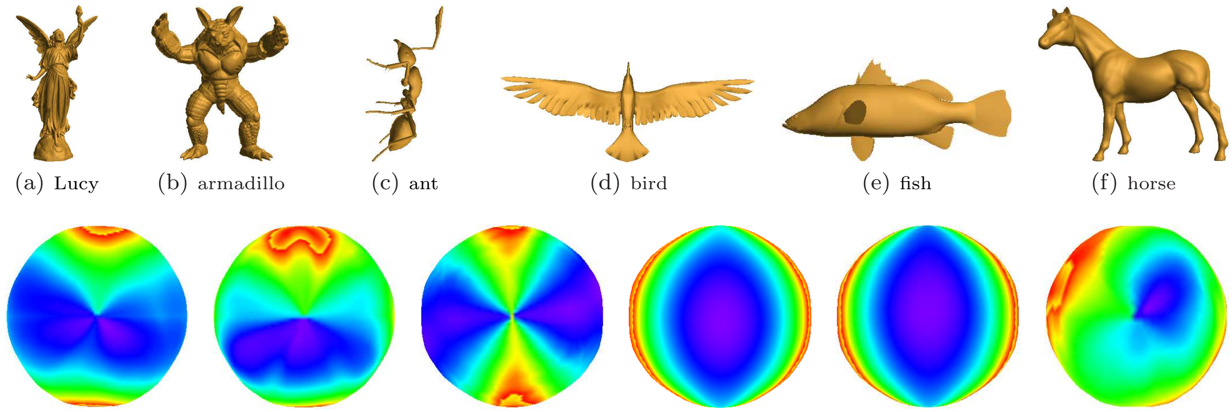
Fig. 2 shows the viewpoint entropy distributions of three models using L_3 for view sampling and mapping their entropy values as colors on the surface of the spheres. We can see that models with similar complexities have similar entropy distribution patterns, while entropy distribution patterns of models with different complexities are also distinctive. For example, as expected, the Lucy (a model which is close to the human class in Fig. 2 (a)) and armadillo models have very similar entropy distribution patterns, which only differ slightly from the pattern of an ant. Similarly, the entropy distribution patterns of the bird and fish models are also very close. However, the distribution pattern of the horse model is quite different from all the five models mentioned above. These also match our expectation. The fact behind this is that classes (i.e. Lucy and armadillo, bird and fish) sharing similar 3D complexity have similar viewpoint entropy distribution patterns, while classes (i.e. horse versus bird/ant/fish/human) with different complexities also differ in their entropy distribution patterns. Motivated by this finding, we propose to measure the 3D complexity of a 3D model based on its viewpoint entropy distribution.

3.1.2 Viewpoint entropy-based 3D information complexity

To assign the same number of representative views to the models belonging to the same class, we perform viewpoint entropy distribution analysis on a class-level and propose an entropy-based metric to measure the 3D information 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 [36] [21] sketch-based retrieval benchmark. It comprises 13 selected classes (260 models, 20 each). One sample from each class is shown in Fig. 3 (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 (measures the average amount of information that a view of the model contains) and s (measures the information difference in different views of the model) among all the 162 views of the model. Finally, we average all of them over the 20 models for each class. Fig. 3 (b) shows the entropy distributions (in terms of m and s) of different classes and the analysis of our clustering results. As shown in the figure, viewpoint entropy reasonably reflects the 3D complexity similarities among different classes and matches the viewpoint distribution patterns shown in Fig. 2 as well. 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" have the following same properties: the areas of certain (usually the front, top and side) views are apparently smaller than other views; most views of this class have bigger projection areas than those of other classes, thus both their mean and standard deviation entropy values are larger. This type has at least one relatively flat face, thus we denote this class as "Flat" type.

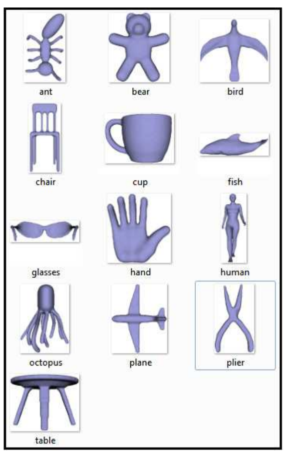
(2) Entropy-based 3D information complexity measures. Based on the above finding and anal-



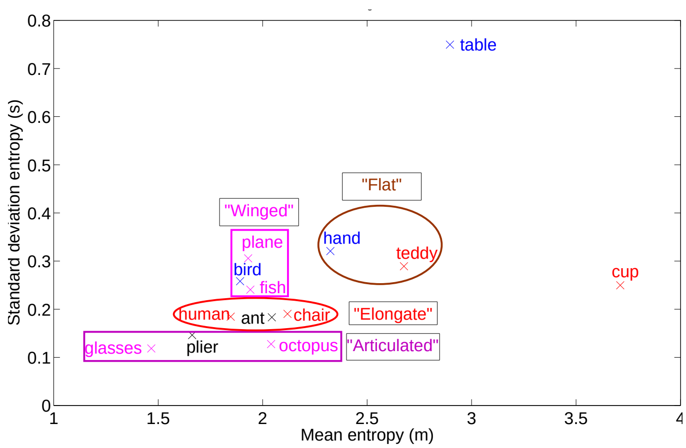
(a) Lucy (b) armadillo (c) ant (d) bird (e) fish (f) horse

(g) Lucy (h) armadillo (i) ant (j) bird (k) fish (l) horse

Fig. 2 Viewpoint entropy distribution examples: the first row shows six models; the second row demonstrates the viewpoint entropy distributions of the corresponding models with respect to their displayed viewpoints. Viewpoint entropy is coded using HSV color model and smooth shading. Red: small entropy; green: mid-size entropy; blue: large entropy.



(a)



(b)

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
Average	0.6970	27

(c)

Fig. 3 Viewpoint entropy distributions and numbers of representative views of different classes in the WMB 3D benchmark: (a) A model example per class; (b) Entropy distributions w.r.t classes and our annotation; (c) 3D information complexity C values and view numbers N_c.

ysis, to define a metric to measure the 3D information complexity C for a class by incorporating both the average value m and the deviation value s, we have the following three basic geometric measurements: angle $C_{an} = \frac{s}{m}$, area $C_{ar} = s * m$ and magnitude (D_2),

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

For the magnitude metric, $\widehat{s}$ and $\widehat{m}$ are the normalized s and m by their respective maximums over all the classes, which gives that $\widehat{s} \in [0,1]$ and $\widehat{m} \in [0,1]$. Thus, the magnitude metric $C \in [0, \sqrt{2}]$, which is to measure the size of the entropy-based 3D information complexity feature vector $< \widehat{s}, \widehat{m} >$. 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, such as shape compression, classification, recognition and retrieval. According to our experiments, for 3D model retrieval, the magnitude metric, which equally combines both values to measure the size information of the feature vector (m, s), has the best performance in terms of retrieval accuracy and we also adopt it in our 2D-3D comparison algorithm. We also have found that it has better class differentiation ability than other metrics such as standard deviation s only. Fig. 3 (c) sorts and lists the 3D information complexity values of the 13 classes according to the D_2 distance metric (Eq. 2).

3.1.3 Viewpoint entropy-based adaptive views clustering

Utilizing the 3D information complexity value C (Eq. 2) of a model, we adaptively assign its number of representative feature views (outline views presented in Section 3.2) and perform view clustering to obtain the representative views.

(1) Sample views generation. Similar as Section 3.1.2, 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 hemisphere, 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 3D information 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, $\alpha \in [0, \frac{1}{\sqrt{2}}]$. Statistically, the average number of representative views N_a over all the classes in a dataset will be close to $\frac{\alpha \cdot N_0}{\sqrt{2}}$. In our algorithm, N_0=81 and in default we set $\alpha = \frac{1}{2}$. The corresponding numbers of representative views N_c for the 13 classes are listed in Fig. 3 (c). In fact, to meet the speed and accuracy requirements of different applications, we can easily adjust the number of representative views by simply assigning different values to α.

We need to mention that rather than measuring the 3D geometrical complexity of a 3D model (i.e., the complexity in the structures and components of a 3D model like a table or a chair), our definition of 3D information complexity is based on the viewpoint entropy distribution of a 3D model, which incorporates the mean and variation in the entropy information of a set of sample views of a model. It can be found from the figure that the table class has much bigger entropy variation than the chair class, though the table and chair classes share similar mean entropy information in their views.

