3D Sketching for Interactive Model Retrieval in Virtual Reality

Daniele Giunchi
University College London
London
d.giunchi@ucl.ac.uk

Stuart James
University College London
London
stuart.james@ucl.ac.uk

Anthony Steed
University College London
London
a.steed@cs.ucl.ac.uk

Figure 1: Our approach searches a dataset of chair 3D models (top left) using free-hand drawn sketches within a virtual reality. Our setup consists of Oculus RIFT, Oculus Touch and a laptop (bottom left). Sketches can be made either on top of a template chair or a prior search result (right).

ABSTRACT

We describe a novel method for searching 3D model collections using free-form sketches within a virtual environment as queries. As opposed to traditional sketch retrieval, our queries are drawn directly onto an example model. Using immersive virtual reality the user can express their query through a sketch that demonstrates the desired structure, color and texture. Unlike previous sketch-based retrieval methods, users remain immersed within the environment without relying on textual queries or 2D projections which can disconnect the user from the environment. We perform a test using queries over several descriptors, evaluating the precision in order to select the most accurate one. We show how a convolutional neural network (CNN) can create multi-view representations of colored 3D sketches. Using such a descriptor representation, our system is able to rapidly retrieve models and in this way, we provide the user with an interactive method of navigating large object datasets. Through a user study we demonstrate that by using our VR 3D model retrieval system, users can perform search more quickly and intuitively than with a naive linear browsing method. Using our system users can rapidly populate a virtual environment with specific models from a very large database, and thus the technique has the potential to be broadly applicable in immersive editing systems.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; HCI design and evaluation methods; • **Computing methodologies** → **Object recognition**; **Ranking**;

KEYWORDS

Sketch, Virtual Reality, CNN, HCI

ACM Reference Format:
Daniele Giunchi, Stuart James, and Anthony Steed. 2018. 3D Sketching for Interactive Model Retrieval in Virtual Reality. In *Expressive '18: The Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering, August 17–19, 2018, Victoria, BC, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3229147.3229166

Expressive '18, August 17–19, 2018, Victoria, BC, Canada

ACM ISBN 978-1-4503-5892-7/18/08…$15.00
https://doi.org/10.1145/3229147.3229166

1 INTRODUCTION

In recent years, with the rapid growth of interest in 3D modeling, repositories of 3D objects have ballooned in size. While many models may be labeled with a few keywords and/or fields that describe their appearance and structure, these are insufficient to convey the complexity of certain designs. Furthermore, in many existing databases, these keywords and fields are incomplete. Thus query-by-example methods have become a very active area of research. In query-by-example systems (see Section 2), the user typically sketches elements of the object or scene they wish to retrieve. A search system then retrieves matching elements from a database.

In our system, the user is immersed in a virtual reality display. We provide a base example of the class of object to act as a reference for the user. The user can then make free-form colored sketches on and around this base model. A neural net system can analyze this sketch and retrieve a set of matching models from a database. The user can then iterate by making further correctional sketches (e.g. adding new pieces to the model) until they find an object that closely matches their intended model. This leverages the strengths of traditional approaches while embracing new interaction modalities uniquely available within a 3D virtual environment.

The main challenge in sketch-based retrieval is that annotations in the form of sketches are an approximation of the real object and may suffer from being a subjective representation and over-simplifications. These abstract representations present challenges to description methods and therefore require unique consideration. For image retrieval, methods focus on enhancing lines through gradients, GF-HOG [Hu and Collomosse 2013] and Tensor Structure [Eitz et al. 2011] or using or multidimensional indexing structure such as NB-Tree [Fonseca et al. 2004], with more recent approaches based on convolutional neural networks (CNNs) [Bui et al. 2017; Yu et al. 2016]. In contrast for 3D, the use of sketching for retrieval has been limited to 2D projections for matching [Eitz et al. 2012]. To match 3D models, it is typical to normalize models to have the same orientation, so that a set of consistent images at set orientation can be rendered to compare the sketch to (see Section 2.2.2). We adopt this view-based method as it allows an interactive experience where users get responses with little delay.

So far, sketching within a virtual environment as a retrieval method has received little attention. There are various tools to allow the user to sketch (e.g. Tiltbrush, or Quill), but these focus on the sketch itself as the end result. Other systems allow free-form manipulation of objects by simple affine manipulation through drag points [Santos et al. 2008]. In contrast, we instead are interested in how a user can utilize sketch as a method of retrieval. We therefore performed a user study to compare sketch-based retrieval to a naive linear browsing to demonstrate that sketching is an effective and usable method of exploring model databases.

The contributions of our work are four-fold. First, we present a novel approach to searching model collections based on annotations on an example model. This example model represents the current best match within the dataset and sketching on this model is used to retrieve a better match. A novel aspect of our method is that we allow users to make sketches directly on top of existing models. The users can express color, textures and the shape of the desired object. Second, we evaluate different descriptors through a preliminary study in order to select the most accurate one, discovering that CNN achieves the highest precision. Third, we perform a user study to demonstrate the advantages of a sketch-based retrieval system in contrast to naive search. We show that users understand the purpose and practical use of a sketch-based retrieval system and that they are easily able to retrieve target objects from a large database. Finally, our system is the first of its type to work online in an immersive virtual environment. This model retrieval technique can be broadly applied in editing scenarios, and it enables the editor to remain immersed within the virtual environment during their editing session.

In the remainder of the paper, Section 2 reviews virtual environment modeling, sketching and representation methods. Section 3 explains the intricacies of our virtual environment model retrieval system and the novel use of interactive machine-learning-based searches to enable an iterative sketch and query refinement process. Section 4 presents a user study to demonstrate its effectiveness in terms of both accuracy and user experience rating. We then discuss the comparison between descriptors and the search technique in Section 5 and describe future work and limitations in Section 6. We then conclude in Section 7.

2 RELATED WORKS

Sketching represents a natural way for people to convey information. Eitz [Eitz M. 2012] give an overview of how people sketch objects and how sketches are recognized by humans and computers. The fundamental supposition is that sketches approximate the real world object. On the other hand, since the average user is not an artist, the subjective representation of an object can be iconic and include possible simplification of the objects. We explore the implications of this for both retrieval and interaction with respect to both 2D (Image) and 3D domains.

2.1 Sketch-based Image Retrieval

Identifying and associating a sketch with a specific object in an image represents a hard challenge. However, it is an attractive strategy because the use of sketch interaction is an opportunity to broaden the user base to those who are unfamiliar with complex interactive editing systems. Various methods for retrieving images from sketches have been developed. These systems are referred to as sketch-based image retrieval (SBIR) systems [Birari D.R. 2015]. SBIR techniques can be classified into two classes: blob-based techniques that focus the attention on features such as shape, color or texture, and contour-based techniques that describe the image using curves and lines. Techniques belonging to the blob-based SBIR class try to describe image through descriptors such as QBIC [Ashley et al. 1995] which use separately color, texture and shape or [Sousa and Fonseca 2010] which uses topology models. Contour-based techniques include elastic matching [Bimbo et al. 1994] and grid and interest points such as edge points [Chalechale et al. 2005].

