Noname manuscript No.
(will be inserted by the editor)

3D Sketching for 3D Object Retrieval

Bo Li · Juefei Yuan · Yuxiang Ye · Yijuan Lu* · Chaoyang Zhang · Qi Tian

Received: date / Accepted: date

Abstract Sketching provides the most natural way to provide a visual search query for visual object search. However, how to draw 3D sketches in a three-dimensional space and how to use a hand-drawn 3D sketch to search similar 3D models are not only interesting and novel, but also challenging research topics. In this paper, we try to answer them by initiating a novel study on 3D sketching and build a 3D sketching system which allows users to freely draw 3D sketches in the air and demonstrate its promising potentials in related applications such as collecting 3D sketch data and conducting 3D sketch-based 3D model retrieval. By utilizing the 3D sketching system, we collect a 3D sketch dataset, build a 3D sketch-based 3D model retrieval benchmark, and organize a Eurographics Shape Retrieval Contest (SHREC) track on 3D sketch-based shape retrieval based on the

B. Li
School of Computing Sciences and Computer Engineering, University of Southern Mississippi, Long Beach, MS, USA
E-mail: bo.li@usm.edu

J. Yuan
School of Computing Sciences and Computer Engineering, University of Southern Mississippi, Long Beach, MS, USA
E-mail: juefei.yuan@usm.edu

Y. Ye
Department of Computer Science, Texas State University, San Marcos, USA
E-mail: ricky.ye14@gmail.com

Y. Lu (*correspondence author)
Department of Computer Science, Texas State University, San Marcos, USA
Tel.: +512-245-6580
Fax: +512-245-8750
E-mail: lu@txstate.edu, luyijuan2@gmail.com

C. Zhang
School of Computing Sciences and Computer Engineering, University of Southern Mississippi, Hattiesburg, MS, USA
E-mail: chaoyang.zhang@usm.edu

Q. Tian
Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA
E-mail: qi.tian@utsa.edu

benchmark. We investigate 3D sketch and model matching problems and propose a novel 3D sketch-based model retrieval algorithm CNN-SBR based on Convolutional Neural Networks (CNNs) and achieve the best performance in the SHREC track. We wish that the 3D sketching system, the 3D sketch-based model retrieval benchmark, and the proposed 3D sketch-based model retrieval algorithm CNN-SBR will further promote sketch-based shape retrieval and its applications. We have made all of these publicly available on the project homepage: `http://orca.st.usm.edu/~bli/SBR16/project.html`.

Keywords 3D sketching · Kinect · Sketch-based 3D model retrieval · Convolutional Neural Networks

1 Introduction

Content-based 3D shape retrieval [66] is important for many various related applications such as computer-aided design (CAD), 3D movie and game production, augmented reality (AR), virtual reality (VR), and 3D printing. Given a query which is often a 2D sketch/image or a 3D model, content-based 3D model retrieval is to retrieve relevant 3D models (typically only single object models) coming from the same category as the query based on a similarity/distance metric. Effectiveness, efficiency, and scalability are the three most important performance aspects, which can be measured by a set of retrieval performance evaluation metrics [59] [39] that are commonly used in the field of information retrieval.

Due to the intuitiveness, convenience and potential for related applications, sketch-based 3D shape retrieval has received a lot of attentions from both inside and outside of the community of 3D shape retrieval. As we know, as a universal form of communication to depict the visual world, sketching has been used by human beings since tens of thousands of years ago [22]. Nowadays, sketching has become one of the most natural and popular ways to provide a visual search query for retrieval, e.g., one can search images [15], videos, and 3D models [39] [38] [18] [16] [17] on the Internet by sketching an object or scene on a touch screen phone/tablet.

However, the type of sketch-based 3D shape retrieval often only involves 2D sketching in a two-dimensional space (i.e. a paper or screen), rather than drawing a 3D sketch (3D skeleton specifically for us) in a three-dimensional space, which we refer to as “3D sketching”. By limiting a sketch to only two dimensions, we also lose most 3D information that the shape can communicate, thus creating a huge semantic gap between the iconic representation of a 2D sketch and the accurate 3D coordinate representation of a model. Due to this semantic gap, 2D sketch-based shape retrieval becomes a very challenging task [39] [38].

In the hope of bridging this semantic gap, why not considering 3D sketching such that we can provide a 3D sketch query and perform 3D sketch-based shape retrieval? If provided with a convenient 3D sketching human-computer interface to draw a 3D sketch query, we have a better chance to achieve significant improvement in the performance of sketch-based 3D model retrieval. This is because inherently both 3D sketches and 3D models are represented by 3D coordinates, that is, these two representations come from the same domain. A 3D sketch query is typically composed of a set of salient 3D feature lines of the shape that the 3D sketch depicts. Therefore, compared with a 2D sketch, a 3D sketch explicitly encodes much more 3D information, such as the depth information and features of more facets of the shape. However, we find that there is a lack of comprehensive research in 3D sketching interfaces that allow users to sketch 3D shapes in a 3D space, 3D sketch understanding, 3D sketches-3D models matching, as well as 3D sketch-based applications.

In this paper, we perform a new study of 3D sketching technology and develop a 3D sketching system which allows users to sketch an object in the air by tracking the user's hand movement with a Microsoft Kinect. Please note that our idea is **general**, thus other 3D motion controller devices are also applicable for this project, as long as they can detect the 3D location of our hands' trajectories and also can receive voice commands to avoid mouse operations during drawing. By utilizing the 3D sketching system, we collected a 3D sketch dataset, built a 3D sketch-based shape retrieval benchmark, and organized a Eurographics Shape Retrieval Contest (SHREC) track on 3D sketch-based shape retrieval [35] based on the benchmark. Further, to investigate 3D sketches-3D models matching, we propose a basic 3D sketch-based shape retrieval system [35] based on 3D shape histogram [1]. We also propose a more advanced and novel 3D sketch-based shape retrieval system CNN-SBR [73] based on Convolutional Neural Networks (CNNs). We participated in the aforementioned SHREC track based on our learning-based method CNN-SBR and won the *First Place* among all the learning-based participating approaches while its performance is also significantly better than any of the non-learning based participating approaches. We believe that the 3D sketching system, the 3D sketch-based shape retrieval benchmark, and the proposed 3D sketch-based shape retrieval algorithm CNN-SBR will further promote sketch-based 3D shape retrieval and its applications. We have made all of these publicly available on the project homepage[1].

This paper is extended from our previous work on 3D sketching technology and 3D sketch-based shape retrieval [36] [73] [35]. To the best of our knowledge, we are the first to explore 3D sketching [36] in a free 3D space and to develop an innovative retrieval system that enables users to search 3D models based on human hand-drawn 3D sketches. Since 3D sketching allows us to have more direct communication based on users' drawings, it has broad impact on many related applications, such as sketch-based shape retrieval and other human sketch related applications, including hand gesture recognition for virtual reality applications, and virtual try-on systems for clothes, glasses and watches. The application background also serves the second motivation of this project since it involves a new interaction mode: rather than using pencil and paper, we use our hands directly. The amazing thing for this is its convenience and reachability since even kids can also do 3D sketching. The main contributions of our work are highlighted as follows:

- We propose and implement a novel **3D sketching system**, which allows users to freely draw 3D sketches in the air (a real 3D space). We also collect the first human 3D sketch dataset based on it.
- To study and tackle the new problem of 3D sketches-3D models matching, a **basic** 3D sketch-based shape retrieval system is first introduced.

[1] `http://orca.st.usm.edu/~bli/SBR16/project.html`

- To deal with this challenging matching problem, a more **advanced** and novel 3D sketch-based shape retrieval system CNN-SBR is further proposed based on Convolutional Neural Networks.
- **Comprehensive experiments** have been conducted to evaluate our CNN-SBR algorithm and the experimental results demonstrate the state-of-the-art performance of our approach.

2 Related Work

2.1 Sketching

Sketches can be artistic sketches, drawings, diagrams, schematics, and blueprints [29], created for quick but informal communication via abstract and uncertain elements which may have ambiguity in their interpretation and different meanings in terms of semantics.

Sketching has a very long history [22] and is a traditional way for people to record, demonstrate, and develop an idea. There is a universal capacity for visual communication in humans [48]. That is why oracle bone script was used by primitive people to tell stories in prehistoric times and even nowadays there is a whiteboard in every meeting room. Sketching is easier and quicker for communication: a complicated shape can be perceived and imagined with only a few strokes. Compared with text, sketches are incredibly intuitive to humans and offer inherently fine-grained visual descriptions in the context of image retrieval [75].

With the proliferation of touch-screen devices, such as smart phones, tablets and smart watches, we see the popularity of the research in hand-drawn sketches and its related applications including sketch recognition, sketch-based image retrieval (SBIR), sketch-based 3D model retrieval, and sketch-based interface for modeling (SBIM). For example, Li et al. [41] proposed detecting visual attributes of shoe sketches and images at part-level for fine-grained (i.e. within the same category, like shoes) sketch-based image retrieval (FG-SBIR). Unfortunately, its experiments and evaluations were only performed on a shoe dataset.

However, none of the above involves 3D sketching and 3D sketch dataset. Facilitated by a Microsoft Kinect, we are the first [36] to study 3D sketching and propose a 3D sketching system to collect 3D sketches and perform 3D sketch-based 3D shape retrieval.

2.2 Sketch Datasets

Most early sketch datasets are small-scale collections including artistic drawings [62], professional CAD figures [46], and specific domain structure sketches [49] [44]. Recently, a large-scale collection of free-hand sketches, the TU Berlin sketch dataset [14], is open to the public, which contains 20,000 single-object sketches in 250 daily object categories. Here is a list of brief survey of several available sketch datasets.

(1) Snograss and Vanderwart's standard line drawings (1980) [61] contains 260 standard line drawings for experiments in cognitive psychology.

(2) Cole et al.'s line drawing benchmark (2008) [9] is composed of 12 line drawings of bones, mechanical parts, tablecloths and synthetic shapes, together with corresponding 3D models, for the relationship study between human-drawn sketches and computer graphics feature lines.

(3) Yoon et al.'s sketch-based 3D model retrieval benchmark (2010) [74] has 250 2D sketches and 260 watertight models [68], divided into 13 classes for sketch-based shape retrieval and evaluation.