(3) Representative view clustering using Fuzzy C-Means [4] 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 $E = (x, y, z, e)$. The main reason for this design is that for a 3D model the viewpoint entropy values of its neighboring viewpoints are close to each other as well. We want to select only one viewpoint as a representative view to represent the views that are close to each other. Therefore, for all the sampled viewpoints of a 3D model, we consider both their entropy values and locations during the view clustering. Then, based on a Fuzzy C-Means clustering algorithm, we cluster all the N_0 feature vectors into N_c clusters, each having a membership 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 membership function value. Finally, for each cluster, we regard the viewpoint that is closest to the center of the cluster, in terms of D_2 distance, as the representative view of the cluster.

3.2 View clustering-based 2D-3D comparison method

3.2.1 Feature views generation

Considering the factors of robustness w.r.t different types of sketch queries, effectiveness and efficiency performance, which also have been demonstrated by our comparative experimental results, we extract outline feature views for both 2D sketches and 3D models in our algorithm. 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: binarization, Canny edge detection, morphological closing, dilation to fill the gaps between sketch curves and region filling.

3.2.2 Feature extraction

Shape context matching [3] is utilized to compute the distance between two outline feature views (one for sketch and one for model) during the retrieval stage. 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 [3] for each point. Finally, Jonker's LAP algorithm [12] is used to match the feature points of two outline feature views and the minimum matching cost is their distance.

3.2.3 Online 2D-3D comparison

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 2D-3D comparison algorithm.

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

(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.

For the query sketch, if we parallelly compute all the sketch-model distances, sort them in an ascending order and list the corresponding models accordingly, then we name this sketch-based retrieval algorithm based on our adaptive view clustering method **SBR-VC**.

4 Semantic sketch-based 3D retrieval approach

In this section, we further present the semantic retrieval approach in our sketch-based 3D model retrieval framework presented in Fig. 1. Considering the existing big semantic gap between the 2D sketches and 3D models, we further propose a novel semantic sketch-based 3D model search approach to bridge the semantic gap by first recognizing the potential semantic meanings of the user sketch and then performing 2D-3D matching for the 3D models within the predicted categories. In general, most, though not all, available 3D model databases have class information. Therefore, by using class information, it is feasible to further improve the retrieval performance [14]. We design our semantic approach mainly for the retrieval scenario whereby the class information of the target 3D dataset is already available, though the idea of our algorithm is general and applicable to unclassified datasets as well since we can perform 3D classification first, i.e., based on the 3D classification techniques reviewed in [14].

As shown in Fig. 1 (b), our proposed semantic sketch-based retrieval approach consists of two stages: sketch recognition training stage and sketch-based retrieval stage. In the sketch recognition training stage, a large sketch training dataset is selected first, which contains sketches from a variety of categories. Then, sketch features are explored and extracted to well describe input sketches' attributes. An intelligent sketch classifier is built up to recognize a user sketch into potential sketch categories. In the sketch-based retrieval stage, a query sketch is first fed into the developed sketch classifier and the possibilities of the input sketch belonging to all the categories are predicted. The higher the possibility is, the larger chance the user sketch describes the same object. Therefore, the 3D models in the top candidate categories are much closer to the input sketch. So, a general sketch-based 3D model retrieval algorithm, like the **SBR-VC** method presented in Section 3, is employed to rank the models in the top $\boldsymbol{L}$ candidate categories. The models in the remaining categories are ranked based on their corresponding categories' prediction values. By this joint usage of sketch recognition and retrieval techniques, the proposed approach can successfully bridge the semantic gap between sketches and 3D models.

4.1 Stage 1: Sketch recognition training

(1) Sketch feature extraction. Eitz et al. [5] extracted local features from a sketch based on the ideas of Scale-Invariant Feature Transform (SIFT) and Histogram of Gradient (HOG) features and embedded them into a Bag-of-Words framework as the feature representation. This feature representation has two limitations: 1) it only captures the local information of sketches and totally ignores the global attributes; 2) it cannot handle the rotation of sketches. In this section, a hybrid feature is developed by further integrating a set of rotation-invariant global features for a sketch. The hybrid feature vector is generated by combining the 500-dimensional Bag-of-Words local feature vector in [5] and a 119-dimensional global feature vector devised by us. First, each sketch is resized into a 300×300 image, then the thickness of the sketch lines is shrunk to a single pixel. Then, a global feature vector is extracted from the sketch as shown in Fig. 4. It is composed of 9 distance histograms: 5 radial distance histograms of the sketch pixels with respect to 5 selected reference points/lines, 2 radial distance histograms of the first intersection points and 2 radial angle histograms of the sketch pixels with respect to the two centers **C** and **FPC**. All the histograms are divided into 11 bins and the mean and standard deviation values of each histogram are also included, thus generating a 117-dimensional (13×9) global feature vector. In addition, the distance between **C** and **FPC**, and the sum of the distances between sketch pixels and **FPC** are also considered. Our experimental results show that a moderate improvement (around 3%∼5%) in sketch recognition has been achieved after incorporating these global features.

(2) Sketch classifier training. Similar as [5], Support Vector Machine (SVM) is chosen to build a sketch recognition model. The same parameter settings in [5] are utilized, such as local feature definitions, "soft" kernel-codebook coding choice, vocabulary size, and 3-fold cross-validation selection except that we may have different RBF kernel parameters.

4.2 Stage 2: Sketch-based retrieval

Given a query sketch q and a target 3D dataset $M = \{m_i\}$, a distance vector D needs to be generated to measure the dissimilarities between q and all the models in M, detailed as follows.

(3) Sketch classification. A query sketch is fed into the developed sketch classifier. The possibilities of the input sketch belonging to all the categories are predicted and the top $\boldsymbol{L}$ candidate categories are found.

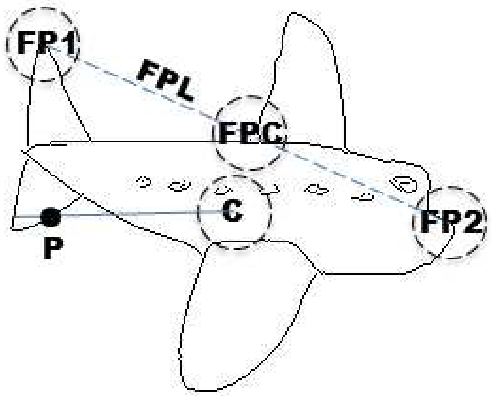


Fig. 4 Illustration of our 5 reference points or lines for the global features: **C**, **FP1**, **FP2**, **FPL** and **FPC**, where **C** is the centroid of a sketch, **FP1** and **FP2** are the two farthest points with respect to the centroid **C**, **FPL** is the line between the two farthest points; and **FPC** is the center of the **FPL** line. **P** is an example of first intersection point.

(4) 2D-3D matching. A 2D-3D comparison algorithm is applied on all the models in the top $\boldsymbol{L}$ candidate categories and the distances between the models and the input sketch, named D_1, are calculated. The proposed semantic sketch-based retrieval algorithm that combines the **SVM**-based sketch recognition (Steps (1)$\sim$(3)) and our view clustering-based 2D-3D comparison algorithm **SBR-VC** (Section 3) is named as **SBR_SVM-VC**.

(5) Distance vector generation. Besides D_1 for the top $\boldsymbol{L}$ candidate categories, we need to assign distances between the input sketch and the models in the remaining categories. As a whole, these distances form the second part of D, which is named as D_2. We first sort all the categories according to their SVM probability estimates generated in Step **(3)** and then assign D_2 distance of each model as the ranking order (larger than 1 since $\boldsymbol{L}\geq$1) of its respective category. To make the models in the top $\boldsymbol{L}$ candidate categories appear first in the rank list, we normalize the values in D_1 into [0,1] and then concatenate D_1 and D_2 sequentially to form one vector D.

When L=1, the retrieval performance of our semantic approach (Section 4) is independent of the 2D-3D matching algorithm used in Step (4) since it will only has impact on the retrieval performance of the top L candidate categories. If L=1, it does not matter which 2D-3D matching algorithm we will use because all the 3D models in the top L (1) candidate category are within the same class. Different 2D-3D matching algorithms will only change the order of the 3D models within that class. We also find that even if the relevant class of the 2D sketch query q is not within the top L candidate categories, usually the relevant class will still appear on the top of the rank list of all classes, which has been verified by the robust performance of our semantic approach in the following section, as well.

(6) Ranking and output. All the distances in D are sorted and the best matching models are retrieved accordingly.

5 Experimental results and discussion

In this section, we test our adaptive view clustering-based 2D-3D comparison algorithm "SBR-VC" and its semantic version "SBR_SVM-VC" on four latest (small-scale and large-scale) sketch-based 3D model retrieval benchmarks and compare with the state-of-the-art performance achieved on those benchmarks. We name the number of feature points (Section 3.2.2) to represent a sketch NUM, such as 50 or 100. 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.