In recent years researchers have applied machine learning algorithms to SBIR. SketchANet [Yang and Hospedales 2015] is a simple neural network based on Alexnet that performs sketch recognition. Qi [Qi et al. 2016] introduce a siamese CNN which aims to measure the compatibility between image edge-map and sketch used as CNN inputs. Bui [Bui et al. 2016] did a review of different

triplet CNN architectures for evaluating the similarity between pictures and sketches, focusing on the capacity to generalize between object classes. Triplet architectures (Wang [Wang et al. 2014], Sangkloy [Sangkloy et al. 2016]) have attracted increasing attention for the relationship of the three branches when processing the loss function: firstly the anchor branch (modeling the reference object), secondly a branch which models positive examples and thirdly a branch that deals with negative examples.

A strategy to improve the performance of image retrieval systems is to put the user 'in the loop' and take advantage of iterative refinement. This technique is called relevance feedback in information retrieval and was introduced in Content-Based Retrieval by Sciascio [Sciascio et al. 1999]. Several applications based on interactive sketch systems have been created. For example, Shadow Draw of Haldankar [Lee et al. 2011], iCanDraw [Collomosse et al. 2008], Sketch-to-Collage [Ruiz et al. 2007] and CALI system from Fonseca [Jota et al. 2006].

2.2 3D Sketch-based Retrieval and Interaction

Finding features that represent 3D objects is a unique challenge in the retrieval domain. Since one of the most important cues in object recognition is 3D geometric shape, sketching in 3D could represent a problem due to the abstract nature of the sketch. In addition, before sketch interpretation, a simplification process of the stroke can be taken for avoiding noisy samples [Fonseca et al. 2012] since both the tracking device and user generate noise during sketch acquisition.

In recent years, to depict a 3D model, researchers have proposed two type of descriptors : model-based and view-based.

2.2.1 Model-based descriptors. Measuring similarities between 3D models is a hard problem. Object models can differ in shape, color and orientation in 3D space, making the definition of a similarity measure challenging. Different categories of descriptors were created to overcome this challenge: geometric moment, surface distribution and volumetric descriptors. Geometric moment [Bronstein M. A. 2009] is a class of topology invariant similarity methods based on vector coefficient extracted by a shape decomposition under specific basis. Surface distribution [Osada et al. 2002] tries to measure the global properties through a shape distribution achieved by sampling a shape function and in this way reduces a shape comparison to a simpler distribution comparison. Volumetric descriptors [Rustamov 2010] combine shape distributions with barycentroid potential for achieving a more robust pose and topology invariant similarity. Despite the extensive research on descriptors that allows extracting shape characteristics, only with the advent of deep learning architectures such as Restricted Boltzmann Machines (RBM), Deep Belief Networks (DBN) and Deep Boltzmann Machines (DBM), and in particular CNN [Lecun et al. 1998] have achieved a relevant improvement of outcomes in object recognition. Wu [Wu et al. 2015] recently proposed a method to represent a 3D object through the distribution of binary variables in a volumetric grid, and use of Convolutional Deep Belief Networks to extract features and recognize them.

2.2.2 View-based descriptors. View-based descriptors use 2D projections of the objects from different points of view. Since a large amount of data that can be collected in this way, these methods outperform model-based descriptor approaches. Ansary [Ansary T.F. 2007] introduce a model-index technique for 3D objects that make uses of 2D views. It uses a probabilistic Bayesian method for 3D model retrieval. Alternatively, Su [Su et al. 2015a] present a framework using view-based descriptors, creating 12 views for each object that feed a first CNN for feature extraction, and after a pooling stage, the results are passed to another CNN for achieving a compact shape descriptor. Similarly, Leng [Leng et al. 2016] proposed a 3DCNN that manages multiple views and considers possible interactions between them. In a pre-process stage a sorting algorithm, which takes in consideration the angles and positions, prepares three different sets of viewpoints and the network is fed with them at the same time. This is a different approach from the classic one which uses only one view at a time and it confers stability during the training stage.

Li [Li et al. 2017] elaborates a technique that combine two components: an adaptive view clustering algorithm that selects representative views of the 3D model, and a sketch-based approach that compensates the difference between iconic representation of the object given by sketch depiction and the detailed appearance of the same object.

Our method uses view-based descriptors, rather than model-based descriptors because they have demonstrated more practical utility on similar problems.

Recent studies combine data achieved by sketch and an additional input to increase accuracy, or infer information from the relationship with other object in the scene. Funkhauser [Funkhouser et al. 2003] proposed a combination of sketch and text query to identify 3D objects. They showed that the combination of the two methods results in a better accuracy of the results. Shin and Igarashi [Shin and Igarashi 2007] with Magic Canvas provided the user with a system for 3D scene construction using sketches, based on sketch-object similarity. In addition, the system determines the position and orientation of the object according to the sketch. Xu [Xu et al. 2013] with Sketch2Scene proposed a novel framework for scene modeling through sketch drawing that suggests also the placement of the objects via functional and spatial relationship between objects.

Critically these methods have generally used 2D sketches. Our system allows the user to sketch in 3D.

2.2.3 3D sketching. 3D sketch-based model retrieval has gained significant attention in recent years. Li [Li et al. 2016a] made a comparative evaluation of different 3D sketch-based model retrieval algorithms showing that CNN in combination with edge or point sketches achieved the best accuracy. Ye [Ye et al. 2016] described CNN-SBR, a CNN architecture based on SketchANet [Yang and Hospedales 2015] and trained with TU Berlin dataset [Eitz M. 2012]. Using data augmentation to prevent overfitting, they showed a considerable improvement in comparison to non-learning based and other learning-based algorithm.

Considering alternative uses of sketch interaction within a 3D context, Wang [Wang et al. 2015] present a minimalist approach in terms of view-based descriptors. They generate only two views for the entire dataset and train a Siamese CNN with the views and the sketches. Nishida [Nishida et al. 2016] proposed a novel method to

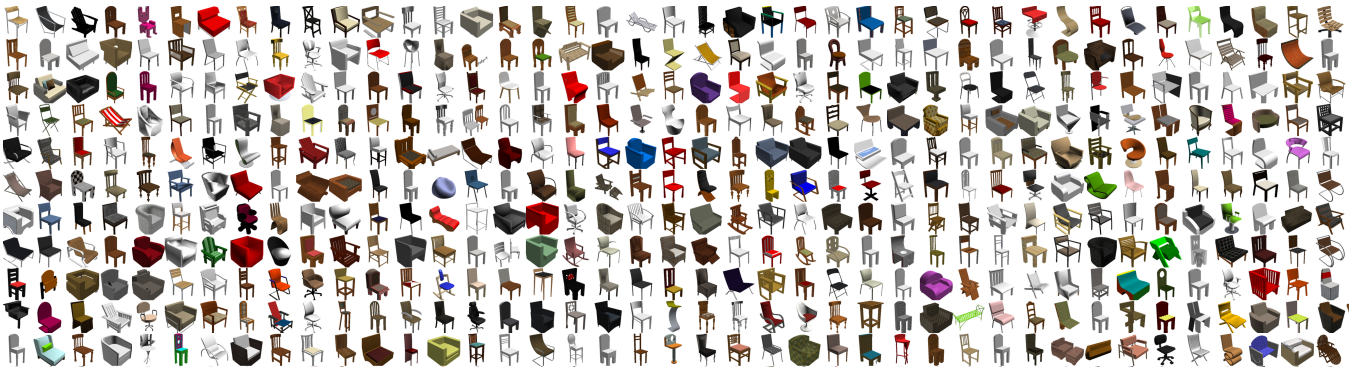

Figure 2: A collage of the *chairs* class models from ShapeNet dataset [Wu et al. 2015]. The collage shows a subset of the total set of 3370 chairs in order to illustrate the diversity of this class of object.

design buildings from sketches of different parts of them. The user sketches few strokes of the current object and through a pre-trained CNN for that specific object type, the system is able to procedurally retrieve the correct grammar snippet and select the most similar one. The final step of the process is to combine all the snippets in a unique grammar of the building just created.