(4) Eitz et al.'s sketch-based shape retrieval benchmark (2012) [16] collects one sketch per model for all the 1814 models existing in the well-known Princeton Shape Benchmark (PSB) [59].

(5) Eitz et al.'s sketch recognition benchmark (TU Berlin sketch dataset, 2012) [14] is the currently largest and most comprehensive 2D sketch recognition benchmark. It contains 20,000 sketches, uniformly divided into 250 categories.

(6) Hu et al.'s sketch-based image retrieval benchmark (Flickr15K dataset, 2013) [24] complies 15K Flickr images and 330 sketches over 33 categories to evaluate sketch-based image retrieval algorithms.

(7) Disney portrait dataset (2013) [7] is composed of 672 portrait sketch images drawn by seven artists at four abstraction levels.

(8) Huang et al.'s sketch segmentation and labeling benchmark (2014) [26] comprises 300 sketches, uniformly divided into 10 classes. Each sketch (i.e. human) has been manually segmented into components and each component has been assigned a label (i.e. head, hand, body and foot). The benchmark also contains 401 relevant 3D models.

(9) Li et al.'s shoe sketch dataset (2016) [41] has 304 images and 912 sketches, each of which is annotated with semantic parts and part-level attributes.

(10) Sangkloy et al.'s Sketchy dataset (2016) [53] is created for fine-grained sketch-based image retrieval. It has 75,471 sketches for 12,500 objects belonging to 125 categories.

According to our knowledge, all of the currently available sketch datasets, including above ones, are for 2D sketches and none of them involves 3D sketches. We are the first to build a 3D sketch dataset in this manner.

2.3 Sketch Recognition

Previously, sketching is introduced as a human-computer interaction (HCI) technology [65] [23], where users use a mouse or pen to draw lines and curves based on a graphical user interface. Before 2014, hand-crafted low-level features, including stroke length, stroke order and even stroke orientation, were proposed to understand human sketch input at a semantic level. A simplified Histogram of Gradients (sHOG) feature coupled with the Bag-of-Words (BoW) method [14] has demonstrated a promising sketch recognition accuracy of 56% based on the TU Berlin sketch dataset [14]. Sun et al. [64] proposed to mine object and shape topics existing in retrieved clipart images for a query sketch by utilizing a probabilistic topic model. In 2014, Schneider [54] achieved human-like sketch recognition accuracy (68.9% vs. 73.1% for human on the TU Berlin dataset) based on Fisher Vectors which represents an image by pooling local image features. However, all of the aforementioned methods involve designing a hand-crafted feature for the sketch recognition purpose, while no single "designed" feature previously proposed by people in this field has ever performed generally well for different types of sketches, including sketching styles and different levels of abstraction. This poses a natural limit on the performance of sketch recognition and its possible extensive applications in our daily life.

After 2014, machine learning, especially deep learning, based sketch recognition approaches have dominated the field of sketch recognition. Li et al. [42] proposed a Multiple Kernel Learning (MKL) framework to fuse several low-level sketch features and high-level attribute features for sketch recognition and employed star graph ensemble matching to address the structure complexity of sketches. Recently, Yu et al. [76] [77] proposed a deep neural network named "Sketch-a-Net" for hand-drawn sketch recognition which outperforms human beings and achieves an accuracy of 74.9% on the TU-Berlin sketch dataset [14]. Deep convolutional neural networks have been utilized in complete [55] or partial [13] sketch recognition, which achieve an accuracy of 75.42% and 77.69% respectively on the same TU-Berlin dataset. While, an extended version has been proposed by the same group [56], which can achieve an accuracy of 79.18%. Li et al. [43] proposed a general free-hand sketch synthesis algorithm which can automatically summarize the stroke composition of a given category and discover semantic parts from stroke data. A recurrent neural network (RNN) representation has been proposed by the Google Brain [21] for generating sketch drawings.

Our proposed Convolutional Neural Network-based 3D sketch-based shape retrieval algorithm CNN-SBR also involves 2D sketch recognition since we transform the problem of direct classification of 3D sketches into 2D sketch views classification. We utilize the aforementioned TU Berlin dataset [14] for training our CNN as well. Thus, it is related to and also can be applied to sketch recognition.

2.4 Sketch-Based Modeling

With the developments of 3D data acquisition tools and multi-media storage and processing techniques, more and more 3D digital media contents are available to us. 3D models become ubiquitous and have been widely used in many fields such as game, movie, medicine, biology and architecture. Scientists also create 3D models for visualization and engineers use 3D models to design new styles for industrial products such as vehicles and computer mice. Usually, a 3D model is a collection of points in 3D space, connected by triangles, lines, curved surfaces, etc.

We can create a 3D model using many 3D modeling techniques. Basic 3D modeling techniques include polygonal modeling, NURBS modeling, constructive solid geometry, implicit surface and subdivision surfaces and procedural modeling. Beside these basic 3D modeling techniques, we have another four advanced 3D modeling techniques: geometric and solid 3D modeling, multi-view images-based 3D modeling, and sketch-based 3D modeling.

Traditional modeling software is so complicated that only professional users can use them to create models in a given time. However, sketch-based 3D modeling systems aim at providing more intuitive and user-friendly interfaces to express our thoughts simply through sketching. Therefore, they will be accessible for novice users, even children. The key issue of designing sketching system is to infer missing information such as depth. Sketching systems also need to handle ambiguity in users' input. Many sketch-based 3D modeling systems have been proposed since the pioneer work of Zeleznik et al. [78]. Usually these sketch-based modeling systems have simple interfaces and are easy to learn and to use. A comprehensive survey has been presented in [48] and [12], respectively.

Igarashi et al. [27] introduced a sketching interface named Teddy to quickly and easily construct free-form models that have spherical topology. It uses free-form strokes as an expressive design tool to specify the silhouette of an object. Nealen et al. [47] aimed at surface detail preservation by sketching significant shape details on already existing coarse or detailed shapes. Bae et al. [4] proposed a 3D curve sketching system named ILoveSketch which facilitates users to iterate on concept 3D curve models during a design process. Shao et al. [58] generated shaded concept sketches by leveraging properties of designer-drawn cross sections to automatically infer and propagate 3D normals everywhere on a sketch. Jung et al. [31] presented the first sketch-based 3D modeling method for developable surfaces with pre-designed folds (i.e., garments or leather products) by introducing a new zippering algorithm for progressive identification of silhouette edges and their binding to silhouette strokes. A sketch-based 3D modeling system called SecondSkin targeting layered 3D model construction was proposed by Paoli and Singh [50]. Huang et al. [25] proposed a sketch-based procedural model construction approach by automatically computing a set of procedural model parameters based on a deep Convolutional Neural Network (CNN) which maps sketches to procedural model parameters after training on a large number of synthetic line drawings. Sketch-based interfaces and modeling (SBIM) have been applied into many application fields such as tree modeling, flower modeling, and cloth modeling. Recently, an important and promising trend in this research axis is to explore VR-based 3D sketching for 3D modeling [2, 28].

Among all the aforementioned 3D modeling techniques, sketch-based 3D modeling is one of the most important, intuitive and popular ones. While, different from previous sketch-based 3D modeling techniques that separate sketch-based interfaces and modeling (SBIM) and sketch-based 3D model retrieval (SBR) from each other, our 3D sketching and retrieval system combines SBIM and SBR together, thus being more concise, versatile and comprehensive. 3D sketches drawn by users in our 3D sketching system may also be different from 3D skeletons of 3D objects being drawn by users in previous sketch-based 3D modeling systems.

2.5 Sketch-Based 3D Model Retrieval

Sketch-based 3D model retrieval targets on retrieving 3D models given a hand-drawn query sketch. Recently, sketch-based shape retrieval has attracted much attention since it can be widely used in sketch-based rapid prototyping, recognition, mobile 3D search, 3D printing, 3D animation production and etc. Many related algorithms that use a 2D hand-drawn sketch as a query have been proposed [39] [38] [18] [16] [17]. A skeleton sketch-based 3D articulated model retrieval approach is proposed in [52]. A series of Shape Retrieval Contest (SHREC) tracks on this topic have been held in conjunction with the Eurographics Workshops on 3D Object Retrieval (3DOR) between 2012 and 2016. Some new benchmark datasets have been built and released to the public, such as the large scale SHREC'13 Sketch Track Benchmark (SHREC13STB) [37] which contains 7,200 2D sketches and 1,258 3D models of 90 classes, and the large scale SHREC'14 Sketch Track Benchmark (SHREC14STB) [39] that has 13,680 sketches and 8,987 models of 171 classes. Due to the semantic gap between the two different representations of rough sketches and accurate 3D models, sketch-based shape retrieval remains one of the most challenging research topics in the field of 3D model retrieval. In order to bridge the gap, we propose a 3D sketching solution and develop two 3D sketch-based 3D model retrieval systems that use human 3D sketches as query, and they are described in detail in Section 5.

Recently, deep Convolutional Neural Networks (CNNs) have shown promising results in many vision recognition tasks in different domains. CNN was introduced in early 1980s, and was applied to solve simple and small vision recognition tasks like handwritten digit recognition [33]. Currently, Wang et al. [69] learned two Siamese Convolutional Neural Networks (CNNs) for sketch-based shape retrieval: one for the views of 3D models, one for sketches. They connected the two CNNs via a loss function to compute cross-domain similarities. Similarly, a Cross-Domain Neural Network (CDNN) and its extended version Pyramid CDNN have been proposed in [79]. An octree-based Convolutional Neural Network (O-CNN) [70] is proposed to represent a 3D shape and it supports various CNN structures and has a quadratic space and time complexity. While, Xie et al. [72] devised a barycentric representation of 3D shapes for sketch-based 3D shape retrieval, which outperforms the state-of-art algorithms on the SHREC'13 [37] [38] and SHREC'14 [40] [39] sketch track benchmarks SHREC13STB and SHREC14STB. Recently, Dai et al. [10] proposed a deep correlated holistic metric learning (DCHML) approach which jointly trains two distinct deep neural networks, one for each domain, with one loss function to depict both inter-class differences and intra-class variations. It has achieved the state-of-the-art performance on SHREC'13 [38] [37] and SHREC'14 sketch [40] [39] track benchmarks, and also outperforms some top algorithms, such as CNN-Siamese [35] on the latest SHREC'16 sketch track dataset [35]. However, there is still much room for further improvement in the retrieval performance. For example, even for the top-performing algorithm DCHML, the average precision on the SHREC'16 sketch track dataset is only 0.147. A general CNN-based model trained on edge maps that can handle multiple tasks including both general and fine-grained sketch-based image retrieval, as well as domain generalization, has been proposed in [51]. Recently, similarity search in 3D human motion data has raised a lot of attentions [5, 6, 57] due to its many application scenarios in VR, AR, MR and Internet of Things (IoTs). It deals with the search, matching, and classification based on 3D human motion data. Giunchi et al. [20] proposed to search appropriate 3D models for 3D scene completion by immersive 3D sketching within a virtual environment. They focused on precise immersive sketching to distinguish between similar objects within a large class of objects, that is, fine-grained retrieval. Also, they enabled sketching over existing models by utilizing a VR equipment, rather than sketching in a VR-free environment, like us. Similar to Su et al. [63], they utilized the standard VGG-M [8] model-based multi-view CNN network to recognize a colored 3D sketch based on its multiple views: first extract the local features for each view based on five convolutional layers, then aggregate the local features across multiple views by element-wise max pooling, finally feed the pooling results into three fully connected layers for classification. However, different from both Giunchi et al. [20] and Su et al. [63], we directly classify each view first using a CNN network pre-trained on a large 2D sketch dataset, and then use majority voting to combine the multiple classification results for a 3D sketch for ranking and retrieval.