Sections 5.2, 5.3 and 5.4 aim at demonstrating the advantages of our view clustering approach, while Section 5.5 is mainly to show the superior performance of our semantic approach.

5.1 Benchmarks and evaluation metrics

(1) Yoon et al.'s [36] [21] benchmark (SHREC12STB). We have introduced its target dataset in Section 3.1.2 and Fig. 3 (a). The query set contains 250 hand-drawn sketches for the 13 classes, each containing 17$\sim$21 sketches. It was used in the Eurographics 2012 Shape Retrieval Contest (SHREC'12) track on Sketch-Based 3D Shape Retrieval.

(2) Eitz et al.'s [7] Princeton Shape Benchmark based Benchmark (PSBB). Eitz et al. [7] built a large scale sketch-based 3D retrieval benchmark based on the Princeton Shape Benchmark (PSB) [30], which comprises train and test datasets, each with 907 models. They collected one sketch for each model based on the Amazon Mechanical Turk application to imitate novel users. That is, it contains 1814 sketches divided into 182 classes. However, PSB has quite different numbers of models for different classes, which is a bias for retrieval performance evaluation.

(3) SHREC'13 Sketch Track Benchmark [16] (SHREC13STB). Recently, a SHREC'13 Sketch Track Benchmark, for the Shape Retrieval Contest (SHREC) 2013 Track on the topic of large scale sketch-based retrieval, was developed based on the shared categories of the sketch recognition dataset built by Eitz et al. [5] and PSB. Eitz et al. [5] built a comprehensive benchmark for sketch recognition by utilizing the Amazon Mechanical Turk application as well. It comprises 20,000 sketches, uniformly divided into 250 classes. The SHREC'13 Sketch Track Benchmark contains 7200 hand-drawn sketches, that is 90 of the total 250 classes, and 1258 relevant 3D models selected from the PSB benchmark to form the target 3D dataset. To evaluate retrieval algorithms based on a learning approach, the "Training" and "Testing" datasets are also built by randomly selecting 50 sketches per class for training and the rest 30 sketches for testing, while the complete target model dataset is remained as a whole for both training and testing purpose.

(4) SHREC'14 Sketch Track Benchmark [20] [19] (SHREC14STB). It is the currently largest and latest large-scale unified retrieval benchmark which can be used for both generic and sketch-based 3D model retrieval. It was used in the Eurographics 2014 Shape Retrieval Contest (SHREC'14) track on Extended Large-Scale Sketch-Based 3D Shape Retrieval. It is comprised of 171 classes of 13680 sketches and 8987 models. To accommodate the evaluation for learning-based algorithms, 50 sketches for each class have been selected as the "Training" query dataset, while the remaining 30 sketches are assigned as the "Testing" query dataset. While, all the 8,987 3D models are remained as the target 3D dataset.

Evaluation metrics. To comprehensively evaluate the retrieval performance, we select the following seven commonly used performance metrics: Precision-Recall (PR) diagram, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-Measure (E), Discounted Cumulative Gain (DCG) [30] and Average Precision (AP). NN / FT / ST / E measures the performance of top one / T / $2T$ / 32 ranking results, where T is the total number of relevant models in the 3D model set. While, PR, DCG and AP measure the overall performance from different aspects. To measure the sketch recognition performance, eight popular metrics are utilized including True Positive rate (TP), False Positive rate (FP), Precision (P), Recall (R), F-Measure (F), Matthews Correlation Coefficient (MCC), area under the Receiver Operating Characteristic Curve (ROC), and area under Precision-Recall Curve (PRC) [26]. It is necessary to mention that for all the above metrics, except FP, a higher value indicates better performance. In the following tables, we also highlight the best results.

5.2 Yoon et al.'s benchmark (SHREC12STB)

To evaluate our 2D-3D comparison algorithm, we test the SBR-VC on the 250 sketches of Yoon et al.'s [36] [21] benchmark. Its average performance is compared with the top two state-of-the-art algorithms reported in the SHREC'12: Sketch-Based 3D Shape Retrieval Track [21]: Li and Johan's SBR-2D-3D algorithm [21] [13] and Eitz et al.'s BOF-SBR approach [21] [6]. Fig. 5 and Table 2 show their Precision-Recall diagram and other comparison results.

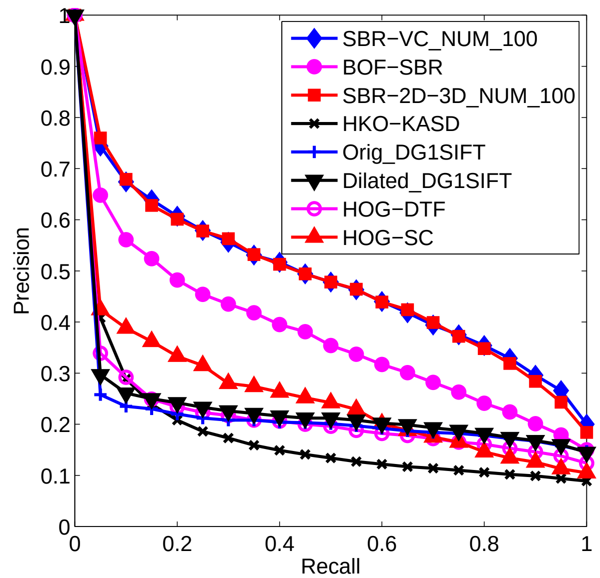


Fig. 5 Precision-Recall diagram performance comparisons on Yoon et al.'s [36] [21] benchmark between our method and other state-of-the-art algorithms.

As can be seen, our retrieval performance is apparently better than BOF-SBR. It also achieves similar performance as SBR-2D-3D algorithm. However, our precomputation for candidate views selection is simpler and more efficient in terms of both time and memory space. Table 1 compares SBR-VC and SBR-2D-3D in terms of the precomputation time and memory cost to load the precomputed features: SBR-2D-3D needs 4.6 times more precomputational time and 1.6 times more memory than SBR-VC. 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.

Table 1 Precomputation time and feature loading memory space comparison of **SBR-VC** and SBR-2D-3D on the Yoon et al.'s [36] [21] benchmark using a modern computer (CPU: Intel Core 2 Duo E7500@2.93 GHz; Memory: 16 GB; OS: Windows 7 (64-bit)).

Method	SBR-VC_NUM_100	SBR-2D-3D_NUM_100
Time (s)	**762.96**	4235.54
Memory (M)	**211.95**	550.72

Table 2 Other performance metrics comparison between our method **SBR-VC** and the top two participating approaches in SHREC'12 Track [21] on the Yoon et al.'s [36] [21] benchmark.

Method	NN	FT	ST	E	DCG	AP
SBR-VC_NUM_100	0.664	**0.427**	**0.587**	**0.413**	0.730	**0.558**
SBR-2D-3D_NUM_100	**0.688**	0.415	0.581	0.411	**0.731**	0.556
BOF-SBR	0.532	0.339	0.497	0.338	0.662	0.450

5.3 Eitz et al.'s PSBB benchmark (PSBB)

To compare with the latest Gabor local line-based feature (GALIF) proposed by Eitz et al.'s [7], we performed the same experiment as [7] on the PSBB benchmark [7]. Due to the big size of the target model dataset, we also tested the case of $\alpha = \frac{1}{6}$ (N_a=9) besides the default setting of $\alpha = \frac{1}{2}$ (N_a=29). In addition, we also ran the SBR-2D-3D algorithm on the dataset. Similarly, we tested setting the number of candidate views, denoted by n_{CV}, to 16 or 4 to observe their performance difference. Fig. 6 and Table 3 compare their Precision-Recall and other performance metrics, respectively.

As can bee seen from the figure and table, SBR-VC shows a comparable performance as the state-of-the-art approaches GALIF and SBR-2D-3D even on this biased benchmark, which also demonstrates the robustness of our approach. PSBB has one sketch for each model of PSB. PSB is the most famous and frequently used 3D shape benchmark and it also covers most commonly seen objects. However, PSB has quite different numbers of models for different classes, which is a bias for retrieval performance evaluation. For example, in the test dataset the "plane" class has 99 models while the "ant" class only has 5 models. What's more, in PSBB, the sketch dataset and the target model dataset share the same distribution in terms of number of objects in each class (one sketch for one model), thus the bias will be coupled. However, even on this biased benchmark, we have achieved similar performance as the GALIF in [7], where PSBB has been proposed. This shows our algorithm is robust with respect to the diversity of object classes.

