SketchMaker: Sketch Extraction and Reuse for Interactive Scene Sketch Composition

FANG LIU, BNRist, MOE-Key Laboratory of Pervasive Computing, Department of Computer Science and Technology, Tsinghua University, China
XIAOMING DENG, State Key Laboratory of Computer Science and Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, China
JIANCHENG SONG, State Key Laboratory of Computer Science and Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences; University of Chinese Academy of Sciences, China
YU-KUN LAI, School of Computer Science and Informatics, Cardiff University, UK
YONG-JIN LIU*, BNRist, MOE-Key Laboratory of Pervasive Computing, Department of Computer Science and Technology, Tsinghua University, China
HAO WANG, Alibaba Group, China
CUIXIA MA*, State Key Laboratory of Computer Science and Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, University of Chinese Academy of Sciences, China
SHENGFENG QIN, School of Design, Northumbria University, UK
HONGAN WANG, State Key Laboratory of Computer Science and Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, University of Chinese Academy of Sciences, China

*Corresponding Authors.

This work was supported by the National Natural Science Foundation of China under Grant 61872346 and 61725204, Tsinghua University Initiative Scientific Research Program, Natural Science Foundation of Beijing under grant 4212029, China Postdoctoral Science Foundation under grant 2021M701889, Newton Prize 2019 China Award under grant NP2PB/100047 and Alibaba Group through Alibaba Innovative Research Program. Yong-Jin Liu and Cuixia Ma are corresponding authors.
Authors' addresses: Fang Liu, BNRist, MOE-Key Laboratory of Pervasive Computing, Department of Computer Science and Technology, Tsinghua University, Beijing, China, liufang515@tsinghua.edu.cn; Xiaoming Deng, State Key Laboratory of Computer Science and Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, Beijing, China, xiaoming@iscas.ac.cn; Jiancheng Song, State Key Laboratory of Computer Science and Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences; University of Chinese Academy of Sciences , Beijing, China, songjiancheng19@mails.ucas.ac.cn; Yu-Kun Lai, School of Computer Science and Informatics, Cardiff University, Cardiff, UK, Yukun.Lai@cs.cardiff.ac.uk; Yong-Jin Liu, BNRist, MOE-Key Laboratory of Pervasive Computing, Department of Computer Science and Technology, Tsinghua University, Beijing, China, liuyongjin@tsinghua.edu.cn; Hao Wang, Alibaba Group, Beijing, China, cashenry@126.com; Cuixia Ma, State Key Laboratory of Computer Science and Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China, cuixia@iscas.ac.cn; Shengfeng Qin, School of Design, Northumbria University, Newcastle upon Tyne, UK, sheng-feng.qin@northumbria.ac.uk; Hongan Wang, State Key Laboratory of Computer Science and Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China, hongan@iscas.ac.cn.

2160-6455/2022/6-ART $15.00
https://doi.org/10.1145/3543956

Sketching is an intuitive and simple way to depict sciences with various object form and appearance characteristics. In the past few years, widely available touchscreen devices have increasingly made sketch-based human-AI co-creation applications popular. One key issue of sketch-oriented interaction is to prepare input sketches efficiently by non-professionals because it is usually difficult and time-consuming to draw an ideal sketch with appropriate outlines and rich details, especially for novice users with no sketching skills, thus sketching brings great obstacles for sketch applications in daily life. On the other hand, hand-drawn sketches are scarce and hard to collect. Given the fact that there are several large-scale sketch datasets providing sketch data resources, but they usually have a limited number of objects and categories in sketch, and do not support users to collect new sketch materials according to their personal preferences. In addition, few sketch-related applications support the reuse of existing sketch elements. Thus, knowing how to extract sketches from existing drawings and effectively re-use them in interactive scene sketch composition will provide an elegant way for sketch-based image retrieval (SBIR) applications, which are widely used in various touch screen devices. In this study, we first conduct a study on current SBIR to better understand the main requirements and challenges in sketch-oriented applications. Then we develop the SketchMaker as an interactive sketch extraction and composition system to help users generate scene sketches via reusing object sketches in existing scene sketches with minimal manual intervention. Moreover, we demonstrate how SBIR improves from composited scene sketches to verify the performance of our interactive sketch processing system. We also include a sketch-based video localization task as an alternative application of our sketch composition scheme. Our pilot study shows that our system is effective and efficient, and provides a way to promote practical applications of sketches.

CCS Concepts: • **Human-centered computing** → **Interactive systems and tools**; • **Information systems** → *Information retrieval*; • **Computing methodologies** → *Scene understanding*; Artificial intelligence.