In this paper, different from almost all of the above algorithms (only excluding Giunchi et al. [20]) that focus on 2D sketch-based shape retrieval, we propose the idea of 3D sketch-based shape retrieval and devise two retrieval systems for this new research topic. This novel work opens and establishes a new direction in dealing with the challenging research topic of sketch-based shape retrieval.

3 3D Sketching

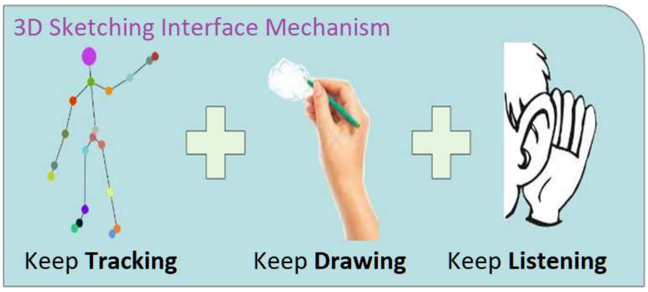
As mentioned in Section 1, a user-friendly 3D sketching interface is important and required to provide a 3D sketch query for the 3D sketch-based shape retrieval system. By utilizing such 3D sketching interface, users can easily depict the shape of a 3D object by drawing several salient 3D feature lines in a 3D space by using a tool, such as a digital pen or just our hands directly. Unfortunately, there are few existing such interfaces available for us to use. However, we have Microsoft Kinect (or other similar motion sensing input devices) which can serve as a perfect platform for us to develop one such interface enabling 3D sketching in the air.

Please note that we prefer Kinect over Leap Motion due to at least its following two limitations for our project. (1) Its usable range (1 to 20 inches) is much smaller than that of Kinect (1.2m to 3.5m), which puts a much stricter limit on the allowable space for our hands' movement. (2) As mentioned in Section 1, voice functionality is required to avoid mouse operations during drawing since we often require to send voice commands for starting the drawing, changing viewpoints, pausing, showing more results, etc. However, Leap Motion does not integrate any voice sensor and related API library for development purposes, which means an additional voice input processing system is required if we adopt it for our 3D sketching interface.

As illustrated in Fig. 1, as one of the most popular and low-cost motion sensing input devices, a Kinect device provides us with the following three features to utilize during 3D sketching: (1) Keep Tracking: a built-in RGB video camera which allows users to closely monitor their hand movement in the air; (2) Keep Drawing: a depth sensor to track the 3D locations of a user's hand movement to generate the 3D sketch, and (3) Keep Listening: a multi-array microphone which has voice recognition capacity such that it can receive voice commands from the users, thus avoiding mouse operations during drawing.

As shown in Fig. 2, a voice-activated Kinect-based 3D sketching Graphical User Interface (GUI), which supports tracking and visualizing the trajectory of a user's hand movement, is developed to help the sketching and retrieval. Besides tracking, the proposed 3D sketching interface always listens to the following five sets of voice commands: (1) drawing process control: "start", "pause", "resume", "reset", and "search" commands, which allow a user to initiate, pause and continue the current sketching while drawing, or just restart a new sketching, and finally perform search after finishing drawing a 3D sketch query; (2) hand change: switch between "left hand" and "right hand" for convenience during 3D sketch drawing; (3) view display: switch between "front view" and "side view" to easily perceive a 3D sketch to help and continue the drawing; (4) sketch display mode: "point mode" or "line mode" for sketch viewing; (5) results browse: displaying next set of results by saying "show more results" or simply "next". According to our experiments, all the aforementioned operations can be performed in real-time and we also have

(a) a user is drawing in the air



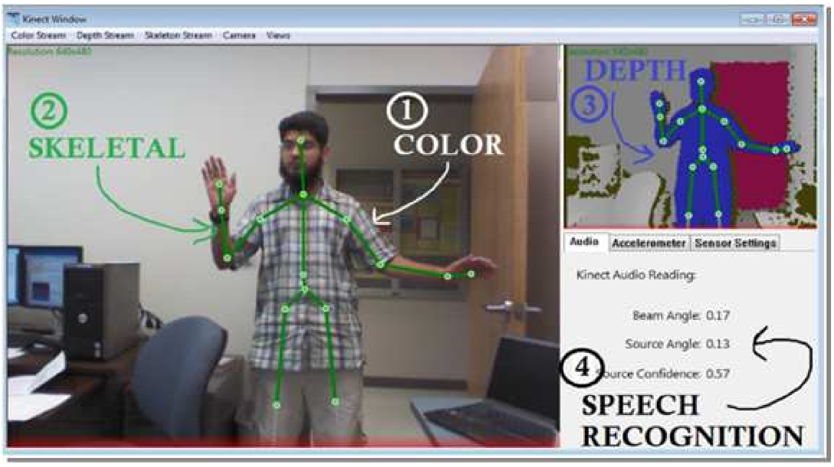
(b) working mechanism of our 3D sketching interface

Fig. 1 3D sketching idea and working mechanism.

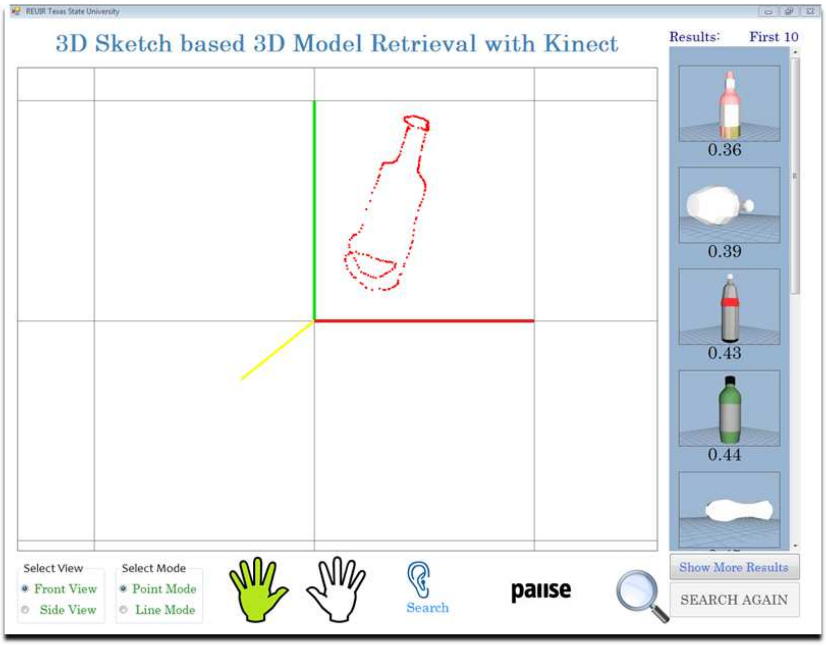
found that these real-time interactions significantly help a user in drawing a 3D sketch and improve their overall drawing experience.

3.1 Kalman Filtering

As we know, we can utilize the Microsoft Kinect to track the 48 important joints of a user's body. However, we have found that usually the captured hand motion data contain a lot of noise, though we have found that using a Kinect for Windows v2 is much better. This is mainly due to the fact that it is very difficult for the users to maintain a very stable and smooth hand movement while drawing in the air. They often shake their hands, resulting a hand trajectory which contains a lot of unnecessary outliers and small zig-zag line segments. These types of noise will have significant impact on the 3D model retrieval performance. In order to produce a smooth 3D sketch, an efficient Kalman filter [32] [45] [71] based noise removal algorithm is developed and applied on the 3D sketches. As an optimal



(a) GUI of the Kinect tracking subsystem



(b) GUI of the retrieval subsystem (basic version)

Fig. 2 3D sketching system Graphical User Interface (GUI).

linear estimator, Kalman filter is capable of inferring parameters of interest from inaccurate and uncertain observations.

Based on an *optimal recursive data processing algorithm* [45], it takes a series of observed measurements over time and produces optimal estimates of unknown signal by incorporating all available information including data measurements, and prior knowledge of the system and devices for measurement. The implementation of the Kalman filter can be found in our project homepage. In our experiments, we find that the Kalman filter successfully filters out partial noises and predicts smoother sketch curves.

4 Dataset Collection and Benchmark Building

4.1 Kinect300 Dataset Collection

Based on the developed 3D sketching system introduced in Section 3, we have collected a 3D sketch dataset named **Kinect300**, containing 300 3D sketches, uniformly divided into 30 object categories, that is, each category has 10 3D sketches. The 30 classes are: alarm clock, basket, bed, book, candle, chair, dog, door, door handle, eyeglasses, floor lamp, fork, hat, house, key, knife, ladder, laptop, mailbox, pen, scissors, screwdriver, shovel, skateboard, spider, table, tire, tree, umbrella, and vase. We base the category selection mainly on the criteria of *popularity* (most commonly seen categories) to decide these 30 classes, while we also consider the coverage/comprehensiveness of our dataset. By using a majority-vote among five people (three undergraduate student and a postdoc voters, and a faculty moderator), we choose the most popular 30 classes from all the available 250 common object labels in the TU-Berlin human sketch dataset [14], which serves as the currently most comprehensive one. One example per class is demonstrated in Fig. 3. **Kinect300** avoids the evaluation bias w.r.t data unbalancing problem since we collected the same number of sketches for every class, while the sketch variation within one class is also adequate enough. During the stage of 3D sketch data collection, we found 17 volunteer student users of our 3D sketching system. They include 4 females and 13 males, while their average age was 21 then, and only two males had a background in art. Each of the 17 users drew a set of sketches for several categories.