It is also interesting to note that SBR-VC has a slightly better performance when choosing $\alpha = \frac{1}{6}$ than that of default setting ($\alpha = \frac{1}{2}$); while SBR-2D-3D achieves a better accuracy when keeping more candidate views. This is mainly because of their different schemes of view selection: SBR-2D-3D (linearly) selects a set of candidate views from 81 sample views according to their similarities with the query sketch while SBR-VC (nonlinearly) clusters 81 sample views into a set of representative views, each of which represents a cluster of sample views.

5.4 SHREC'14 Sketch Track Benchmark (SHREC14STB)

In [20] [19], an extensive comparative evaluation of six methods (including our SBR-VC) contributed by four participating groups has been performed on the SHREC 14STB benchmark. Readers can refer to them for complete details. For inclusiveness, we present the most important and relevant comparison results here.

Fig. 7 compares their PR performance, while Table 5 compares the other six reciprocally weighted performance metrics [20] on the "Testing" dataset. We choose the reciprocally weighted performance metrics due to their better accuracy and robustness since they are designed to reduce the bias in the different number of models for each class by multiplying each query's performance values by the reciprocal of the number of relevant model in the target 3D model dataset.

As can be seen from the figure and tables, the two machine learning-based semantic approaches, that is SCMR-OPHOG and CDMR, have the best performance. While, the overall performance of the top methods from other non-learning based approaches are very close. Within the non-learning based category, our SBR-VC ranks #4.

Their time efficiency has been compared in Table 4. BF-fGALIF has the best efficiency, followed by BOF-JESC and SBR-VC ($\alpha = \frac{1}{2}$).

5.5 SHREC'13 Sketch Track Benchmark (SHREC13STB)

5.5.1 Without semantic approach: performance of SBR-VC

We have tested our SBR-VC algorithm on the "Training", "Testing", and "Complete" datasets and compared it with other participating approaches in the SHREC'13 Sketch Track [16], which include Li et al.'s SBR-2D-3D, Saavedra's FDC, and Aono and Tashiro's EFSD. For SBR-VC, to accustom to the large-scale retrieval for efficiency consideration, we keep less representative views by setting $\alpha = \frac{1}{6}$ (N_a=9). For SBR-2D-3D, we set NUM to 50 because when NUM=100, it needs a large amount of memory (e.g., an ASCII text file of 2.8 GB is

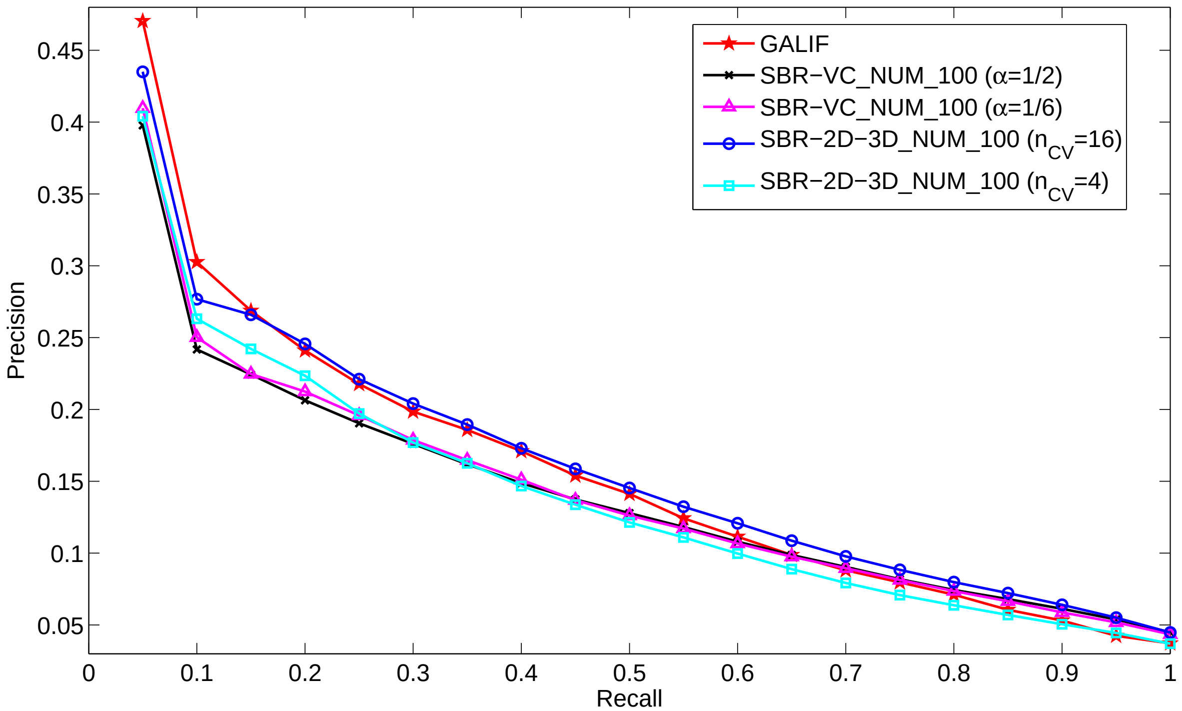


Fig. 6 Precision-Recall diagram performance comparisons on the testing dataset of the PSBB [7] benchmark between our method and other state-of-the-art algorithms.

Table 3 Other performance metrics for the performance comparison on the testing dataset of the PSBB [7] benchmark. "-" means the performance data are not available.

Method	NN	FT	ST	E	DCG	AP
SBR-VC_NUM_100 (α=1/2)	0.192	0.117	0.174	0.100	0.389	0.130
SBR-VC_NUM_100 (α=1/6)	0.198	0.118	0.180	0.104	0.391	0.131
SBR-2D-3D_NUM_100 (n_{CV}=16)	**0.228**	**0.137**	**0.201**	**0.116**	**0.408**	**0.147**
SBR-2D-3D_NUM_100 (n_{CV}=4)	0.213	0.120	0.177	0.102	0.390	0.128
GALIF	-	-	-	-	-	0.143

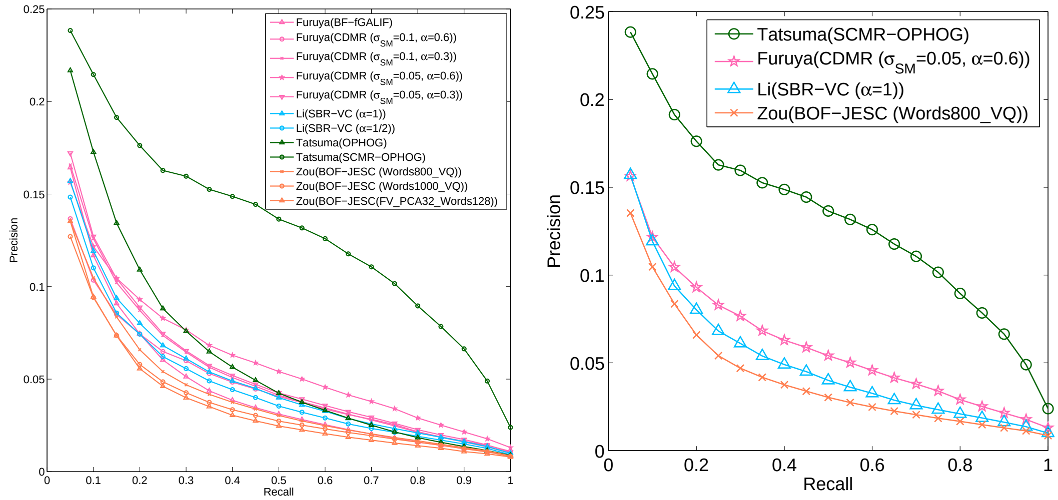
needed to save the shape context features) to load the relative shape context features and the retrieval speed will be largely decreased.

Fig. 8 and Table 6 compare their performance in terms of Precision-Recall diagram performance. 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 feature points but less representative views because of the change in N_c. Compared with SBR-2D-3D_NUM_50, they are comparable in retrieval efficiency, but SBR-VC_NUM_100 has much better retrieval accuracy. For example, considering the retrieval on the complete benchmark, SBR-VC_NUM_100 outperforms SBR-2D-3D_NUM_50 by 21.1%, 22.8%, 19.2%, 14.9% and 6.1% in terms of NN, FT, ST, E, and DCG respectively.

5.5.2 With semantic approach: performance of SBR_SVM-VC

Now, we further test our semantic approach on the SHREC'13 Sketch Track Benchmark. We train the SVM-based classifier on the training dataset based on our hybrid features (containing both local and global features) and obtain the best parameters values: Gaussian kernel parameter γ=0.1 and penalty parameter C_p=20.