2.2.4 Immersive Sketching. Immersive sketch-based modeling has gained a lot of attention over the years. A very early example is Clark's 3D modeling system for a head-mounted display [Clark 1976]. The system of Butterworth [Butterworth et al. 1992] supported several geometric modeling features, including freehand operations. More recently many immersive 3D modeling systems have exploited freehand sketching such as BLUI [Brody and Hartman 1999], CavePainting [Keefe et al. 2001], Drawing on Air [Keefe et al. 2007], FreeDrawer [Wesche and Seidel 2001], Holosketch [Deering 1996] and Surface Drawing [Schkolne et al. 2001]. Very recently applications for consumer virtual reality systems such as Tiltbrush from Google and Quill from Facebook have raised awareness of sketching for content development. The most similar work to ours is the system Air Sketching for Object Retrieval [Beatriz S. 2015]. This combines 3D sketch and a search engine based on the spherical harmonic descriptor. Our system uses a different type of lightweight sketching over basic models and a view-based descriptor. Another similar system that is that of Li [Li et al. 2016b], a content retrieval system that can benefit from sketch-based interaction using a Microsoft Kinect. Possibly because the sketches are relatively crude, they focus on distinguishing between classes of object. We focus on precise sketching to distinguish between similar objects in a large class of objects. Also, we enable sketching over existing models, rather than sketching from scratch.

3 3D SKETCH-BASED RETRIEVAL DESIGN

Sketch-based retrieval has had great success within 2D image retrieval, yet is still cumbersome when extended to 3D. We propose that by utilizing recent advances in virtual reality and by providing a guided experience, a user will more easily be able to retrieve relevant items from a collection of objects. We explore the proposed methodology on ShapeNet [Wu et al. 2015]. ShapeNet is an extensive 3D model collection that includes a large set of model classes. We demonstrate our method to the subset of this collection that contains chairs, although our method is applicable to many classes of object. The chair subset is large and exhibits a large amount of variation that is particularly suitable for our method (see fig. 2). We first outline our proposed Sketch-based Retrieval pipeline (subsec. 3.1) then go on to define a study to demonstrate the benefits of using such a method compared to naive linear searching (sec. 4).

3.1 3D Sketch-based Retrieval

Searching for a model in a large collection using 2D sketches can be tedious and requires an extended period of time. It also requires a particular set of skills, such as understanding perspective and occlusion. By using virtual reality this experience can be improved because ambiguity between views is greatly reduced and the user no longer has to imagine the projections from 2D to 3D.

3.1.1 3D Sketch Descriptor. Sketching within a 3D environment has been explored through stroke analysis [Choi et al. 2005; Fiorentino et al. 2003; Rausch et al. 2010], but little work has been performed to describe the set of strokes in a compact representation, i.e. descriptor, such as in SBIR [Eitz M. 2012] or SBVR [James and Collomosse 2014]. Therefore we explore state-of-the-art model descriptions approaches. We apply four traditional Bag of Words approaches: SIFT [Lowe 2004],Histogram of Gradients(HoG) [Dalal and Triggs 2005],Gradient Field Histogram of Gradients (GF-HoG) [Hu and Collomosse 2013] and ColorSIFT [Abdel-Hakim and Farag 2006]. It is worth noting that only ColorSIFT descriptor incorporates a description of color. In addition, we apply a multi-view CNN architecture to describe the content of the model.

Each proposed method generates a unique descriptor of the chair. To generate a single vector description of a model the chair is projected into 12 distinct views as shown in fig. 3. Each view is then described by an independent model. This exhibits an early fusion approach which we describe for both deep and shallow descriptor generation methods.

In the multi-view CNN architecture [Su et al. 2015b] the standard VGG-M network of [Chatfield et al. 2014] is applied. This model consists of five convolutional layers and three fully connected layers (depicted in fig. 4). As in [Su et al. 2015b] the model is trained on ImageNet then fine-tuned on the 2D images of the 3D Shapes

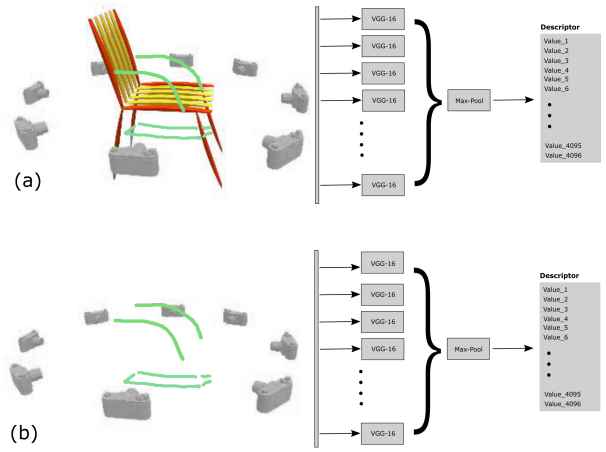

Figure 3: (a) CNN can be triggered with snapshots with both sketch and chair model. (b): CNN can be triggered with snapshots with only sketch present.

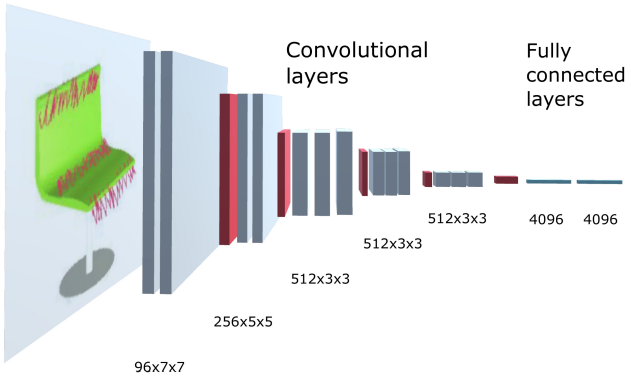

Figure 4: Each view is processed by the shown VGG-M architecture model [Chatfield et al. 2014]. As demonstrated in fig. 3 the network is split after convolutional layers the final Multi-View descriptor is the output of the network a vector of 4096 scalars.

dataset. For each view of the model the convolutional layers of the VGG-M are applied where the resulting descriptors are aggregated by element-wise max pooling. The result of the max-pooling is then fed through a second part of the VGG-M network (i. e. fc layers) where the second fully connected layer (fc_7) is used as the descriptor for the view (V) resulting in $V \in \mathbb{R}^{4096}$. The VGG-M network is trained once and shared amongst views.

For SIFT, ColorSIFT, HOG and GFHOG, we used the bag of words (BoW) mechanism to generate a descriptor from all the views. The BoW implementation is defined with $K = 1024$ clusters that represent the visual words and where the frequency histogram across views is accumulated to generate a singular descriptor, $V \in \mathbb{R}^{1024}$ for these methods.

We perform a preliminary evaluation of the descriptors for retrieval of models (See Section 5.1) and identify the approach of Su [Su et al. 2015b] to significantly outperform the alternative methods, henceforth we discuss the approach in regards to this descriptor. An index is generated from the dataset by repeating the aforementioned process over the dataset generating a matrix $M = D^{n \times 4096}$ where n is the number of items in the collection.