4.2 SHREC'16 3D Sketch-Based 3D Shape Retrieval Benchmark (SHREC16STB)

Based on the **Kinect300** dataset and SHREC'13 Sketch Track Benchmark (**SHREC13STB**) [37], we built a 3D sketch-based shape retrieval benchmark (**SHREC16STB**) and organized the Eurographics 2016 Shape Retrieval Contest (SHREC) track on 3D sketch-based 3D shape retrieval [35].

Query 3D sketch dataset: The query 3D sketch dataset contains all the 300 3D sketches (split into 30 classes, 10 sketches for each) available in **Kinect300**.

Target 3D model dataset: The target 3D model dataset of the SHREC'13 Sketch Track Benchmark (**SHREC13STB**) [37], which has 1,258 models categorized int 90 classes, is utilized to form the target 3D model dataset of our **SHREC16STB** benchmark. Fig. 4 demonstrates some examples.

The reason that we chose SHREC16STB to form the target dataset of this proposed 3D sketch-based 3D shape retrieval benchmark SHREC16STB is that the large-scale SHREC'13 Sketch Track Benchmark (SHREC13STB) is the currently available largest 2D sketch-based 3D model retrieval benchmark. However, the set of the 90 classes that are available in SHREC13STB is not a superset of the set of the 30 classes available in Kinect300. There are 9 classes coming from Kinect30 which do not appear in the 90 classes of SHREC13STB. That is to say, only 21 of the 30 classes in **Kinect300** can find their relevant models in the target 3D model dataset of this benchmark **SHREC16STB**. Therefore, when computing

Fig. 3 Example 3D sketches (one example per class, shown in one view) of our **Kinect300** dataset.

Fig. 4 Example target 3D models of our **SHREC16STB** benchmark.

the retrieval performance, we only consider the results of these 21 classes. The 9 classes that have no relevant 3D models are: alarm clock, basket, candle, door handle, eyeglasses, fork, key, pen, and scissors.

To help evaluate learning-based retrieval approaches as well, for each class we randomly pulled out 7 sketches for training and the remaining 3 sketches are used for testing, while all the 1,258 target models are still kept as a whole to serve as the target 3D model dataset.

4.3 Evaluation Method

To comprehensively evaluate the performance of the proposed 3D sketch-based shape retrieval systems, we employ the following seven evaluation metrics [59] [39]: Precision-Recall (PR) diagram, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-Measures (E), Discounted Cumulated Gain (DCG) and Average Precision (AP). They are commonly used in the field of 3D object retrieval as well as information retrieval. We have made the SHREC16STB benchmark, together with its evaluation toolkit, publicly available online [34].

5 3D Sketch-Based 3D Model Retrieval Systems

Although a 3D sketch has provided us with a better and more comprehensive description of the object than a 2D sketch, there is little related research work [11, 19,20] in 3D sketch understanding including recognition and classification, even less in 3D sketches-3D models matching. In this section, we propose two 3D sketch-based shape retrieval systems. Section 5.1 presents the basic non-learning one, which serves as our novel work as well as a baseline approach in this aspect and it is also integrated into our 3D sketching system, as demonstrated in Fig. 2 (b). Section 5.2 focuses on the best performance, which represents our advanced and learning-based work on this research topic.

5.1 Basic Version: 3D Shape Histogram-Assisted 3D Sketch-Based 3D Model Retrieval System (3DSH)

Based on the above 3D sketching platform and 3D shape histogram [1], we build a simple and basic, yet efficient 3D sketch-based shape retrieval system, named 3DSH. As demonstrated in Fig. 5, our retrieval system is composed of offline and online processes and it has three major components which are 3D outline generation, feature extraction, and 3D sketches-3D models matching.

(1) 3D outline generation. The objective of this stage is to generate a 3D outline for each target 3D model. Firstly, we normalize each 3D model by first aligning it based on Principle Component Analysis (PCA) [30], and then translating the center of the model's bounding sphere to the origin, and finally scaling the model to make the radius of its bounding sphere be 1. Please note that the same normalization process will be applied on an online hand-drawn 3D sketch (Fig. 5 (h)). Secondly, we integrate all the 3D contour points of its six principle views (that is front/back, left/right, and top/bottom views) to generate a 3D outline for the model, as demonstrated in Fig. 5 (b) for a bicycle model. Thirdly, to improve the robustness of our algorithm w.r.t different resolutions of 3D outlines and sketches, an approximately uniform point sampling is applied on the 3D outline by setting a threshold for the distance between any two 3D points in the outline, as shown in Fig. 5 (c) for the final 3D outline of the bicycle model.

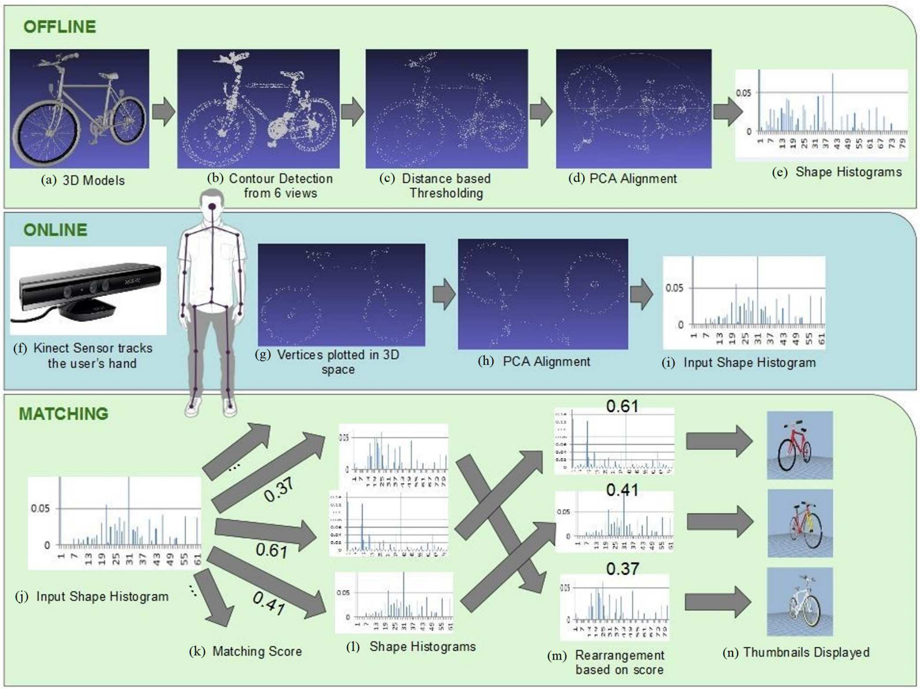
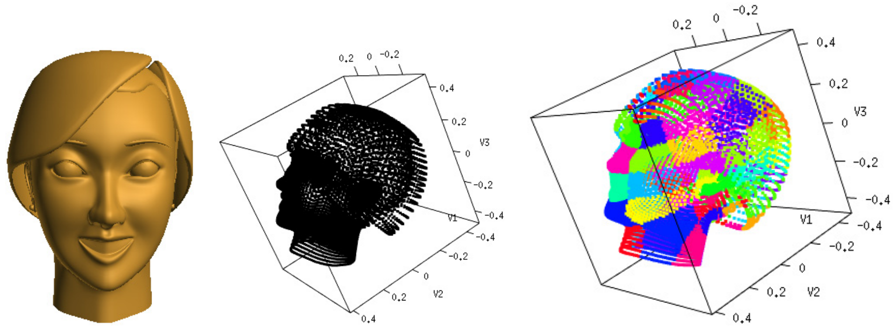


Fig. 5 3DSH's system framework.

Finally, a PCA alignment is applied (Fig. 5 (d)) on the 3D outline to make it ready for extracting the rotation-dependent shape descriptor 3D shape histogram.

(2) Feature extraction. Most existing shape descriptors, such as the local features proposed in [60], target a complete mesh model, rather than a 3D sketch model, which is substantially a sparse set of 3D points. Here, we investigate using the 3D shape histogram [1] to characterize both 3D models and 3D sketches considering its *descriptiveness*, *high efficiency*, and *simplicity*. The 3D shape histogram descriptor depicts a 3D model based on a histogram feature generated by calculating the percentages of the model's vertices falling in a set of pre-defined shell, sector or spiderweb bins, which partition the space occupied by the model from the center of the model. Shell bins are defined by the different radii of a set of concentric shells surrounding the center of a model; sector bins are defined based on the idea of uniformly distributing the number of vertices of a 3D model into a set of different surface regions of a surrounding sphere of the model. Decomposition of regular polyhedrons and Voronoi diagrams [3] are utilized for the space partition. While, the spiderweb bins are defined by a combined utilization of both the Shell and the Sector models, resulting the Combined model which provides a finer-grained 3D decomposition for the model's 3D shape histogram feature extraction. One visualization example for the Spiderweb model-based 3D shape histogram feature is shown in Fig. 6. Considering the inherent nature of the representations for a 3D sketch and a 3D outline, as well as the efficiency issue, for each 3D sketch or outline, we extract its 3D shape histogram [1] descriptor based on the following spiderweb model: 20 shells, 6 sectors, that is, 120 bins in total, as shown in Fig. 5 (e).



(a) 3D model (b) point mode (c) feature visualization

Fig. 6 Visualizing the Spiderweb model-based 3D shape histogram (3DSH) feature of an example model (m349) in the Princeton Shape Benchmark (PSB) [59] dataset. Different spiderweb bins are colored differently to differentiate from each other.

(3) 3D sketches-3D models matching. We measure the similarity between the 3D sketch query and each target 3D model based on the Euclidean distance (Fig. 5 (k)) between their 3D shape histograms generated based on their 3D outlines. We sort all the distances in an ascending order (Fig. 5 (m)) and finally rank the target 3D models accordingly (Fig. 5 (n)).