1) Sketch recognition. The developed sketch recognizer (Section 4.1) in our semantic retrieval approach is tested on both the SHREC'13 Sketch Track Benchmark dataset and Eitz et al.'s complete sketch recognition benchmark [5]. Its recognition performance is compared with the newest sketch recognition algorithm (local feature based approach) proposed in [5], as shown in Table 7. As can be seen, our approach achieves better results in all the metrics on both benchmark datasets. It validates that our sketch recognizer is more robust to



(a) Testing dataset (b) Testing dataset (best runs)

Fig. 7 Precision-Recall diagram performance comparisons [19] on the "Testing" dataset of the **SHREC14STB** benchmark for the twelve runs of six participating sketch-based 3D model retrieval methods from the four participating groups, including our SBR-VC method.

Table 4 Timing information comparison [19] of the six participating sketch-based 3D model retrieval algorithms, including our SBR-VC method: T is the average response time (in seconds) per query based on the "Testing" dataset. "R" denotes the ranking order of all the twelve runs, while "R$_p$" denotes the ranking order of all the runs that do not utilize any machine learning techniques, that is, the runs of the pure shape descriptors themselves.

Group (with computer configuration)	**Method**	**Language**	T	**R**	**R$_p$**
Furuya (CPU: Intel(R) Core i7 3930K @3.20 GHz, GPU: NVIDIA GeForce GTX 670 (on a single thread); Memory: 64 GB; OS: Ubuntu 12.04)	BF-fGALIF	C++	**1.82**	**1**	**1**
	CDMR	C++, CUDA	126.81	7	-
Li (CPU: Intel(R) Xeon(R) CPU X5675 @3.07 GHz (2 processors, 12 cores); Memory: 20 GB; OS: Windows 7 64-bit)	SBR-VC (α=1)	C/C++	27.49	6	5
	SBR-VC ($\alpha = \frac{1}{2}$)	C/C++	15.16	3	3
Tatsuma (CPU: Intel(R) Xeon(R) CPU E5-2630 @2.30GHz (2 processors, 12 cores); Memory: 64 GB; OS: Debian Linux 7.3)	OPHOG	C++, Python	23.85	4	4
	SCMR-OPHOG	C++, Python	25.67	5	-
Zou (CPU: Intel(R) Xeon(R) W3550@3.07GHz (the programs ran on a single thread); Memory: 24 GB; OS: Windows 7 64-bit)	BOF-JESC	Matlab	6.10	2	2

sketch rotation and can well describe user sketches by incorporating global attributes of sketches. For our algorithm, the time taken to recognize the 2700 sketches is 269.95 seconds, thus averagely 0.1 second is needed to classify a sketch.

2) Semantic retrieval. The proposed **SBR_SVM-VC** (Section 4) method is tested on the SHREC'13 Sketch Track Benchmark and mainly compared with our SBR-VC algorithm which does not utilize the semantic approach. As can be found in the competition track report [16], SBR-VC achieved the best performance on the SHREC'13 Sketch Track Benchmark, while SBR-2D-3D closely followed SBR-VC. For completeness and reference, we also list the performance of other participating approaches in the competition, such as SBR-2D-3D, FDC, and EFSD. We need to mention that the purpose of listing the performance of other approaches that do not utilize the available class information, is only for a contrast and to demonstrate the improvement that our semantic retrieval approach can achieve. It is not used for a direct evaluation, or even comparison. In addition, to compare with other latest established approaches, we also compare with the latest deep learning-based approach [34], which has performed the same experiment on this SHREC'13 Sketch Track Benchmark.

In our approach, a variety of length values ($\boldsymbol{L}$) for the candidate category list is tested. Figs. 9∼10 and Table 8 show the comparison results. As can be seen, compared to SBR-VC, after employing a semantic retrieval approach, the retrieval performance is significantly improved. What's more, the generality of our semantic approach has been verified as well: it can be used with other retrieval techniques, e.g. SBR_SVM-2D-3D_NUM_50 in the figures and table. It also can

Table 5 Reciprocally weighted performance metrics comparison on the "Testing" dataset of **SHREC14STB** benchmark for the twelve runs of six sketch-based 3D model retrieval methods from the four SHREC'14 Sketch Track participating groups. "R" denotes the ranking order of all the twelve runs, while "R_p" denotes the ranking order of all the runs that do not utilize any machine learning techniques, that is, the runs of the pure shape descriptors themselves.

Contributor	**Method**	**NN**	**FT**	**ST**	**E**	**DCG**	**AP**	**R**	$\mathbf{R}_p$
Testing dataset				1.0e-05*					
Furuya	BF-fGALIF	0.802	0.520	0.735	0.289	3.408	0.596	4	2
	CDMR (σ_{SM}=0.1, α=0.6)	0.299	0.237	0.406	0.222	2.861	0.281	11	-
	CDMR (σ_{SM}=0.1, α=0.3)	0.679	0.467	0.719	0.308	3.323	0.553	6	-
	CDMR (σ_{SM}=0.05, α=0.6)	0.576	0.467	0.782	0.318	3.305	0.583	5	-
	CDMR (σ_{SM}=0.05, α=0.3)	0.789	0.526	0.773	0.330	3.430	0.626	2	-
Li	SBR-VC (α=1)	0.449	0.264	0.425	0.264	3.051	0.291	9	5
	SBR-VC ($\alpha = \frac{1}{2}$)	0.414	0.265	0.405	0.259	3.088	0.311	8	4
Tatsuma	OPHOG	0.917	0.509	0.777	0.396	3.539	0.615	3	**1**
	SCMR-OPHOG	**0.993**	**0.743**	**1.035**	**0.541**	**3.676**	**0.886**	**1**	-
Zou	BOF-JESC (Words800_VQ)	0.462	0.271	0.467	0.236	3.149	0.370	7	3
	BOF-JESC (Words1000_VQ)	0.403	0.208	0.356	0.194	3.020	0.286	10	6
	BOF-JESC (FV_PCA32_Words128)	0.455	0.225	0.336	0.170	2.910	0.254	12	7

Table 6 Other performance metrics for the performance comparison on the SHREC'13 Sketch Track Benchmark [16].

Method	NN	FT	ST	E	DCG	AP
Training dataset						
SBR-VC-NUM-100	**0.160**	**0.097**	**0.149**	**0.085**	**0.349**	**0.113**
SBR-VC-NUM-50	0.131	0.082	0.130	0.076	0.333	0.098
SBR-2D-3D-NUM-50	0.133	0.080	0.126	0.075	0.330	0.097
FDC	0.051	0.039	0.069	0.041	0.279	0.051
EFSD	0.024	0.019	0.038	0.020	0.241	0.032
Testing dataset						
SBR-VC-NUM-100	**0.164**	**0.097**	**0.149**	**0.085**	**0.348**	**0.114**
SBR-VC-NUM-50	0.132	0.082	0.131	0.075	0.331	0.098
SBR-2D-3D-NUM-50	0.132	0.077	0.124	0.074	0.327	0.095
FDC	0.053	0.038	0.068	0.041	0.279	0.051
EFSD	0.023	0.019	0.036	0.019	0.240	0.031
Complete benchmark						
SBR-VC-NUM-100	**0.161**	**0.097**	**0.149**	**0.085**	**0.349**	**0.113**
SBR-VC-NUM-50	0.131	0.082	0.130	0.076	0.332	0.098
SBR-2D-3D-NUM-50	0.133	0.079	0.125	0.074	0.329	0.096
FDC	0.052	0.039	0.069	0.041	0.279	0.051
EFSD	0.023	0.019	0.037	0.019	0.241	0.032

Table 7 Sketch recognition performance comparison: Average classification performance comparison between our SVM-based approach proposed in Section 4.1 and the local feature based approach proposed in [5] in terms of eight metrics.

Method	TP	FP	P	R	F	MCC	ROC	PRC
SHREC'13 Sketch Track Benchmark								
Our approach	**0.613**	**0.004**	**0.623**	**0.613**	**0.614**	**0.612**	**0.982**	**0.664**
Eitz et al. [5]	0.594	0.005	0.597	0.594	0.593	0.590	0.974	0.637
Eitz et al.'s [5] Sketch Benchmark								
Our approach	**0.545**	**0.002**	**0.549**	**0.545**	**0.544**	**0.544**	**0.772**	**0.326**
Eitz et al. [5]	0.520	**0.002**	0.523	0.520	0.519	0.518	0.759	0.298

be found that our SVM-based semantic approach also outperforms by a large margin the deep learning-based approach CNN-Siamese [34]. We need to point out that deep learning has been regarded as one of the most promising techniques in developing better sketch-based 3D model retrieval algorithms. In a word, all the above facts have demonstrated the apparent advantages and better robustness of our semantic approach framework.