3.1.2 Online Queries. At query time, the multiple views are generated from the user's sketches and, optionally the current 3D model that is the best match (see below), and a forward pass through the network returns the descriptor. For simplicity and ease of comparison of results, we leave M to be linearly searched at query time. Improved efficiency could be achieved by using KD-Trees or other popular index structures. Therefore, we define the distance d as squared Euclidean:

$$d_i = |M_i - Q|^2 \quad (1)$$

where Q is the query descriptor. After comparing the descriptor with the descriptor collection the system replies with the K-nearest models that fit the input sent. In our experiments we use $K = 40$. The retrieved models are ordered by their respective r_i distance.

We provide the user two ways to perform the query: sketch-only query or both sketch and model query. This is achieved by enabling or disabling the visualization of the model (see fig. 3). After the system proposes results, if the user's target model is not present the user can edit the sketch or conversely can replace the current model with a new one that better matches represents the desired target. Such a possibility helps the user to minimize the time sketching: they can focus on sketching the missing or different parts relative to the current best match model. This facilitates a step by step refinement to navigate through the visual feature space of the collection, commonly achieving the target model only after a few iterations. In the current implementation (see Section 4.3) the response time after each user search request is 2 seconds. This is sufficiently quick to allow a tight interactive loop between sketching and querying. Users are free to either make a complex sketch that will likely match on the first attempt, or add features to the model in several iterations, thus facilitating a 'walk' through the model collection towards the desired target.

4 USER STUDY: COMPARISON OF SKETCHED QUERIES OVER LINEAR SEARCH IN VR

4.1 Task Overview

We designed an experiment to compare two methods: the proposed sketch-based method, and a naive scrolling panel method. For each session of the test, we first showed the participant the twelve views of a target chair as generated for the descriptor. We then asked the participant to retrieve the chair from the database, using one of the two methods. For both methods the participant started in a scene of a furnished room where a chair is positioned on the floor to the user's left-hand side. We perform this initialization step to minimize the required hand travel distance avoiding any mobility bias. We tracked the success rate, the time to complete the task and a subjective evaluation of the user experience through a questionnaire.

The scroll method consists of finding the target chair from the entire collection of 3370 chairs using a panel that shows 10 chairs at once and which can be scrolled forward and backward very quickly. After the user starts the session, the chairs are randomly shuffled to prevent recall of the order from memory. The user then simply searches for the target chair (see fig. 5). When the user is confident that they have found the chair, they select it from the panel in order to replace the current chair in the room. When the participant clicks the end label, the time required to complete the task is taken and the session is finished. For the sketch method, the user makes colored sketches on top of the initial model (see fig. 6) and then uses the hand-held device to trigger the search method. The system proposes 40 chairs as outcome, shown 10 at a time in a scroll panel which is navigable in the same fashion as the scroll method. The participants would iteratively sketch, triggering the retrieval system or selecting models from the 40 suggestions then continue to refine the sketch. The search refinement process continues until the target chair is located and the user can terminate the session.

4.2 Procedure

All participants are asked to complete an introduction form with basic information related to their previous user experience in 3D software and VR applications. Each user performs two sessions of tests. Where each session is comprised of two sub-sessions. In each sub-session, the user performs three search tasks for different chairs models with one method, and then the same three searches with the other method.

Participants were instructed before each of the four sub-sessions with an application demo in which it will be shown the modality they had to use. In addition, they could select to practice for a short time to familiarize with the interaction. Each of the search tasks was started by asking the user to look at a particular target chair with the instruction find it using the selected method. For the sketch-based method, we instructed the user to use the style they prefer, that could be based predominantly on making a single sketch or on system interrogation with multiple iterations of model replacement. Each user was allowed to perform the task seated or standing. An upper time limit was defined as 4 minutes in order to keep each user session slightly less than an hour. In the event the user was unable to locate the target chair within the time limit or the wrong chair was selected, the search was considered a failure

Figure 5: The scroll method provides a simple scrolling panel for navigating the database of all the chairs.

Figure 6: An example of a user's sketch within the sketch interface. The query is comprised of colored 3D strokes drawn on top of a chair model.

and the time cropped to 4 minutes. The two sessions differed in starting method used and from the different set of target chairs; thus the order of the methods is counter-balanced, and each subject uses both methods twice. We split the users into two groups: the first group started with the naive scroll method in the first session, while the second started with the sketch method. In total each participant performed 12 searches. In this way, we were able to analyze the task completion time considering the contribution related to the different techniques, to the chair types and to the learning curve effect of VR interaction. We choose six different chairs with specific structure and colors. In particular, both striped and curvy shapes are present in the sets with a variety of different colors as shown in fig.8. After completing all four sub-sessions (12 search tasks), participants filled in a final form with their rating on user experience and level of confidence for both scroll method and sketch method. The scale of the rating was expressed in the form of a scalar from 1 to 5.

4.3 Implementation

The participants used an Oculus Consumer Version 1 (CV1) head-mounted display (HMD) as well as Oculus Touch controllers. The experiment was performed on a PC laptop with a Processor Intel(R) Core(TM) i7-6700 CPU, NVIDIA GeForce GTX 980M graphics card and 64 GB of RAM.

The virtual reality software was created within Unity. The scene consisted of a furnished room, with the addition of a chair when the system was initialized. During the scroll-based method, the user can select models from a floating panel in which can scroll pages of models and display 10 models at time. The panel is attached to the left hand and the selection is performed using right-hand controller. Ten models were chosen so as to provide a panel that was small enough not to occlude large parts of the environments, but large enough that features in the chair were easily legible inside the HMD.

The sketching mechanism is implemented through the generation of colored lines. Lines are implemented as narrow strips that expose their wider section to the current camera. Therefore, each virtual camera, used for multi-view generation, renders the larger section of the strip independent from the sketch path. The user can the color using a palette connected to the left-hand GUI. The user

can draw 3D lines in the virtual environment on top of the current model and can submit to the system using the controller's triggers. We provided also a simple UNDO function that acted on the sketch stack. We did not provide additional tools in order to stimulate users to play essentially with pure sketch interaction. The back-end is a separate service thread in which a CNN Model is preloaded and ready to respond to user queries. We integrated the VGG-M Matlab implementation of Su [Su et al. 2015b]. This is triggered to produce a unique visual descriptor given the snapshots generated by VR application as described in sec.3.1.
To maintain a reasonable computation time, the first convolutional layers (see figure 4) use stride 2, while the latter layers are used as normal. On average the CNN process takes approximately 0.5 seconds to produce a descriptor after receiving input.

5 RESULTS

In this section we describe the results achieved by preliminary study that compares different descriptors followed by the outcomes of the user test.

5.1 Comparison between 3D Sketch Descriptors

We perform a preliminary study using a set of six queries over the different descriptors and evaluate their retrieval precision with regards to a set of criteria for the returned model. Following the approach of Collomosse [Collomosse et al. 2008; James and Collomosse 2014] we evaluate the precision in terms of of this different facets of the retrieval, therefore for each correctly returned facet of the model the score is incremented. These correspond to: 1) Structure – majority of the parts arms back, seat, legs; 2) Style – curvy, straight, with many lines; 3) Color – dominant color matches query.