5.2 Advanced Version: Convolutional Neural Network-Based 3D Sketch-Based 3D Model Retrieval System (CNN-SBR)

As shown in Section 5.1, the most straightforward way to perform 3D sketches-3D models matching is to compute their feature distances based on some hand-crafted shape descriptors for both 3D sketches and 3D models. However, it turns out such kind of approach cannot achieve top retrieval accuracy, mainly due to their limited descriptive power, as demonstrated by the 3D shape histogram descriptor in Section 6.1. Fortunately, as reviewed in Section 2.5, in recent years Convolutional Neural Networks (CNNs) have demonstrated their great potentials (e.g. in terms of retrieval accuracy) and advantages (i.e. automatic feature extraction, instead of hand-crafted features) in the field of 2D sketch-based shape retrieval. Inspired by this, to further advance the retrieval accuracy in the hope of finally bridging the semantic gap between 2D sketch queries and 3D target models by conducting 3D sketch-based shape retrieval, in this section we propose a novel CNN-based 3D sketch-based shape retrieval system "CNN-SBR" by utilizing multiple state-of-the-art deep CNNs in sketch object recognition and 3D model processing techniques.

5.2.1 CNN-SBR Architecture Overview

As illustrated in Fig. 7, our CNN-SBR system is inspired by early 2D sketch-based image retrieval work. Firstly, multiple 2D sketch views are rendered for

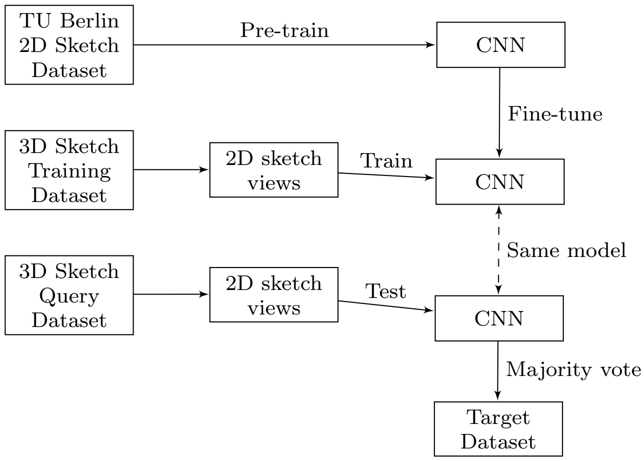


Fig. 7 CNN-SBR architecture.

each 3D sketch coming from either the training or the testing dataset, before a data augmentation technique is applied on these generated 2D sketch views. Secondly, by utilizing the much larger and more comprehensive TU Berlin 2D sketch dataset [14], we pre-train our deep CNN model such that we can populate our model with a set of well-learned initial weights. Thirdly, further fine-tune the CNN model based on the augmented 2D training sketch views. Fourthly, based on the fine-tuned CNN model, we generate the classification result for each query 3D sketch by feeding its 2D sketch views into the model. Finally, generate the rank list for the query based on a majority-vote based label matching method.

5.2.2 Data Processing

To fit a 3D sketch query into our proposed CNN-SBR framework, we need to transform a 3D sketch into a set of 2D sketch views. To do this, we generate six depth images by projecting all the coordinates of a 3D sketch onto the six faces (viewpoints) of its bounding cube by mapping the 3D coordinates to a 2D depth image, where the pixel values encode the distances between the 3D coordinates and their corresponding viewpoints: 0 indicates the nearest while 255 represents the farthest.

To avoid overfitting our CNN models, a data augmentation technique is also applied on several involved datasets. In our experiments, we expand the sizes of both the TU Berlin 2D sketch dataset and the 2D sketch view set generated from the training dataset by 500 times based on random rotations, shifts and flips. Specifically, **Algorithm** 1 details our data augmentation algorithm. By utilizing this algorithm, a sketch image will generate 500 new transformed images, which increase the variety of sketches and also significantly reduce the noises in users' hand-drawings.

Input: S: Original 2D hand-drawn sketch / sketch views dataset
Output: T: Enlarged 2D hand-drawn sketch / sketch views dataset with random shifts, rotations, and flips
Initialization;
$w = width_{original} - width_{target}$;
foreach $I \in S$ **do**
  for $i \leftarrow 1$ **to** 500 **do**
    $M \leftarrow copy(I)$;
    $x_{shift} \leftarrow random(0, w)$;
    $y_{shift} \leftarrow random(0, w)$;
    $M \leftarrow shift(M, x_{shift}, y_{shift})$;
    $roll \leftarrow random(0, 1)$;
    if $roll < 0.5$ **then**
      $rd \leftarrow random(-5, 5)$;
      $M \leftarrow rotate(M, rd)$;
    end
    $roll \leftarrow random(0, 1)$;
    if $roll < 0.5$ **then**
      $M \leftarrow flip(M)$
    end
    $append(T, M)$;
  end
end

Algorithm 1: Data augmentation algorithm.

5.2.3 CNN-SBR System

In our CNN-SBR system, Sketch-a-Net [76] [77] serves as our core CNN model for 3D sketch (reduced to 2D sketch views) recognition. Based on AlexNet, Sketch-a-Net is designed for 2D single object sketch recognition. It has five convolutional layers and three fully-connected lays. Its CNN architecture and explanations of its parameters and differences from previous well-known CNNs for images such as AlexNet can be found in [76] [77]. Since we have less number of categories, the number of neurons in the last layer will be less, for example it is 30 for our **SHREC16STB** benchmark. Considering the big number of categories (250) and diverse variation (80) within one category existing in the TU Berlin 2D sketch dataset, we use it as an ancillary training dataset and feed it to Sketch-a-Net for pre-training. Although the TU Berlin dataset is obtained from human's 2D sketching (rather than for 3D sketching), it provides Sketch-a-Net with a comprehensive set of human drawings' features, which helps build a set of initial learning weights for future fine-tuning on a 3D sketch dataset.

In fine-tuning step, we choose the pre-trained model at epoch 500 trained on the TU Berlin 2D sketch dataset, while fine-tuning it on the 3D sketch training dataset (resulting 2D sketch views) at epoch 5000.

5.2.4 Majority Vote and Label Matching

A majority vote algorithm is utilized to decide the final classification result for each 3D sketch based on its six 2D sketch views' corresponding six CNN-SBR output similarity vectors. That is, for each 2D sketch view we have a CNN-SBR classification output, which is a similarity vector for predicting its categories. Each similarity vector contains a top first label which indicates the category that has the maximum similarity value in the similarity vector. We utilize both the counts of top first labels and average similarity values to rank all the target category labels for each 3D sketch. The reason to consider average similarity values as well is because some target categories may have the same top first label count. Thus, we further consider the difference in their average similarities to rank those categories. After considering both differences, we are able to simply rank all the target categories for each query 3D sketch and accordingly list all relevant target 3D models based on their category ranks. In detail, we use the following procedure for ranking.

(1) Step 1: similarity vector scaling: we scale the similarity values in each of the six similarity vectors to make them fall into the range of $[0, 1]$. A higher value indicates bigger similarity.

(2) Step 2: top first label vector generation: for each target 3D category label, we first count its appearance on the top first among the six similarity vectors, that is, the top first label count which is an integer belonging to the range of $[0, 6]$. We thus form a top first label vector $T{=}[t_1, t_2, \cdots, t_n]$, where n is the total number of categories.

(3) Step 3: average similarity vector calculation: an average similarity vector $S{=}[s_1, s_2, \cdots, s_n]$ is calculated by averaging over the six similarity vectors to measure the similarity between the 3D sketch and all the target 3D category labels. Each similarity value in this average similarity vector is in the range of $[0, 1]$.

(4) Step 4: summation vector computation and ranking: all the target category labels are ranked based on the summation vector R of the top first label vector T and the average similarity vector S: $R{=}T{+}S$, and then all the 3D target models are ranked accordingly.

6 Experiments and Discussions

6.1 Basic Version: 3DSH Retrieval Experiments

To conduct a comprehensive evaluation on the performance of our basic retrieval system presented in Section 5.1, we conduct the following three different types of experiments.

(1) Outline-model retrieval. This experiment evaluates the performance of 3D model similarity retrieval given a perfect 3D outline, rather than a rough human 3D sketch. The target dataset (1,258 3D models, 90 classes) of our **SHREC16STB** benchmark described in Section 4.2 is used as the target 3D model dataset here, while the 3D outline queries are directly generated from all the target 3D models.

(2) Sketch-sketch retrieval. We conduct this 3D sketch similarity retrieval using a hand-drawn 3D sketch query considering the fact that the intra-class variations of human-drawn 3D sketches in the same category can be extremely high. **Kinect300** (Section 4.1) is used as both the query and the target datasets in this experiment.

(3) Sketch-model retrieval. This experiment targets our main research topic: 3D sketch-based shape retrieval, which measures the retrieval performance of searching similar 3D models based on a human 3D sketch query. Our **SHREC16STB** benchmark introduced in Section 4.2 is used for this experiment.

6.1.1 Running Time

We implemented the 3DSH system using C/C++ and performed all the above three experiments on a modern computer with an Intel Xeon X5675 @3.07 GHz CPU. Due to the high efficiency of both the Kalman filter algorithm and the 3D shape histogram-based matching, we have achieved a real-time performance for all the above three types of retrieval experiments. For instance, the average time to conduct the third type of retrieval is only 1.22 sec.

6.1.2 Evaluation Results

Fig. 8 and Table 1 show the experimental results. As one would expect, the performance of outline-model retrieval is the best, followed by that of sketch-sketch retrieval, which again beats that of sketch-model retrieval. It has been found that the 3D shape histogram descriptor has good representation capacity in describing different vertex distribution patterns among accurate 3D models or perfect outlines, but much weaker discrimination power in differentiating models from different categories. This also explains why sometimes 3D models from irrelevant categories may be returned in the front part of the rank list for a 3D outline query.

It has also been found that the 3D shape histogram feature is sensitive to noise. As can be seen, all the 3D sketches collected in **Kinect300** are both extremely abstract and very noisy due to the fact that most of the 17 users have little drawing experience, especially in 3D sketching. We have found that the sketch lines drawn by naive users are often not smooth and even non-continuous. This is because during sketching they often pause their hands, which produces many densely located noisy points. Even though we have applied Kalman filter to remove outliers and smooth the sketching, usually the query sketches still have some noise. This partially explains the relative lower performance of sketch-sketch retrieval, if compared with that of outline-model retrieval.