In addition, during online retrieval, the parallelizations in the shape context feature computation for a query sketch and the 2D-3D matching between the query sketch and all the 3D models, significantly (9x for our CPUs with 12 cores) accelerates the retrieval speed and make our algorithm real-time.

5.6 Discussions

5.6.1 Classification

To know the overall information and guide our future research, in Table 9 we classify all the eighteen SBR methods evaluated in this section with respect to the techniques employed according to the classification standards described in [17]: local/global 2D features, Bag-of-Words/Bag-of-Features (BoW/BoF) framework or direct feature matching (DFM), fixed/clustered views, and with/without view selection. In the end of Table 9, we also list their statistical information to have an overall picture of current status in sketch-based 3D model retrieval research as well as to solicit potentials for further advancements. We conclude that semantic approaches and machine learning techniques, especially deep learning methods, which have been validated in recent related work, like [37] [8] and [38], are among the most promising techniques that can help to further improve SBR performance to meet the practical requirements of vast related applications.

5.6.2 On semantic approach

Comments on our semantic approach w.r.t the experiment on the SHREC'13 Sketch Track Benchmark. As can be seen from Figs. 9∼10, we have the following findings or conclusions. The significant improvement in the retrieval performance further validates that our approach can bridge the semantic gap between the diverse query sketches and 3D models effectively. What's more, the average retrieval time for each query is also substantially decreased. In fact, the time for recognizing a sketch is only about 0.1 second, which adds little online computational load. Thus, our sketch recognizer can help to find a much smaller number of 3D models for direct 2D-3D comparison, which not only significantly improves the retrieval accuracy, but also manifestly reduces the computational time.

5.6.3 On view clustering-based 2D-3D comparison method

View selection contribution. To find out the contribution of our view selection step, we have an analogical analysis based on SBR-2D-3D's algorithm component contribution analysis (Section 6.5 of [13]) and the following two facts. First, both our approach and SBR-2D-3D adopt relative shape context matching for the matching part and the difference is mainly in the candidate views selection, therefore the difference in the retrieval performance of the two approaches is now only dependent on the goodness of the views selected. Second, in Sections 5.2∼5.5, we have demonstrated that our view selection-based retrieval algorithm SBR-VC has a comparable performance as SBR-2D-3D.

Based on the above two facts and analysis, we can draw a conclusion that the algorithm component contribution analysis results of SBR-2D-3D are also applicable to our approach. That is, both the 3D information complexity metric-based view clustering approach and the relative shape context matching on the outline feature views, have important contributions to our good performance.

Comparison with other view clustering approaches. An-sary et al. [2] proposed the Bayesian view clustering algorithm for Query-by-Model retrieval. The main idea of their approach is based on the fact that "not all the views of a 3D model have equal importance: there are views that contain more information than others". Compared with Ansary et al.'s approach, our algorithm has the following differences or advantages.

Firstly, rather than using region-based shape descriptor Zer-nike moments like Ansary et al. [2], we employ an information-related metric viewpoint entropy to measure the information of a view contains and further base on it to measure the 3D information complexity of a 3D model. Compared with the physics-based Zernike moments, the information theory-based viewpoint entropy measurement contains more semantic information, and is thus more reasonable and effective in measuring the 3D information complexity of a 3D model.

Secondly, our framework has more freedom in assigning the number of representative views. We can easily adjust the number of representative views by simply changing the constant parameter α in Eq. 3, e.g., we have changed it to $\frac{1}{6}$ for large scale sketch-based 3D model retrieval application in Section 5.5. The generality property is very important for different kinds of applications where we have various considerations, such as the availability of resources, time and accuracy requirements.

Thirdly, Ansary et al.'s approach favors views with more information and thus prefers selecting views with bigger projection areas. On the other hand, by incorporating the position information of the viewpoint into the view features $E = (x, y, z, e)$, we cluster the views over the full viewing sphere. Thus, it has an advantage to adapt to sketches with different variations.

Last but not least, the algorithm developed in Ansary et al. [2] is for the Query-by-Model scenario, which is reasonable and feasible. However, there is no guarantee that this mechanism is effective for sketch-based 3D retrieval while we have already proved the effectiveness and advantages of our method.

6 Conclusions and future work

A new sketch-based 3D retrieval framework integrating adaptive view clustering and semantic search has been proposed in the paper. Evident advantages and significantly better performance have been demonstrated via extensive experiments on both small-scale and large-scale retrieval benchmarks. Below, we conclude our work with respect to the two main components of the framework, respectively.

2D-3D comparison using viewpoint entropy-based view clustering. We have presented a 2D-3D comparison algorithm by first adaptively clustering the sampled views of a 3D model into a set of representative feature views and then employing parallel shape context matching to compare the sketch with the representative feature views of all the models. A 3D information complexity metric is first proposed based on the viewpoint entropy distribution of a set of sample views, and based on it 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 and superior performance of the correspondent SBR-VC retrieval algorithm.

Semantic sketch-based 3D model retrieval. A semantic sketch-based 3D model retrieval algorithm is further proposed in this paper by first performing sketch recognition to find a set of candidate categories for the sketch and then applying direct 2D-3D comparison on the models within the candidate classes. It is an important improvement to encompass the semantic gap between the sketches and models. The experimental results demonstrate the significant improvements in both retrieval accuracy and computationally efficiency. The developed sketch recognition algorithm also further improves sketch recognition by integrating global sketch features.

Future work. As the future work for the view clustering-based 2D-3D comparison algorithm, we plan to test the performance of the relating SBR-VC 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 information complexity metric into other related applications. In addition, a soft acceptance of the first category only is also worthy of further study.

Acknowledgments

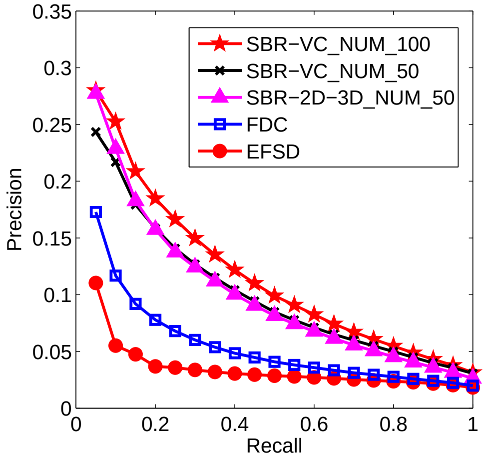
The work of Bo Li and Yijuan Lu is supported by the Texas State University Research Enhancement Program (REP), Army Research Office grant W911NF-12-1-0057, and NSF CNS 1305302 to Dr. Yijuan Lu.

The research done by Henry Johan in Fraunhofer IDM@NTU is supported by the National Research Foundation, Prime Ministers Office, Singapore under its International Research Centres in Singapore Funding Initiative.

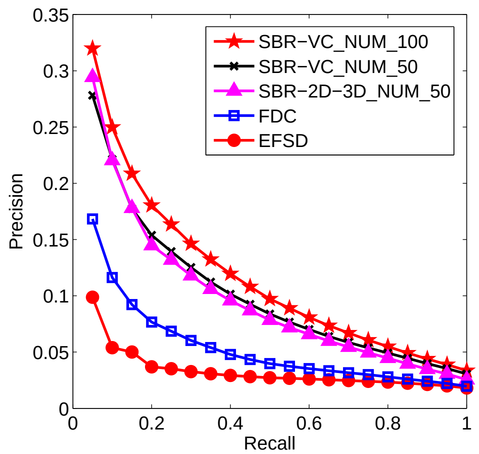
References

1. AIM@SHAPE: SHREC Contest. Home Page (2010). http://www.aimatshape.net/event/SHREC/
2. Ansary, T.F., Daoudi, M., Vandeborre, J.P.: A Bayesian 3-D search engine using adaptive views clustering. IEEE Transactions on Multimedia **9**(1), 78–88 (2007)
3. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4), 509–522 (2002)
4. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms (1981)
5. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Trans. Graph. **31**(4), 44:1–44:10 (2012)
6. 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), 1624–1636 (2011)
7. Eitz, M., Richter, R., Boubekeur, T., Hildebrand, K., Alexa, M.: Sketch-based shape retrieval. ACM Trans. Graph. **31**(4), 31:1–31:10 (2012)
8. Fang, Y., Xie, J., Dai, G., Wang, M., Zhu, F., Xu, T., Wong, E.K.: 3D deep shape descriptor. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pp. 2319–2328. IEEE Computer Society (2015)
9. Gong, B., Liu, J., Wang, X., Tang, X.: Learning semantic signatures for 3D object retrieval. IEEE Transactions on Multimedia **15**(2), 369–377 (2013)
10. Hou, S., Lou, K., Ramani, K.: SVM-based Semantic Clustering and Retrieval of A 3D Model Database. Computer-Aided Design and Applications **2**, 155–164 (2005)
11. Jayanti, S., Kalyanaraman, Y., Iyer, N., Ramani, K.: Developing an engineering shape benchmark for CAD models. Computer-Aided Design **38**(9), 939–953 (2006)
12. Jonker, R., Volgenant, A.: A shortest augmenting path algorithm for dense and sparse linear assignment problems. Computing **38**(4), 325–340 (1987)
13. Li, B., Johan, H.: Sketch-based 3D model retrieval by incorporating 2D-3D alignment. Multimedia Tools and Applications pp. 1–23 (2012)
14. Li, B., Johan, H.: 3D model retrieval using hybrid features and class information. Multimedia Tools Appl. **62**(3), 821–846 (2013)
15. Li, B., Lu, Y., Fares, R.: Semantic sketch-based 3D model retrieval. In: ICME, pp. 1–4 (2013)
16. 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, pp. 89–96 (2013)