This study aims to identify the descriptor that achieves the best precision for the search task. The most accurate method is then used in the user test. In addition we prepared two sets of queries, the first are pure sketch queries, while the second are a combination of the sketch and the model. We considered the top 10 retrieved chairs proposed by each method, ranked from position 1 to position 10. Each rule can assign only one point if matched and focuses on a specific feature of the model. We formalized the rules as follows:

(1) we consider four components of the chair: back, seat, arms and legs. Where if more than 75% are similar to the target, the result is considered correct;
(2) if the proposed chair shows a dominant style (curvy, stripes, convex, etc.) similar to the target chair;
(3) if the proposed chair shows a dominant color similar to the target chair.

With each result receiving points for the facets a final score in the range of $[0, 3]$ is calculated, which is then normalized across facets and queries for a result in $[0, 1]$. The precision is calculated from the scores for each result, using the equations:

$$P_r = \frac{\sum_{i=1}^{r} S_i}{r}, \tag{2}$$

where P_r is the average precision for the rank r, S_i is the score for rank i assigned by our metric. We compare SIFT, ColorSIFT, HOG, GF-HOG and VGG-M, calculating the average precision for each chair of the top 10 retrieved models. VGG-M method outperforms all the other methods using sketch and model queries (as shown in fig. 7a) and also using only sketches (as shown in supplemental material).
We calculate Mean Average Precision (MAP) for each descriptor. For the sketch and model queries VGG-M's MAP achieves 0.28, followed by GF-HOG with 0.18. This pattern is similarly reflected within the Sketch only queries, with VGG-M's MAP highest at 0.22, followed by SIFT with 0.13. Therefore, we perform the user test using descriptors generated by VGG-M.

5.2 User Study

Our user study consists of 30 participants recruited from the ANON department and general public. We split the participant into two equal size groups (15 users per group). The first group of participants started with scroll method, while the second group started with the sketch method. Twenty of the participants were male (10 female) while the average age of the participants is 26 years. Each of the participants in the study performed 6 scroll and 6 sketch tasks, giving a total of 360 search tasks across all participants. The tasks splits are demonstrated in fig. 8 with regard to group and session (see supplementary material for user final queries), i. e. twelve trials per user, with 15 participants doing the first task with scroll, 15 doing the first task with sketch.

The number of successful task completions for the scroll method was 119 out of 180 (66%) and for the sketch method 171 out of 180 (95%). In fig. 13 we show the total number of completions for each method for each task, in the order that participants completed the tasks in their respective groups. This graph shows the impact of individual tasks being found easier or harder by the participants. As there does not appear to be a trend over the sequence of tasks for the sketch method, it demonstrates minimal learning required and the intuitive nature of the method.

The task completion performance for the sketch method can be affected by the complexity of the target model, where difficult models are challenging to depict. The participant may have improved their depiction ability or efficiency with the system, but this can not be conclusively drawn from these results. While the significant factor for the linear search is the position within the dataset. It also can be seen a much larger variation in completions per task for scroll than sketch. For task three, only 3 participants completed the search with the scroll method. This in comparison with sketch, the minimum number of completions was 12.

We show the time to complete all tasks in fig. 10 for each of the methods. We can see that the distributions are very different, with a cross-over point at around 60 seconds. This can be explained by the fact that the completion time for the scroll method is largely determined by the page number that the result appears on; while for the sketching method there is an additional interaction overhead for completing the query sketch and the search time.

By comparing the average time to complete all six models for the sketch or scroll methods in a paired-comparison per user – i. e. each pair comprised the average time to complete all six sketch tasks and the average time to complete all six scroll tasks. Additionally where any failures to complete were clamped to 240s (4 minutes). The median time to complete the sketch tasks was 99.8s, and the

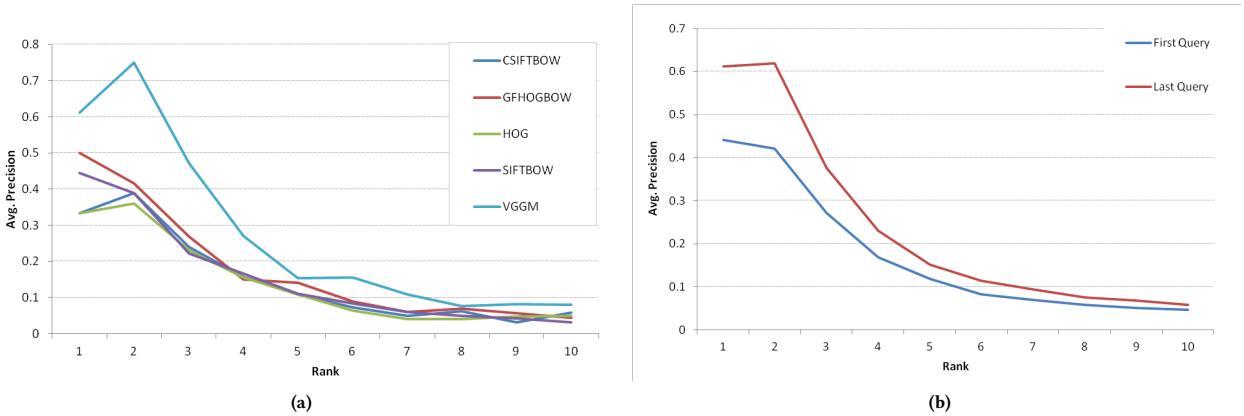

Figure 7: (a) Average precision calculated across ranked results from preliminary study. (b) Comparison of the first and last query average precision from user study.

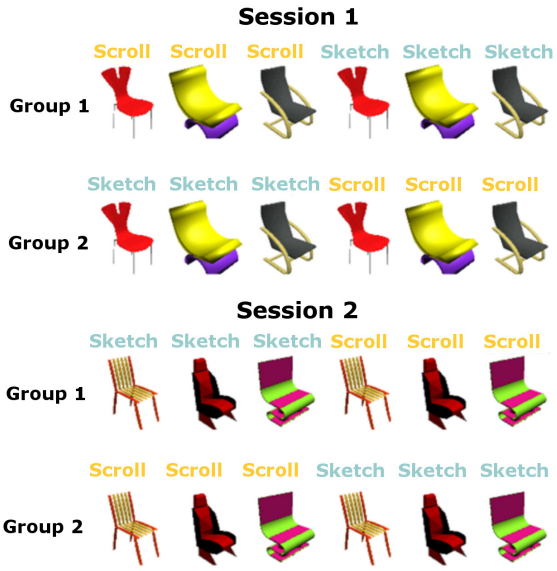

Figure 8: Two groups of 15 users are created. The first group performed the scroll method as the first method for the first set of chairs, then with the sketch method for the first set of chairs, then swapped the methods over for the second set of chairs. The second group did the opposite order of methods on the same order of sets of chairs.

median time to complete the scroll tasks was 156.5s. Because of the distribution of times, and the clamping on failure, we used the exact sign test to compare the differences. This showed that the difference in medians was significant, with p less than 0.0005. We asked participants to report a feedback on user experience. In fig. 11 we show an average rating of sketch and scroll methods for all users. We can see quite clearly that users strongly prefer the sketch method, with only two users rating the scroll method as favorite one, four showing no preference and the remainder (24) preferring the sketch method.

Qualitative examples are shown in fig. 9a and fig. 9b, showing the types of sketch created by the participants. We discuss further the difference between the types of sketch in Section 6.

Finally, we reflect on the development of the precision of results across the session for users in the case more than one query was performed. Our purpose is to quantify the improvement between the first and the last query, without considering the cases in which the user found the target chair after the first interaction, and therefore considering the refinement of the results over time. We evaluate using the same mechanism as in the comparison of descriptors (sec. 5.1) but solely for the selected descriptor VGG-M, in fig. 7b (b) we can observe for each rank an improvement of the scores achieved by the last query compared with the first. To quantify this improvement we calculated the MAP for the first queries that achieves 0.17, while the MAP for the last queries is 0.24, showing an improvement during time.