In fact, sketch-model retrieval is the most challenging one among the three types of experiments due to the different levels of accuracy and abstraction in the query and target datasets. In addition, though small, **Kinect300** contains diversified categories, including some challenging ones (i.e., dog and human) for people to draw a simple and compact 3D sketch. All these facts pertinent to the sketch-model retrieval type motivate us to solve this challenging but also interesting specific research problem: what is the most effective way to measure the distance between a rough and abstract hand-drawn 3D sketch and a 3D model with many more details and much more accurate 3D information? On the other hand, we have found that even based on this simple 3DSH method, we have achieved good retrieval results on many simple categories including wineglass, sword, airplanes, and balloons. This suggests that it is very promising to significantly improve the retrieval performance after further research on the above research problem. Three such examples using 3D sketches with different levels of complexity are demonstrated in Fig. 9.

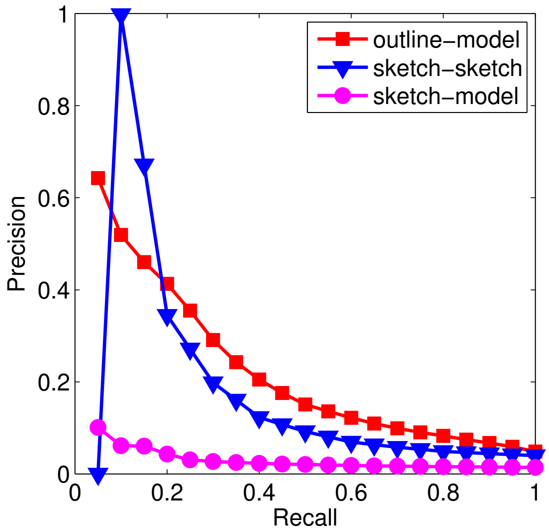


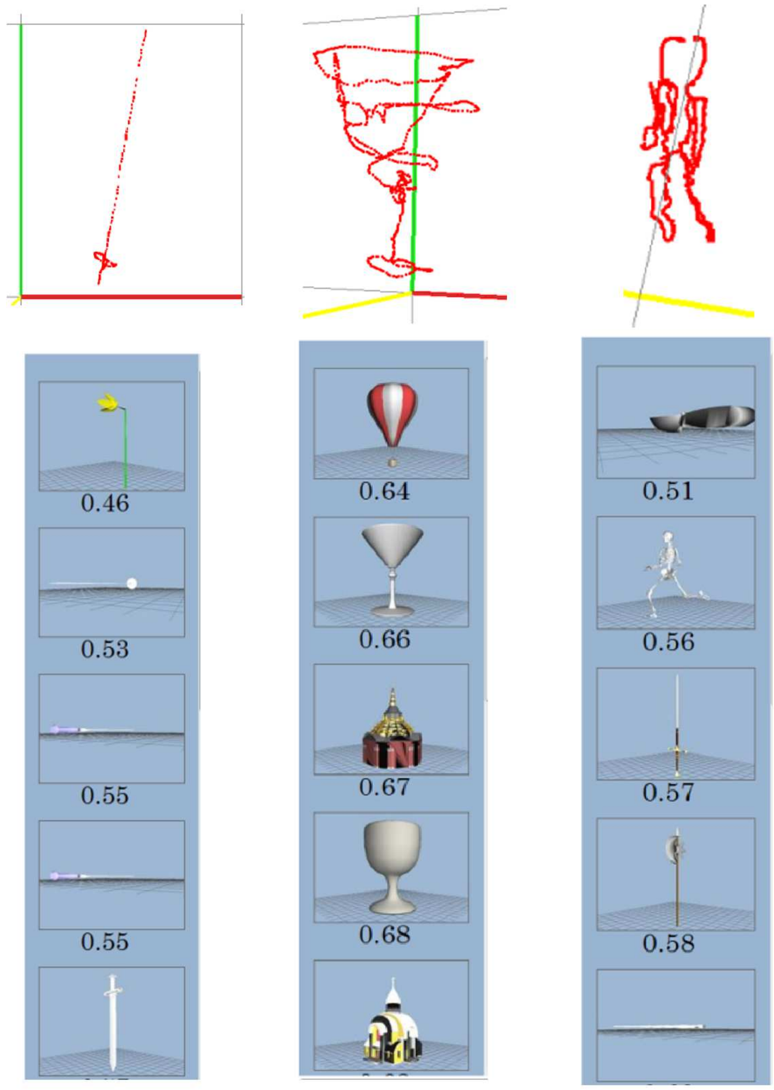
Fig. 8 Precision-Recall diagram performance of our 3DSH system.

Table 1 Other performance metrics of our 3DSH system.

Benchmark	NN	FT	ST	E	DCG	AP
outline-model	0.391	0.156	0.238	0.121	0.486	0.217
sketch-sketch	0.167	0.087	0.139	0.092	0.360	0.176
sketch-model	0.029	0.021	0.038	0.021	0.254	0.029

6.2 Advanced Version: CNN-SBR Experiments

To further advance the retrieval performance for the sketch-model retrieval, we have developed the CNN-SBR system. In order to have a comprehensive evaluation on its performance, we participated in the Eurographics 2016 Shape Retrieval Contest (SHREC'16) 3D sketch-based shape retrieval track [35] which is based on the **SHREC16STB** benchmark (Section 4.2). Six teams participated in this SHREC'16 track, while our basic 3D sketch-based shape retrieval algorithm 3DSH (Section 5.1) is also included here as a baseline non-learning based method. All the participating algorithms are evaluated either on **SHREC16STB**'s test dataset (i.e. for learning-based algorithms such as our CNN-SBR) or on its complete dataset (i.e. for non-learning based algorithms such as our 3DSH). Our CNN-SBR system achieved the *best* accuracy in terms of all the seven evaluation metrics (Section 4.3). In this section, we will conduct a comparative evaluation between our CNN-SBR system and several other participating methods in order to find out possible reasons that contribute to the state-of-the-art performance of our approach. We also compare our CNN-SBR with the most recently proposed approach **DCHML** [10].



(a) Simple (b) Medium (c) Complicated

Fig. 9 3DSH's three example retrieval results using sketches with different complexities based on the 3D sketch-based 3D model retrieval system demonstrated in Fig. 2 (b). The first row shows the 3D sketch queries, while the second row demonstrates the top five retrieval results.

6.2.1 Running Time

Our CNN-SBR system was implemented using Matlab and the MatConvNet toolbox [67]. A server with an 8-core 3.5 GHz CPU and a GeForce GTX Titan X GPU was used to perform the experiments. Based on the GPU, it took approximately 1 hour for the pre-training on the TU Berlin 2D sketch dataset and about 30 minutes for the fine-tuning on the **Kinect300** dataset. Only five minutes were needed for the majority voting and label matching process, that is the response time for each query is about 1.4 seconds.

6.2.2 Evaluation Results

Table 2 and Fig. 10 show the evaluation results. It is apparent that on all the seven evaluation metrics CNN-SBR has achieved much better performance than other learning-based participating algorithms, including the most recent **DCHML** [10] approach. CNN-SBR's performance is also much higher than non-learning based approaches. On the other hand, it is also evident that **SHREC16STB** is a challenging dataset for 3D sketch-based shape retrieval evaluation considering the still low overall performance of most methods.

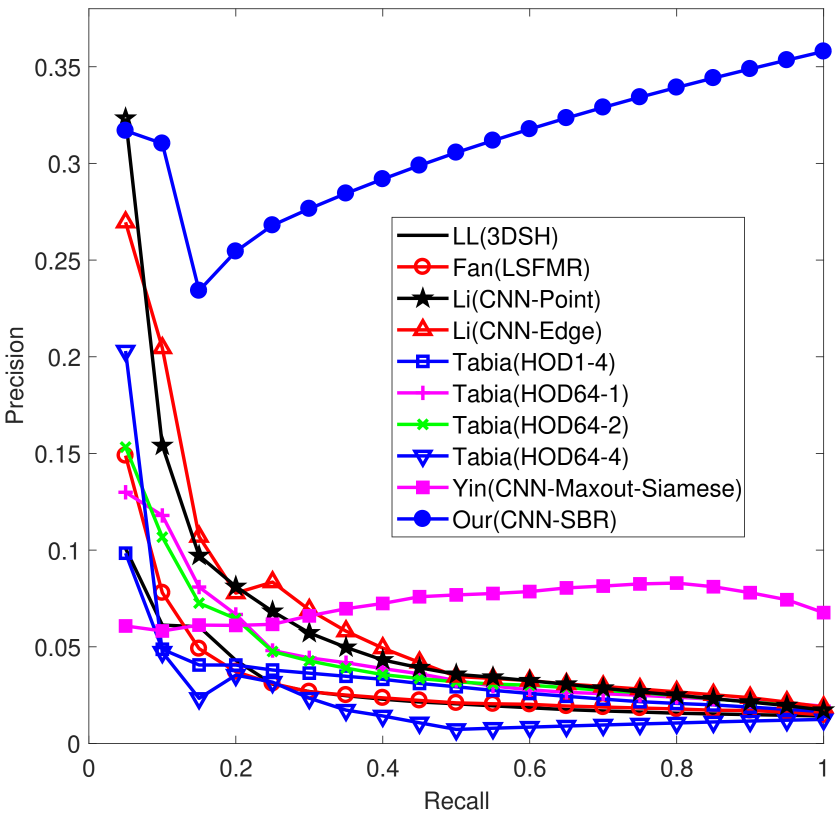


Fig. 10 CNN-SBR's Precision-Recall diagram comparison on the complete dataset of **SHREC16STB** for non-learning based algorithms and on its testing dataset for learning based algorithms.

Table 2 CNN-SBR's performance metrics comparison on the **SHREC16STB** benchmark.