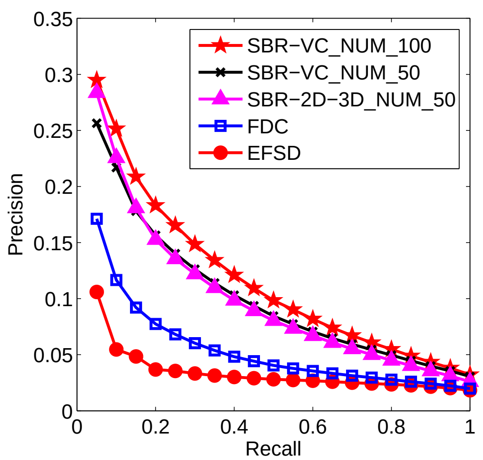
17. Li, B., Lu, Y., Godil, A., Schreck, T., Bustos, B., Ferreira, A., Furuya, T., Fonseca, M.J., Johan, H., Matsuda, T., Ohbuchi, R., Pascoal, P.B., Saavedra, J.M.: A comparison of methods for sketch-based 3D shape retrieval. Computer Vision and Image Understanding **119**, 57–80 (2014)
18. Li, B., Lu, Y., Johan, H.: Sketch-based 3D model retrieval by viewpoint entropy-based adaptive view clustering. In: Eurographics Workshop on 3D Object Retrieval, Girona, Spain, 2013. Proceedings, pp. 49–56 (2013)
19. Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., Burtscher, M., Chen, Q., Chowdhury, N.K., Fang, B., Fu, H., Furuya, T., Li, H., Liu, J., Johan, H., Kosaka, R., Koyanagi, H., Ohbuchi, R., Tatsuma, A., Wan, Y., Zhang, C., Zou, C.: A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries. Computer Vision and Image Understanding **131**, 1–27 (2015). URL `http://dx.doi.org/10.1016/j.cviu.2014.10.006`
20. Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., Burtscher, M., Fu, H., Furuya, T., Johan, H., Liu, J., Ohbuchi, R., Tatsuma, A., Zou, C.: SHREC'14: Extended Large Scale Sketch-Based 3D Shape Retrieval. In: Eurographics Workshop on 3D Object Retrieval, Strasbourg, France, 2014. Proceedings, pp. 121–130 (2014)
21. 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., Yoon, S.M.: SHREC'12 track: Sketch-based 3D shape retrieval. In: Eurographics Workshop on 3D Object Retrieval 2012, Cagliari, Italy, May 13, 2012. Proceedings, pp. 109–118 (2012)
22. Mokhtarian, F., Abbasi, S.: Robust automatic selection of optimal views in multi-view free-form object recognition. Pattern Recognition **38**(7), 1021–1031 (2005)
23. Nie, L., Yan, S., Wang, M., Hong, R., Chua, T.: Harvesting visual concepts for image search with complex queries. In: N. Babaguchi, K. Aizawa, J.R. Smith, S. Satoh, T. Plagemann, X. Hua, R. Yan (eds.) Proceedings of the 20th ACM Multimedia Conference, MM '12, Nara, Japan, October 29 - November 02, 2012, pp. 59–68. ACM (2012). DOI 10.1145/2393347.2393363. URL `http://doi.acm.org/10.1145/2393347.2393363`
24. Ohbuchi, R., Yamamoto, A., Kobayashi, J.: Learning semantic categories for 3D model retrieval. In: Multimedia Information Retrieval, pp. 31–40 (2007)
25. Page, D.L., Koschan, A., Sukumar, S.R., Roui-Abidi, B., Abidi, M.A.: Shape analysis algorithm based on information theory. In: ICIP (1), pp. 229–232 (2003)
26. Powers, D.M.W.: Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. Journal of Machine Learning Technologies **2**(1), 37–63 (2007)
27. Rossignac, J.: Shape complexity. The Visual Computer **21**(12), 985–996 (2005)
28. Saleem, W., Belyaev, A.G., Wang, D., Seidel, H.P.: On visual complexity of 3D shapes. Computers & Graphics **35**(3), 580–585 (2011)
29. ShapeNet: `http://shapenet.cs.stanford.edu/` (2016)
30. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The Princeton shape benchmark. In: SMA, pp. 167–178 (2004)
31. Takahashi, S., Fujishiro, I., Takeshima, Y., Nishita, T.: A feature-driven approach to locating optimal viewpoints for volume visualization. In: IEEE Visualization, pp. 495–502 (2005)
32. 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), 689–700 (2003)
33. Vranic, D.: 3D Model Retrieval. PhD thesis, University of Leipzig (2004)
34. Wang, F., Kang, L., Li, Y.: Sketch-based 3D shape retrieval using convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pp. 1875–1883 (2015)
35. Wessel, R., Blümel, I., Klein, R.: A 3D shape benchmark for retrieval and automatic classification of architectural data. In: Eurographics Workshop on 3D Object Retrieval, 3DOR 2009, pp. 53–56 (2009)
36. Yoon, S.M., Scherer, M., Schreck, T., Kuijper, A.: Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours. In: ACM Multimedia, pp. 193–200 (2010)
37. Zhu, F., Xie, J., Fang, Y.: Learning cross-domain neural networks for sketch-based 3D shape retrieval. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA., pp. 3683–3689 (2016)
38. Zhu, J., Zhu, F., Wong, E.K., Fang, Y.: Learning pairwise neural network encoder for depth image-based 3D model retrieval. In: X. Zhou, A.F. Smeaton, Q. Tian, D.C.A. Bulterman, H.T. Shen, K. Mayer-Patel, S. Yan (eds.) Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM '15, Brisbane, Australia, October 26 - 30, 2015, pp. 1227–1230. ACM (2015)



(a) Training dataset

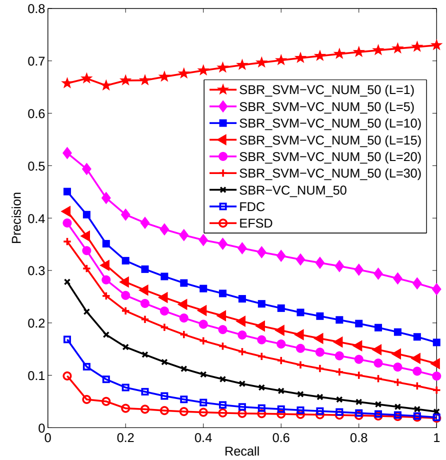


(b) Testing dataset

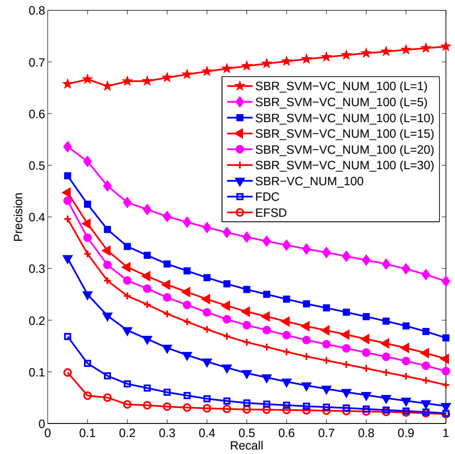


(c) Complete dataset

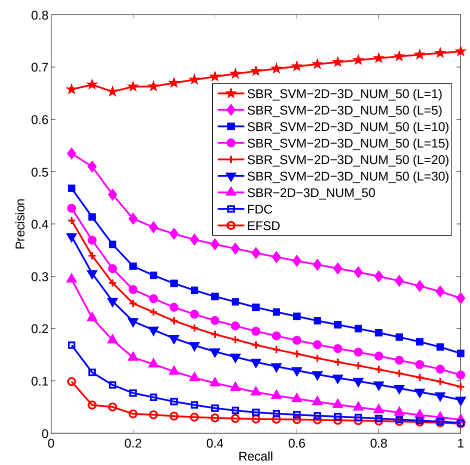
Fig. 8 Precision-Recall diagram performance comparisons on different datasets of the SHREC'13 Sketch Track Benchmark [16] between our method and other state-of-the-art algorithms.