6 DISCUSSION

In this section we discuss the outcomes achieved by the study on different descriptors and the results obtained by the user study.

6.1 Comparison between 3D Sketch Descriptors

Our preliminary study compares the precision achieved by different descriptors in order to decide the most accurate method for the user test. We defined the metric rules in such a way that it avoids assigning additional points if the target chair is present in the results. Despite this, VGG-M clearly achieves highest precision scores for all the top 10 ranks. Consequently, this result shows that VGG-M descriptor is the most accurate in retrieving different facets (color, style and shape).

6.2 User study

Our purpose is to explore 3D sketch interaction for object retrieval in order to understand its validity and possible developments. Therefore, we designed an experiment to identify different

Figure 9: (a) Examples of users that successfully triggered the system using a combination of sketches and model. The left column contains the target chairs, while the other columns contain a subset of the snapshots used by the system. (b) Examples of users that successfully triggered the system using only sketches. The left column contains the target chairs, while the other columns contain a subset of the snapshots used by the system.

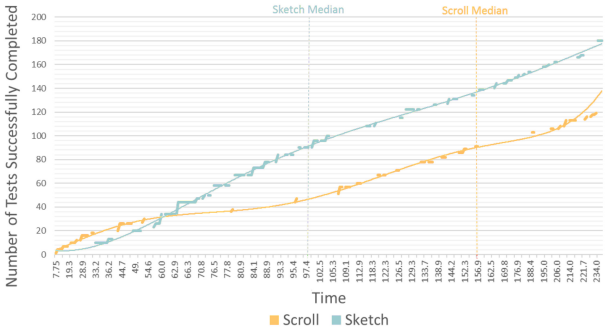

Figure 10: Cumulative time distribution for the scroll and sketch method. If the target chair was not found within the time limit (240 seconds) the time is limited to this.

user approaches between our method and a simple linear search. In addition, we avoided to include complex functionalities during sketch phase to study the effectiveness of pure sketch interaction.

Our experiment shows that it is possible, through an iterative process of sketching and model selection, to perform an effective search for a model in a large database while immersed in a virtual environment. Further the accuracy and the completion time are significantly improve on naive scroll method and the participants also prefer the sketch based approach.

While the scroll method represents a baseline with a clear and linear work-flow to the user, the sketch method allows different strategies. In general two different techniques emerged from the experiment: sketch only as shown by examples in fig. 9b and sketch with a model as shown in fig. 9a. The first and more intuitive approach is to make a single sketch and detail it step by step until most features of the chairs are resolved without replacing the model. The user can interrogate the system to have a feedback but essentially will continue to sketch. The downside is that the user can waste time on detailing a sketch and, in addition, can depict features that are not relevant. Determining whether features are relevant or not is not a trivial task for two reasons. The first one is that different users will over-rate the saliency of the feature (e. g. it may be an uncommon feature but it has not been captured by the descriptor). The second one is the possibility that the specific feature is common to many objects of the database. Both cases can lead to an unsatisfactory answer from the system as it proposes a chair set without that feature or conversely many chairs containing it.

The second approach is to only model differences to the current object: that is the user queries the system and then only adds features that are different in the target object. The sketch is usually started again after each query. The advantage of this method is that the quick response from the system (~2 seconds) enables fast iterative refinement. Every time the system receives a different combination of sketch and model it will retrieve a different set of chairs. This method requires more experience from the user, but after few iterations we observed several participants starting to adopt it. In addition we demonstrate, through the comparison between first and last query outcomes, that user improves the precision as the search progresses with time, increasing the similarity of the facet of the retrieved models with the facet of the target.

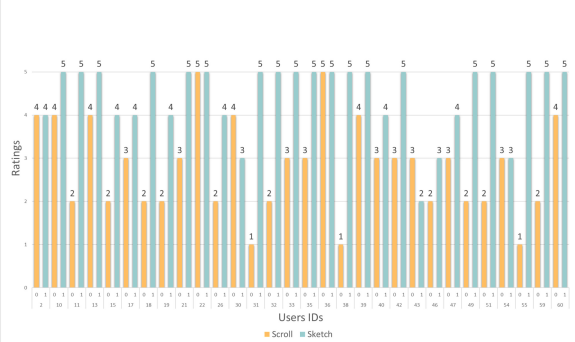

Figure 11: Each participants ratings for both the scroll and sketch method on a scale of one to five.

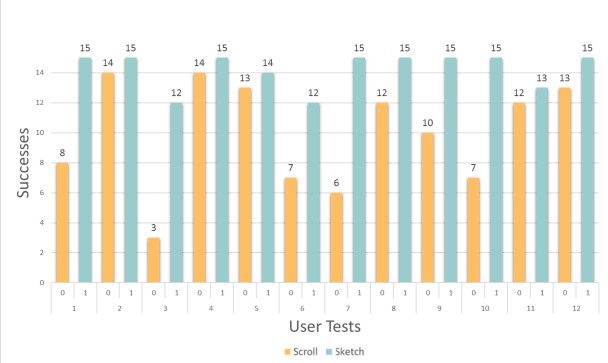

Figure 13: The number of successful searches made for scroll and sketch for each user.

7 LIMITATIONS AND FUTURE WORK

Despite the benefits of the using sketch and the positive feedback from the user study, several aspects could be investigated to improve the search accuracy or experience. These are outlined below:

Multiple Object Categories. In addition to working with chairs we performed an additional experiment with the table collection within the ShapeNet database. We verified the same behavior of the system using the proposed approach. As the approach has no fine-tune training for the chair object category it is plausible that results can further be extrapolated over the larger collection, with an initial object category selection at initialization.

Gestures, Brushes and UI. We opt to avoid additional interaction learning that can occur from gestures or brushes. But, these are useful tools allowing a user to shortcut through tasks. It is easy to imagine using gesture recognition for object type identification (table etc.) avoiding NLP or text selection. Alternatively familiar tools from photo editing e. g. brushes to aid in depicting large region color or fill-bucket tool to specify a regions texture (an element not easily depicted).

Baseline method selection. We used a linear search as the baseline as it avoids taking the user out of the immersive reality and requires minimal training that could introduce bias. An alternative method of searching collections is based on text filtering or faceted search.

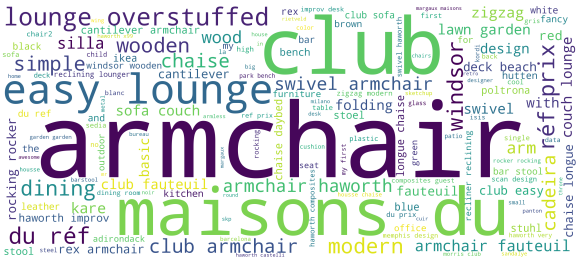

Figure 12: Word Cloud of the 150 most fequent words in tags and descriptions (with stopwords removed) generated over the ShapeNet Chair subset.

Although an attractive approach, model collections are rarely annotated with adjectives to convey the visual appearance of the object. This is indeed the case with ShapeNet, even with the considerable amount of human annotation in terms of both keyword tags and brief descriptions these fail to describe the diversity of the model. This can be seen through fig. 12 where we analyze both tags and words in the description showing the top 150 keywords as a tag cloud scaled dependent on their frequency. It can be seen words describe their specific object names or location that would be difficult or unlikely to be conveyed by the user (Meta-data for target models is provided in the supplementary material). It would be expected that a keyword search may only marginally improve search time.