Participant/Author Group	Method	NN	FT	ST	E	DCG	AP
Non-learning based methods		**Complete benchmark**					
LL	3DSH	0.029	0.021	0.038	0.021	0.254	0.029
Fan	LSFMR	0.033	0.020	0.033	0.018	0.248	0.032
Tabia	HOD1-4	0.029	0.015	0.035	0.026	0.259	0.032
	HOD64-1	0.052	0.031	0.053	0.034	0.274	0.044
	HOD64-2	0.067	0.031	0.057	0.032	0.272	0.044
	HOD64-4	0.124	0.019	0.022	0.013	0.230	0.026
Learning-based methods		**Testing dataset**					
Li	CNN-Point	0.124	0.044	0.075	0.046	0.294	0.060
	CNN-Edge	0.114	0.056	0.084	0.051	0.302	0.063
Yin	CNN-Maxout-Siamese	0.000	0.031	0.108	0.048	0.293	0.072
[10]	DCHML	0.117	0.106	0.148	0.086	0.327	0.147
Our Group	**CNN-SBR**	**0.206**	**0.249**	**0.340**	**0.196**	**0.472**	**0.310**

6.2.3 Discussions

First, we comment on other competition methods in the SHREC'16 challenge. For **DCHML**, please refer to Section 2.5. Fan's Localized Statistical Feature (LSF) and Manifold Ranking (MR) approach (**LSFMR**) is a non-learning based method which mainly consists of two components: LSF local feature extraction and manifold feature distance ranking for enhancing its retrieval performance. Tabia's Histogram of Oriented Distances (**HOD**) is also a non-learning based method which builds a dubbed histogram representing a joint distribution of distances and angles for a 3D sketch. Another two CNN-based shape retrieval approaches, Li's **CNN-Point** and **CNN-Edge**, are non-learning based methods which train a CNN based on a large number of sample views of the point cloud forms of our target 3D models. While, a query 3D sketch is first transformed to a point-based or edge-based 3D sketch, whose sample views are fed into the trained CNN to generate classification output. One more CNN-based approach Yin's **CNN-Maxout-Siamese** is a learning-based method which employs random view sampling and Siamese CNN network.

It should be noted that both learning-based participating methods (CNN-SBR and CNN-Maxout-Siamese) are based on CNN. However, compared with our CNN-SBR approach, either non-learning based or other learning-based methods still have a big performance gap to catch up.

After looking over their implementation details, we have identified the following advantages or differences of our CNN-SBR system.

- **Adaptive feature learning.** Unlike **HOD**, **3DSH** and **LSFMR** which extract conventional hand-crafted features, we utilize deep CNN to conduct automatic and adaptive feature learning.
- **Data augmentation.** A data augmentation process has been performed in our CNN-SBR system both on the TU-Berlin dataset for pre-training and the 2D sketch views of each 3D sketch during fine-tuning. By enlarging the dataset by 500 times, we significantly reduce the chance of overfitting. However, none of other CNN-based participating methods employs this type of data augmentation technique to avoid the overfitting problem, which may partially explain the much better performance of our algorithm.
- **Pre-training on TU Berlin.** We pre-train CNN-SBR based on the currently largest and also the most comprehensive sketch dataset: TU Berlin

2D sketch dataset. However, other CNN-based methods, including **DCHML**, **CNN-Point**, **CNN-Edge**, and **CNN-Maxout-Siamese**, do not employ this pre-training technique by training their CNNs on a large dataset to boost the performance, which also, in some degree, explains their relatively unsatisfactory performance even similarly based on CNNs as ours.

7 Conclusions and Future Work

3D sketching in 3D space and 3D sketch-based shape retrieval are novel, interesting and promising research topics. There exists very little preliminary work in this field, which allows us enough room to further explore and produce possibly very exciting and useful research results. In this paper, we have developed a novel 3D sketching system, based on which we collected a 3D sketch dataset and built a benchmark for 3D sketch-based shape retrieval. We also proposed two 3D sketch-based shape retrieval systems. The results of performance evaluation have demonstrated the promising potentials of our 3D sketching technique in related applications such as collecting 3D sketch data and conducting 3D sketch-based shape retrieval. Our CNN-based 3D sketch-based shape retrieval algorithm (CNN-SBR) also achieves top performance in a related Shape Retrieval Contest (SHREC) track. We believe all these will have many implications in related research and applications.

Nonetheless, there are quite a few open questions to further promote the two challenging tasks: 3D sketching and 3D sketch-based shape retrieval. Firstly, developing an even more convenient and effective human-computer interface to help people draw 3D sketches freely in a 3D space is among the top list of our future work. More advanced and specific 3D sketches-3D models matching algorithms are deserved for our further exploration as well for this specific type of retrieval. In addition, the 3D sketches currently we have collected are quite abstract and noisy. It will be rewarding to propose even better denoising and smoothing algorithms to further filter out more noisy points existing in a 3D sketch. Finally, to further promote the performance of our retrieval system, we plan to collect a large-scale 3D sketch dataset from more diverse users, and then use it to train our system for better matching between 3D sketches and 3D models.

Acknowledgments

This work is supported by Army Research Office grant W911NF-12-1-0057 to Dr. Yijuan Lu and Dr. Qi Tian, by NSF CRI-1305302, NSF CNS-1358939 and NSF OCI-1062439 to Dr. Yijuan Lu, and by the University of Southern Mississippi Faculty Startup Funds Award to Dr. Bo Li. We gratefully acknowledge the support from NVIDIA Corporation for the donation of the Titan X/Xp GPUs used in this research.

References

1. M. Ankerst, G. Kastenmüller, H. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *Advances in Spatial Databases, 6th Inter-*

national Symposium, SSD'99, Hong Kong, China, July 20-23, 1999, Proceedings, pages 207–226, 1999.
2. C. Araújo, D. Cabiddu, M. Attene, M. Livesu, N. Vining, and A. Sheffer. Surface2Volume: surface segmentation conforming assemblable volumetric partition. *ACM Trans. Graph.*, 38(4):80:1–80:16, 2019.
3. F. Aurenhammer. Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.
4. S. Bae, R. Balakrishnan, and K. Singh. Ilovesketch: as-natural-as-possible sketching system for creating 3D curve models. In *UIST*, pages 151–160. ACM, 2008.
5. C. Beecks and A. Grass. Efficient Point-Based Pattern Search in 3D Motion Capture Databases. In M. Younas and J. P. Disso, editors, *6th IEEE International Conference on Future Internet of Things and Cloud, FiCloud 2018, Barcelona, Spain, August 6-8, 2018*, pages 230–235. IEEE Computer Society, 2018.
6. C. Beecks, M. Hassani, B. Brenger, J. Hinnell, D. Schüller, I. Mittelberg, and T. Seidl. Efficient Query Processing in 3D Motion Capture Gesture Databases. *Int. J. Semantic Comput.*, 10(1):5–26, 2016.
7. I. Berger, A. Shamir, M. Mahler, E. J. Carter, and J. K. Hodgins. Style and abstraction in portrait sketching. *ACM Trans. Graph.*, 32(4):55:1–55:12, 2013.
8. K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In M. F. Valstar, A. P. French, and T. P. Pridmore, editors, *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*. BMVA Press, 2014.
9. F. Cole, A. Golovinskiy, A. Limpaecher, H. S. Barros, A. Finkelstein, T. Funkhouser, and S. Rusinkiewicz. Where do people draw lines? *ACM Trans. Graph.*, 27(3), 2008.
10. G. Dai, J. Xie, and Y. Fang. Deep correlated holistic metric learning for sketch-based 3D shape retrieval. *IEEE Transactions on Image Processing*, 27(7):3374–3386, July 2018.
11. J. Delanoy, M. Aubry, P. Isola, A. A. Efros, and A. Bousseau. 3D sketching using multi-view deep volumetric prediction. *Proc. ACM Comput. Graph. Interact. Tech.*, 1(1):21:1–21:22, 2018.
12. C. Ding and L. Liu. A survey of sketch based modeling systems. *Front. Comput. Sci.*, 10(6):985–999, 2016.
13. S. Dupont, O. Seddati, and S. Mahmoudi. Deepsketch 2: Deep convolutional neural networks for partial sketch recognition. In *CBMI 2016, Bucharest, Romania, June 15-17, 2016*, pages 1–6, 2016.
14. M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? *ACM Trans. Graph.*, 31(4):44:1–44:10, 2012.
15. M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE Trans. Vis. Comput. Graph.*, 17(11):1624–1636, 2011.
16. M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa. Sketch-based shape retrieval. *ACM Trans. Graph.*, 31(4):31:1–31:10, 2012.
17. T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3D models. *ACM Trans. Graph.*, 22(1):83–105, Jan. 2003.
18. T. Furuya and R. Ohbuchi. Ranking on cross-domain manifold for sketch-based 3D model retrieval. In *Cyberworlds (CW), 2013 International Conference on*, pages 274–281, Oct 2013.
19. D. Giunchi, S. James, and A. Steed. Model retrieval by 3D sketching in immersive virtual reality. In *IEEE*.
20. D. Giunchi, S. James, and A. Steed. 3D sketching for interactive model retrieval in virtual reality. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, Expressive '18, pages 1:1–1:12, New York, NY, USA, 2018. ACM.
21. D. Ha and D. Eck. A neural representation of sketch drawings. *CoRR*, abs/1704.03477, 2017.
22. C. S. Henshilwood, F. D'errico, K. L. van Niekerk, L. Dayet, A. Queffelec, and L. Pollarolo. An abstract drawing from the 73,000-year-old levels at Blombos Cave, South Africa. *Nature*, 562:115–118, 2018.
23. C. F. Herot. Graphical input through machine recognition of sketches. *SIGGRAPH Comput. Graph.*, 10(2):97–102, July 1976.
24. R. Hu and J. P. Collomosse. A performance evaluation of gradient field HOG descriptor for sketch based image retrieval. *Computer Vision and Image Understanding*, 117(7):790–806, 2013.