(a) SBR_SVM-VC_NUM_50

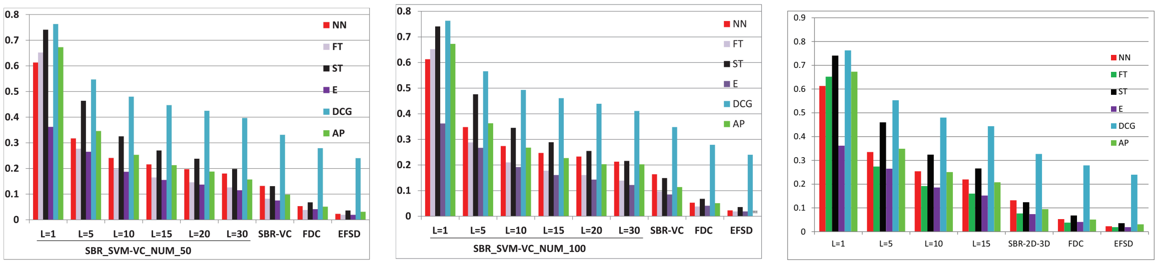


(b) SBR_SVM-VC_NUM_100



(c) SBR_SVM-2D-3D_NUM_50

Fig. 9 Precision-Recall diagram performance comparison between our method **SBR_SVM-VC** (different $\boldsymbol{L}$ values), together with SBR_SVM-2D-3D, and the participating approaches in the SHREC'13 Sketch Track Contest on the "Testing" dataset.



(a) SBR_SVM-VC_NUM_50 (b) SBR_SVM-VC_NUM_100 (c) SBR_SVM-2D-3D_NUM_50

Fig. 10 Other performance metrics comparison between our semantic algorithms **SBR_SVM-VC** (different $\boldsymbol{L}$ values: $\boldsymbol{L}$=1, 5, 10, 15, 20, 30), together with SBR_SVM-2D-3D, and the participating approaches in the SHREC'13 Sketch Track Contest on the "Testing" dataset. Please note different NUM values for the SBR-VC method in (a) (NUM=50) and (b) (NUM=100) and NUM=50 for the SBR-2D-3D method in (c).

Table 8 Other performance metrics comparison between our semantic algorithm SBR_SVM-VC, together with SBR_SVM-2D-3D, and the participating approaches in the SHREC'13 Sketch Track on the "Testing" dataset. $\boldsymbol{t}$ is the average response time (s) per query based on a modern computer (CPU: Intel(R) Xeon(R) CPU X5675 @3.07 GHz (2 processors, 12 cores); Memory: 20 GB; OS: Windows 7 (64-bit)).

Method	**NN**	**FT**	**ST**	**E**	**DCG**	**AP**	$\boldsymbol{t}$ **(s)**
SBR-VC_NUM_50							
SBR_SVM-VC_NUM_50 (L=1)	**0.613**	**0.652**	**0.741**	**0.362**	**0.763**	**0.673**	**0.16**
SBR_SVM-VC_NUM_50 (L=5)	0.317	0.277	0.464	0.265	0.547	0.346	0.43
SBR_SVM-VC_NUM_50 (L=10)	0.241	0.197	0.325	0.187	0.480	0.253	1.02
SBR_SVM-VC_NUM_50 (L=15)	0.216	0.165	0.270	0.155	0.447	0.213	1.44
SBR_SVM-VC_NUM_50 (L=20)	0.197	0.146	0.238	0.137	0.425	0.188	2.22
SBR_SVM-VC_NUM_50 (L=30)	0.180	0.126	0.198	0.115	0.397	0.157	2.88
SBR-VC_NUM_50	0.132	0.082	0.131	0.075	0.331	0.098	7.37
SBR-VC_NUM_100							
SBR_SVM-VC_NUM_100 (L=1)	**0.613**	**0.652**	**0.741**	**0.362**	**0.763**	**0.673**	**0.34**
SBR_SVM-VC_NUM_100 (L=5)	0.348	0.288	0.476	0.267	0.566	0.363	1.63
SBR_SVM-VC_NUM_100 (L=10)	0.274	0.210	0.345	0.192	0.493	0.268	2.21
SBR_SVM-VC_NUM_100 (L=15)	0.247	0.178	0.289	0.161	0.461	0.227	3.22
SBR_SVM-VC_NUM_100 (L=20)	0.233	0.161	0.255	0.143	0.439	0.203	4.92
SBR_SVM-VC_NUM_100 (L=30)	0.213	0.139	0.216	0.122	0.411	0.2026	7.23
SBR-VC_NUM_100	0.164	0.097	0.149	0.085	0.348	0.114	22.33
SBR-2D-3D_NUM_50							
SBR_SVM-2D-3D_NUM_50 (L=1)	**0.613**	**0.652**	**0.741**	**0.362**	**0.763**	**0.673**	0.15
SBR_SVM-2D-3D_NUM_50 (L=5)	0.335	0.274	0.460	0.265	0.553	0.349	0.52
SBR_SVM-2D-3D_NUM_50 (L=10)	0.254	0.192	0.324	0.186	0.480	0.251	1.00
SBR_SVM-2D-3D_NUM_50 (L=15)	0.220	0.160	0.266	0.152	0.444	0.208	1.42
SBR_SVM-2D-3D_NUM_50 (L=20)	0.206	0.142	0.233	0.133	0.421	0.182	1.82
SBR_SVM-2D-3D_NUM_50 (L=30)	0.180	0.119	0.193	0.113	0.392	0.152	2.68
SBR-2D-3D_NUM_50	0.132	0.077	0.124	0.074	0.327	0.095	4.70
FDC	0.053	0.038	0.068	0.041	0.279	0.051	**0.02**
EFSD	0.023	0.019	0.036	0.019	0.240	0.031	20.24
CNN-Siamese [34]	0.405	0.403	0.548	0.287	0.607	0.469	0.002

Table 9 Classification of the eighteen evaluated methods in our experiments on the four benchmarks: SHREC12STB, PSBB, SHREC13STB and SHREC14STB. G: global, L: local, FIX: Predefined views, VC/VS: view clustering/selection, DFM: direct feature matching, BoW: Bag-of-Words, BoF: Bag-of-Features, SVC: super-vector coding, MR: manifold ranking, CDMR: Cross-Domain Manifold Ranking, SCMR: Similarity Constrained Manifold Ranking, DL: Deep Learning, ML: Machine Learning-based, NonL: Non-learning based, SEM: Semantic approach, NonSEM: Non-semantic approach.

Index	Evaluated method	Feature type	View Sampling	Feature coding/ matching	Learning scheme	Semantic information
			SHREC12STB			
1	**SBR-VC**	G	VC	DFM	✗	✗
2	**BOF-SBR**	L	FIX	BoF	✗	✗
3	**HKO-KASD**	L	FIX	DFM	✗	✗
4	**Orig_DG1SIFT**	L	FIX	BoW	✗	✗
5	**HOG-DTF**	G	FIX	DFM	✗	✗
6	**HOG-SC**	G	FIX	DFM	✗	✗
			PSBB			
7	**GALIF**	L	FIX	BoF	✗	✗
8	**SBR-2D-3D**	L	VS	DFM	✗	✗
			SHREC14STB			
9	**BF-fGALIF**	L	FIX	BoW	✗	✗
10	**CDMR**	L	FIX	BoW	MR (CDMR)	✗
11	**OPHOG**	L	FIX	DFM	✗	✗
12	**SCMR-OPHOG**	L	FIX	DFM	MR (SCMR)	✗
13	**BOF-JESC**	L	FIX	BoF	✗	✗
			SHREC13STB			
14	**FDC**	G	FIX	DFM	✗	✗
15	**EFSD**	G	FIX	DFM	✗	✗
16	**CNN-Siamese**	L	FIX	DL	DL	✓
17	**SBR_SVM_2D_3D**	L	VS	DFM	SVM	✓
18	**SBR_SVM_VC**	L	VC	DFM	SVM	✓
			Statistics of the Classification (#)			
All Eighteen Evaluated Methods		**G:** 5	**FIX:** 14	**DFM:** 11	**ML:** 5	**SEM:** 3
		L: 13	**VC/VS:** 4	**BoF/BoW:** 6	**NonL:** 13	**NonSEM:** 15
				DL: 1		