8 CONCLUSION

The benefits of the virtual reality in the field of scene modeling have been investigated for several years. Previous research has focused on free-form modeling rather than developing a way to retrieve models from a large database. Current strategies for navigating an existing dataset use queries on tags or simply show to the user the entire set of models. In addition, large collections can suffer from a lack of meta-information which hampers model search and thus excludes part of the dataset from query results. We proposed a novel interaction paradigm that helps users to select a target item using an iterative sketch-based mechanism. We improve this interaction with the possibility of combining sketches and a background model together to form a query to search for a target model. We run a study to determine the most accurate descriptor. An experiment collected information about the time taken to complete the task and user experience rating. We compared our method with a naive scrolling selection method. The sketch-based method was clearly preferred by users and led to a significant reduction in search time. We thus believe that sketch-based queries are a very promising complement to existing immersive sketching systems.

ACKNOWLEDGMENTS

This project received funding from the European Union's Horizon 2020 research and innovation programme, under the Marie Skłodowska-Curie grant agreements No 642841 (DISTRO).

REFERENCES

Alaa E. Abdel-Hakim and Aly A. Farag. 2006. CSIFT: A SIFT Descriptor with Color Invariant Characteristics. In *2006 IEEE Computer Society Conference on Computer*

Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA. 1978–1983. https://doi.org/10.1109/CVPR.2006.95

Vandeborre J.P. Ansary T.F., Daoudi M. 2007. A Bayesian 3-D Search Engine Using Adaptive Views Clustering. *IEEE Transaction on Multimedia* 9, 1 (2007), 78 âĂŞ–88.

Jonathan Ashley, Myron Flickner, James L. Hafner, Denis Lee, Wayne Niblack, and Dragutin Petkovic. 1995. The Query By Image Content (QBIC) System. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, May 22-25, 1995*. 475. https://doi.org/10.1145/223784.223888

Martins N. V. Beatriz S. 2015. *Air-Sketching for Object Retrieval*. Technical Report. Instituto Superior T ÌĄecnico, Lisboa, Portugao.

Alberto Del Bimbo, Pietro Pala, and Simone Santini. 1994. Visual Image Retrieval by Elastic Deformation of Object Sketches. In *Proceedings IEEE Symposium on Visual Languages, St. Louis, Missouri, USA, October 4-7, 1994*. 216–223. https://doi.org/10.1109/VL.1994.363615

Shinde J.V. Birari D.R. 2015. Survey on Sketch Based Image Retrieval. *International Journal of Advanced Research in Computer and Communication Engineering* 4, 12 (2015).

Arthur W. Brody and Chris Hartman. 1999. BLUI: a body language user interface for 3D gestural drawing. In *Human Vision and Electronic Imaging (SPIE Proceedings)*, Vol. 3644. SPIE, 356–363.

Kimmel R. Bronstein M. A., Bronstein M. M. 2009. Topology-Invariant Similarity of Nonrigid Shapes. *Int. J. Comput. Vis.* 81 (2009), 281 âĂŞ– 301. https://doi.org/10.1007/s11263-008-0172-2

T. Bui, L. Ribeiro, M. Ponti, and J. Collomosse. 2017. Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network. *Computer Vision and Image Understanding* (2017). https://doi.org/10.1016/j.cviu.2017.06.007

Tu Bui, Leonardo Ribeiro, Moacir Ponti, and John P. Collomosse. 2016. Generalisation and Sharing in Triplet Convnets for Sketch based Visual Search. *CoRR* abs/1611.05301 (2016).

Jeff Butterworth, Andrew Davidson, Stephen Hench, and Marc. T. Olano. 1992. 3DM: A Three Dimensional Modeler Using a Head-mounted Display. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics (I3D '92)*. ACM, New York, NY, USA, 135–138. https://doi.org/10.1145/147156.147182

Abdolah Chalechale, Golshah Naghdy, and Alfred Mertins. 2005. Sketch-based image matching Using Angular partitioning. *IEEE Trans. Systems, Man, and Cybernetics, Part A* 35, 1 (2005), 28–41. https://doi.org/10.1109/TSMCA.2004.838464

K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *British Machine Vision Conference*. arXiv:cs/1405.3531

Han-wool Choi, Hee-joon Kim, Jeong-in Lee, and Young-Ho Chai. 2005. Free Hand Stroke Based Virtual Sketching, Deformation and Sculpting of NURBS Surface. In *Proceedings of the 2005 International Conference on Augmented Tele-existence (ICAT '05)*. ACM, New York, NY, USA, 3–9. https://doi.org/10.1145/1152399.1152401

James H. Clark. 1976. Designing Surfaces in 3-D. *Commun. ACM* 19, 8 (Aug. 1976), 454–460. https://doi.org/10.1145/360303.360329

John P. Collomosse, Graham McNeill, and Leon Adam Watts. 2008. Free-hand sketch grouping for video retrieval. In *ICPR*. IEEE Computer Society, 1–4.

Navneet Dalal and Bill Triggs. 2005. Histograms of Oriented Gradients for Human Detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. 886–893. https://doi.org/10.1109/CVPR.2005.177

Michael F. Deering. 1996. The HoloSketch VR Sketching System. *Commun. ACM* 39, 5 (May 1996), 54–61. https://doi.org/10.1145/229459.229466

Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. 2011. Sketch-Based Image Retrieval: Benchmark and Bag-of-Features Descriptors. *IEEE Transactions on Visualization and Computer Graphics* 17, 11 (2011), 1624–1636.

Mathias Eitz, Ronald Richter, Tamy Boubekeur, Kristian Hildebrand, and Marc Alexa. 2012. Sketch-based shape retrieval. *ACM Trans. Graph.* 31, 4 (2012), 31:1–31:10. https://doi.org/10.1145/2185520.2185527

Alexa M. Eitz M., Hays J. 2012. How Do Humans Sketch Objects?. In *ACM Trans. Graphics, 31(4)*.

Michele Fiorentino, Giuseppe Monno, Pietro Renzulli, and Antonio Uva. 2003. 3D sketch stroke segmentation and fitting in virtual reality. (01 2003).

Manuel J. Fonseca, Alfredo Ferreira, and Joaquim A. Jorge. 2004. Towards 3D Modeling Using Sketches and Retrieval. In *Proceedings of the First Eurographics Conference on Sketch-Based Interfaces and Modeling (SBM'04)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 127–136. https://doi.org/10.2312/SBM/SBM04/127-136

M. J. Fonseca, S. James, and J. Collomosse. 2012. Skeletons from sketches of dancing poses. In *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 247–248. https://doi.org/10.1109/VLHCC.2012.6344537

Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. 2003. A Search Engine for 3D Models. *ACM Trans. Graph.* 22, 1 (Jan. 2003), 83–105. https://doi.org/10.1145/588272.588279

Rui Hu and John Collomosse. 2013. A Performance Evaluation of Gradient Field HOG Descriptor for Sketch Based Image Retrieval. *Comput. Vis. Image Underst.* 117, 7 (July 2013), 790–806. https://doi.org/10.1016/j.cviu.2013.02.005

Stuart James and John Collomosse. 2014. Interactive Video Asset Retrieval Using Sketched Queries. In *Proceedings of the 11th European Conference on Visual Media Production (CVMP '14)*. ACM, New York, NY, USA, Article 11, 8 pages. https://doi.org/10.1145/2668904.2668940

Ricardo Jota, Alfredo Ferreira, Mariana Cerejo, José Santos, Manuel J. Fonseca, and Joaquim A. Jorge. 2006. Recognizing Hand Gestures with CALI. In *SIACG*. Eurographics Association, 187–193.