25. H. Huang, E. Kalogerakis, E. Yumer, and R. Mech. Shape synthesis from sketches via procedural models and convolutional networks. *IEEE Trans. Vis. Comput. Graph.*, 23(8):2003–2013, 2017.
26. Z. Huang, H. Fu, and R. W. H. Lau. Data-driven segmentation and labeling of freehand sketches. *ACM Trans. Graph.*, 33(6):175:1–175:10, 2014.
27. T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3D freeform design. In *SIGGRAPH 1999, Los Angeles, CA, USA, August 8-13, 1999*, pages 409–416, 1999.
28. B. Jackson and D. F. Keefe. Lift-Off: Using reference imagery and freehand sketching to create 3D models in VR. *IEEE Trans. Vis. Comput. Graph.*, 22(4):1442–1451, 2016.
29. G. Johnson, M. D. Gross, J. Hong, and E. Yi-Luen Do. Computational support for sketching in design: A review. *Found. Trends Hum.-Comput. Interact.*, 2(1):1–93, Jan. 2009.
30. I. Jolliffe. *Principal Component Analysis.* Springer Series in Statistics. Springer, 2002.
31. A. Jung, S. Hahmann, D. Rohmer, A. Bégault, L. Boissieux, and M. Cani. Sketching folds: Developable surfaces from non-planar silhouettes. *ACM Trans. Graph.*, 34(5):155:1–155:12, 2015.
32. R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82:35–45, 1960.
33. Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *NIPS, Denver, Colorado, USA, 1989]*, pages 396–404, 1989.
34. B. Li and Y. Lu. `http://cs.txstate.edu/~yl12/SBR2016/`, 2016.
35. B. Li, Y. Lu, F. Duan, S. Dong, Y. Fan, L. Qian, H. Laga, H. Li, Y. Li, P. Liu, M. Ovsjanikov, H. Tabia, Y. Ye, H. Yin, and Z. Xue. SHREC'16: 3D sketch-based 3D shape retrieval. In *3DOR*, pages 47–54. Eurographics Association, 2016.
36. B. Li, Y. Lu, A. Ghumman, B. Strylowski, M. Gutierrez, S. Sadiq, S. Forster, N. Feola, and T. Bugerin. 3D sketch-based 3D model retrieval. In *ICMR 2015, Shanghai, China, June 23-26, 2015*, pages 555–558, 2015.
37. B. Li, Y. Lu, A. Godil, T. Schreck, M. Aono, H. Johan, J. M. Saavedra, and S. Tashiro. SHREC'13 track: Large scale sketch-based 3D shape retrieval. In *Eurographics Workshop on 3D Object Retrieval, Girona, Spain, 2013. Proceedings*, pages 89–96, 2013.
38. B. Li, Y. Lu, A. Godil, T. Schreck, B. Bustos, A. Ferreira, T. Furuya, M. J. Fonseca, H. Johan, T. Matsuda, R. Ohbuchi, P. B. Pascoal, and J. M. Saavedra. A comparison of methods for sketch-based 3D shape retrieval. *Computer Vision and Image Understanding*, 119:57–80, 2014.
39. B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, Q. Chen, N. K. Chowdhury, B. Fang, H. Fu, T. Furuya, H. Li, J. Liu, H. Johan, R. Kosaka, H. Koyanagi, R. Ohbuchi, A. Tatsuma, Y. Wan, C. Zhang, and C. Zou. 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.
40. B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, H. Fu, T. Furuya, H. Johan, J. Liu, R. Ohbuchi, A. Tatsuma, and C. Zou. SHREC'14: Extended large scale sketch-based 3D shape retrieval. In *3DOR*, pages 121–130. Eurographics Association, 2014.
41. K. Li, K. Pang, Y. Song, T. M. Hospedales, H. Zhang, and Y. Hu. Fine-grained sketch-based image retrieval: The role of part-aware attributes. In *WACV 2016, Lake Placid, NY, USA, March 7-10, 2016*, pages 1–9, 2016.
42. Y. Li, T. M. Hospedales, Y. Song, and S. Gong. Free-hand sketch recognition by multi-kernel feature learning. *Computer Vision and Image Understanding*, 137:1–11, 2015.
43. Y. Li, Y. Song, T. M. Hospedales, and S. Gong. Free-hand sketch synthesis with deformable stroke models. *International Journal of Computer Vision*, 122(1):169–190, 2017.
44. T. Lu, C. Tai, F. Su, and S. Cai. A new recognition model for electronic architectural drawings. *Computer-Aided Design*, 37(10):1053–1069, 2005.
45. P. S. Maybeck. *Stochastic models, estimation, and control: Volume 1*, volume 141 of *Mathematics in Science and Engineering.* Academic Press, 1979.
46. M. Mohamad, M. Shafry, M. Rahim, N. Othman, and Z. Jupri. A comparative study on extraction and recognition method of cad data from cad drawings. In *ICIME 2009*, pages 709–713, 2009.
47. A. Nealen, O. Sorkine, M. Alexa, and D. Cohen-Or. A sketch-based interface for detail-preserving mesh editing. In *SIGGRAPH 2007, San Diego, California, USA, August 5-9, 2007, Courses*, page 42, 2007.

48. L. Olsen, F. F. Samavati, M. C. Sousa, and J. A. Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 33(1):85–103, 2009.
49. T. Y. Ouyang and R. Davis. Chemink: A natural real-time recognition system for chemical drawings. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, pages 267–276, New York, NY, USA, 2011. ACM.
50. C. D. Paoli and K. Singh. Secondskin: sketch-based construction of layered 3D models. *ACM Trans. Graph.*, 34(4):126:1–126:10, 2015.
51. F. Radenovic, G. Tolias, and O. Chum. Deep Shape Matching. In *The European Conference on Computer Vision (ECCV)*, September 2018.
52. Y. Sahillioglu and T. M. Sezgin. Sketch-based articulated 3D shape retrieval. *IEEE Computer Graphics and Applications*, 37(6):88–101, 2017.
53. P. Sangkloy, N. Burnell, C. Ham, and J. Hays. The Sketchy database: learning to retrieve badly drawn bunnies. *ACM Trans. Graph.*, 35(4):119:1–119:12, 2016.
54. R. G. Schneider and T. Tuytelaars. Sketch classification and classification-driven analysis using fisher vectors. *ACM Trans. Graph.*, 33(6):1–9, Nov. 2014.
55. O. Seddati, S. Dupont, and S. Mahmoudi. Deepsketch: Deep convolutional neural networks for sketch recognition and similarity search. In *CBMI 2015, Prague, Czech Republic, June 10-12, 2015*, pages 1–6, 2015.
56. O. Seddati, S. Dupont, and S. Mahmoudi. Deepsketch 3. *Multimedia Tools and Applications*, May 2017.
57. J. Sedmidubský and P. Zezula. Similarity Search in 3D Human Motion Data. In A. El-Saddik, A. D. Bimbo, Z. Zhang, A. G. Hauptmann, K. S. Candan, M. Bertini, L. Xie, and X. Wei, editors, *Proceedings of the 2019 on International Conference on Multimedia Retrieval, ICMR 2019, Ottawa, ON, Canada, June 10-13, 2019*, pages 5–6. ACM, 2019.
58. C. Shao, A. Bousseau, A. Sheffer, and K. Singh. Crossshade: shading concept sketches using cross-section curves. *ACM Trans. Graph.*, 31(4):45:1–45:11, 2012.
59. P. Shilane, P. Min, M. M. Kazhdan, and T. A. Funkhouser. The Princeton Shape Benchmark. In *(SMI 2004, 7-9 June 2004, Genova, Italy*, pages 167–178, 2004.
60. I. Sipiran, J. Lokoc, B. Bustos, and T. Skopal. Scalable 3D shape retrieval using local features and the signature quadratic form distance. *Vis. Comput.*, 33(12):1571–1585, 2017.
61. J. G. Snodgrass and M. Vanderwart. A standardized set of 260 pictures: norms for name agreement, image agreement, familiarity, and visual complexity. *Journal of Experimental Psychology: Human Learning and Memory*, 6(2):174–215, 1980.
62. P. Sousa and M. J. Fonseca. Geometric matching for clip-art drawing retrieval. *J. Vis. Comun. Image Represent.*, 20(2):71–83, Feb. 2009.
63. H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 945–953, 2015.
64. Z. Sun, C. Wang, L. Zhang, and L. Zhang. Query-adaptive shape topic mining for hand-drawn sketch recognition. In *ACM MM'12, Nara, Japan, October 29 - November 02, 2012*, pages 519–528, 2012.
65. I. E. Sutherland. Sketchpad: a man-machine graphical communication system. In *Proceedings of the SHARE Design Automation Workshop*, DAC '64, pages 329–346, New York, NY, USA, 1964. ACM.
66. J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3d shape retrieval methods. *Multimedia Tools Appl.*, 39(3):441–471, 2008.
67. A. Vedaldi and K. Lenc. MatConvNet: Convolutional Neural Networks for MATLAB. In *ACM MM '15, Brisbane, Australia, October 26 - 30, 2015*, pages 689–692, 2015.
68. R. C. Veltkamp and F. B. ter Haar. *SHREC 2007 3D Retrieval Contest*. Technical Report UU-CS-2007-015, Department of Information and Computing Sciences, Utrecht University, 2007.
69. F. Wang, L. Kang, and Y. Li. Sketch-based 3D shape retrieval using convolutional neural networks. In *CVPR 2015, Boston, MA, USA, June 7-12, 2015, pages = 1875–1883, year = 2015*,.
70. P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Transactions on Graphics (SIGGRAPH)*, 36(4), 2017.
71. G. Welch and G. Bishop. An introduction to the Kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.
72. J. Xie, G. Dai, F. Zhu, and Y. Fang. Learning barycentric representations of 3D shapes for sketch-based 3D shape retrieval. In *CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3615–3623, 2017.

73. Y. Ye, B. Li, and Y. Lu. 3D sketch-based 3D model retrieval with convolutional neural network. In *ICPR 2016, Cancún, Mexico, December 4-8, 2016, pages = 2936–2941, year = 2016.*
74. S. M. Yoon, M. Scherer, T. Schreck, and A. Kuijper. Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours. In *ACM Multimedia*, pages 193–200, 2010.
75. Q. Yu, F. Liu, Y. Song, T. Xiang, T. M. Hospedales, and C. C. Loy. Sketch me that shoe. In *CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 799–807, 2016.
76. Q. Yu, Y. Yang, F. Liu, Y. Song, T. Xiang, and T. M. Hospedales. Sketch-a-Net: a deep neural network that beats humans. *International Journal of Computer Vision*, in press, 2017.
77. Q. Yu, Y. Yang, Y. Song, T. Xiang, and T. M. Hospedales. Sketch-a-Net that beats humans. In *BMVC 2015, Swansea, UK, September 7-10, 2015*, pages 7.1–7.12, 2015.
78. R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. SKETCH: an interface for sketching 3D scenes. In *SIGGRAPH 2007, San Diego, California, USA, August 5-9, 2007, Courses*, page 19, 2007.
79. F. Zhu, J. Xie, and Y. Fang. 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.*, pages 3683–3689, 2016.