Daniel F. Keefe, Daniel Acevedo Feliz, Tomer Moscovich, David H. Laidlaw, and Joseph J. LaViola, Jr. 2001. CavePainting: A Fully Immersive 3D Artistic Medium and Interactive Experience. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics (I3D '01)*. ACM, New York, NY, USA, 85–93. https://doi.org/10.1145/364338.364370

Daniel F. Keefe, Robert C. Zeleznik, and David H. Laidlaw. 2007. Drawing on Air: Input Techniques for Controlled 3D Line Illustration. 13, 5 (2007), 1067–1081.

Yann Lecun, LÃĺon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*. 2278–2324.

Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen. 2011. ShadowDraw: real-time user guidance for freehand drawing. *ACM Trans. Graph.* 30, 4 (2011), 27:1–27:10.

Biao Leng, Yu Liu, Kai Yu, Xiangyang Zhang, and Zhang Xiong. 2016. 3D Object Understanding with 3D Convolutional Neural Networks. *Inf. Sci.* 366, C (Oct. 2016), 188–201. https://doi.org/10.1016/j.ins.2015.08.007

Bo Li, Yijuan Lu, Fuqing Duan, Shuilong Dong, Yachun Fan, Lu Qian, Hamid Laga, Haisheng Li, Yuxiang Li, Peng Liu, Maks Ovsjanikov, Hedi Tabia, Yuxiang Ye, Huanpu Yin, and Ziyu Xue. 2016a. 3D Sketch-based 3D Shape Retrieval. In *Proceedings of the Eurographics 2016 Workshop on 3D Object Retrieval (3DOR '16)*. Eurographics Association, Goslar Germany, Germany, 47–54. https://doi.org/10.2312/3dor.20161087

Bo Li, Yijuan Lu, Fuqing Duan, Shuilong Dong, Yachun Fan, Lu Qian, Hamid Laga, Haisheng Li, Yuxiang Li, Peng Liu, Maks Ovsjanikov, Hedi Tabia, Yuxiang Ye, Huanpu Yin, and Ziyu Xue. 2016b. 3D Sketch-Based 3D Shape Retrieval. In *Eurographics Workshop on 3D Object Retrieval*, A. Ferreira, A. Giachetti, and D. Giorgi (Eds.). The Eurographics Association. https://doi.org/10.2312/3dor.20161087

Bo Li, Yijuan Lu, Henry Johan, and Ribel Fares. 2017. Sketch-based 3D Model Retrieval Utilizing Adaptive View Clustering and Semantic Information. *Multimedia Tools Appl.* 76, 24 (Dec. 2017), 26603–26631. https://doi.org/10.1007/s11042-016-4187-3

David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 2 (01 Nov 2004), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Gen Nishida, Ignacio Garcia-Dorado, Daniel G. Aliaga, Bedrich Benes, and Adrien Bousseau. 2016. Interactive Sketching of Urban Procedural Models. *ACM Trans. Graph.* 35, 4, Article 130 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925951

Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. 2002. Shape Distributions. *ACM Transactions on Graphics* 21 (2002), 807–832.

Yonggang Qi, Yi-Zhe Song, Honggang Zhang, and Jun Liu. 2016. Sketch-based image retrieval via Siamese convolutional neural network. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*. 2460–2464. https://doi.org/10.1109/ICIP.2016.7532801

Dominik Rausch, Ingo Assenmacher, and Torsten W. Kuhlen. 2010. 3D Sketch Recognition for Interaction in Virtual Environments. In *Proceedings of the Seventh Workshop on Virtual Reality Interactions and Physical Simulations, VRIPHYS 2010, Copenhagen, Denmark, 2010*. 115–124. https://doi.org/10.2312/PE/vriphys/vriphys10/115-124

David Gavilan Ruiz, Suguru Saito, and Masayuki Nakajima. 2007. Sketch-to-collage. In *SIGGRAPH Posters*. ACM, 35.

Raif M. Rustamov. 2010. Robust Volumetric Shape Descriptor. In *3DOR*. Eurographics Association, 1–5.

Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. 2016. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Trans. Graph.* 35, 4 (2016), 119:1–119:12.

Tiago Santos, Alfredo Ferreira, Filipe Dias, and Manuel J. Fonseca. 2008. Using Sketches and Retrieval to Create LEGO Models. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling (SBM'08)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 89–96. https://doi.org/10.2312/SBM/SBM08/089-096

Steven Schkolne, Michael Pruett, and Peter Schröder. 2001. Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, New York, NY, USA, 261–268. https://doi.org/10.1145/365024.365114

Eugenio Di Sciascio, G. Mingolla, and Marina Mongiello. 1999. Content-Based Image Retrieval over the Web Using Query by Sketch and Relevance Feedback. In *VISUAL (Lecture Notes in Computer Science)*, Vol. 1614. Springer, 123–130.

HyoJong Shin and Takeo Igarashi. 2007. Magic Canvas: Interactive Design of a 3-D Scene Prototype from Freehand Sketches. In *Proceedings of Graphics Interface 2007 (GI '07)*. ACM, New York, NY, USA, 63–70. https://doi.org/10.1145/1268517.1268530

Pedro Manuel Antunes Sousa and Manuel J. Fonseca. 2010. Sketch-based retrieval of drawings using spatial proximity. *J. Vis. Lang. Comput.* 21, 2 (2010), 69–80. https://doi.org/10.1016/j.jvlc.2009.12.001

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015a. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV '15)*. IEEE Computer Society, Washington, DC, USA, 945–953. https://doi.org/10.1109/ICCV.2015.114

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. 2015b. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV*.

Fang Wang, Le Kang, and Yi Li. 2015. Sketch-based 3D Shape Retrieval using Convolutional Neural Networks. *CoRR* abs/1504.03504 (2015). arXiv:1504.03504 http://arxiv.org/abs/1504.03504

Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning Fine-grained Image Similarity with Deep Ranking. *CoRR* abs/1404.4661 (2014).

Gerold Wesche and Hans-Peter Seidel. 2001. FreeDrawer: A Free-form Sketching System on the Responsive Workbench. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '01)*. ACM, New York, NY, USA, 167–174. https://doi.org/10.1145/505008.505041

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*. IEEE Computer Society, 1912–1920.

Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu. 2013. Sketch2Scene: Sketch-based Co-retrieval and Co-placement of 3D Models. *ACM Transactions on Graphics* 32, 4 (2013), 123:1–123:12.

Yongxin Yang and Timothy M. Hospedales. 2015. Deep Neural Networks for Sketch Recognition. *CoRR* abs/1501.07873 (2015). http://arxiv.org/abs/1501.07873

Yuxiang Ye, Bo Li, and Yijuan Lu. 2016. 3D sketch-based 3D model retrieval with convolutional neural network. , 2936-2941 pages.

Q. Yu, F. Liu, Y. Z. Song, T. Xiang, T. M. Hospedales, and C. C. Loy. 2016. Sketch Me That Shoe. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 799–807. https://doi.org/10.1109/CVPR.2016.93