Additional Key Words and Phrases: Sketch Composition, Scene Sketch Extraction, Sketch Reuse

1 INTRODUCTION

Sketching is a simple and effective way to describe objects' visual appearance and spatial layout, and can express user intents in an intuitive way. With the rapid developments of touch devices, sketch has been widely used in computer-assisted designing [32, 45] and human-AI co-creation tasks [6, 31, 44] in the last decade. Recently, sketch-based image retrieval, which aims at finding natural images with sketches, becomes one of the most popular sketch-based applications, especially with the proliferation of touch-screen devices in daily life. As claimed in [51],"a sketch speaks for a 'hundred' words.", sketch is intuitive and easy to use, and it is able to contain rich information that is hard to describe by using text. When users want to retrieve fine-grained images, sketches have great advantage over text. For example, people may type in "shoe" to conduct a search when he/ she wants to find *a shoe*, but they may prefer to draw a sketch to retrieve *a particular shoe* [51] with such as expected forms. However, drawing an sketch with rich details is important but usually hard and time-consuming in SBIR for most non-professionals, and the obstacles become even greater when drawing a scene containing multiple objects. Although generating sketches via reusing existing object sketch datasets is a highly potential approach to address the sketch drawing issue, these datasets are scarcely reused according to users' personal preferences. Thus, effective reuse of existing sketches and quick composition of input sketches for SBIR and other sketch-related applications brings a new perspective, which is an important problem for investigation.

We know that large amounts of object sketches exist as parts of scene sketches, e.g. in storyboards, animations, comics or illustrations in story books, where overlapping and occlusion often occur. While object-level sketches are easier to be organized and manipulated than scene sketches, these sketches in the form of scenes and sketches of multiple overlapping objects make direct reuse difficult. For example, Fig. 1 shows a storyboard of the famous animation *Gertie the Dinosaur*, which was cast in 1914 and has been seen as the first animation movie in the world [36]. The objects in the storyboard scenes (the dinosaur, the elephant, the tree, the stones, etc.) tend to be partially

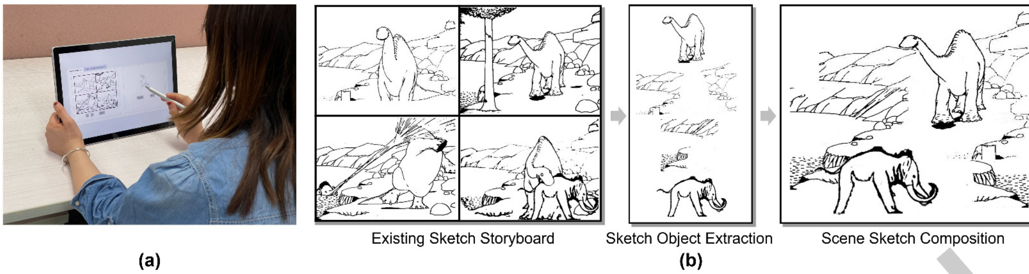

Fig. 1. (a) An user using SketchMaker with a pad. (b) A storyboard of the animation *Gertie the Dinosaur*. In this paper, our vision is to extract object sketches from existing sketch scenes and use them to help users with no sketching skills conduct sketch composition. Our system can both enhance the reuse of existing sketch data and offer users a convenient way to composite scene sketches at the same time.

occluded and incomplete. In most situations, only sketches of complete single objects may be reused for recreation in practice, while it is often impossible to obtain the same complete single objects drawn by the original authors. Our vision in this paper is to provide users with a convenient way to extract existing object sketches. Furthermore, these extracted object sketches can be reused for new sketch composition. In other words, our method can improve the reuse of sketch resource for downstream sketch-based orientations.

As pointed by “iCanDraw?” [10] that most people do not believe that they are able to draw well and do not know how to adjust their work during the drawing process, preparing sketch data is not an easy thing for novice users. Several systems have been long proposed to assist user sketching [12, 24, 48]. In the deep learning era, though there are several large-scale sketch datasets, e.g., QuickDraw [18] and TU-Berlin [11], providing some source materials for sketch-based human-AI cooperation applications, the numbers of object categories in these datasets are limited (e.g. TU-Berlin [11] contains 250 classes of objects, of which there are about 80 instances in each class, the Sketchy database contains 125 classes of objects). Moreover, the sketches in QuickDraw is collected by an online game where the players are asked to draw objects belonging to a particular object class in less than 20 seconds. The sketches in QuickDraw are relatively rough, lacking details and tending to be iconic or canonical views. These pioneering sketching assistant tools and databases only focus on object-level sketches, where there is only one object instance in each sketch. Consequently, most sketch-oriented applications only study on single object processing, as the input sketch of multiple objects (scene sketch) is even harder to obtain. Exceptions are the newly proposed SketchyScene [53] and SketchyCOCO [14]. SketchyScene consists of about 7,000 scene sketch templates, but the sketches are all cartoon images. Moreover, the sketches in SketchyCOCO are relatively rough and the number of object categories is limited to 14. Although a few sketch datasets have been established, to the best of our knowledge, few attempts have been made to interactively extract existing sketch data (at either object-level or scene-level) and create new scenes for further reuse. In this sense, these datasets have not been well designed for practical applications. Compared with the existing datasets, one of the goals of our system is to allow more flexible creation of new sketch data. We offer users a way to automatically obtain completed object-level sketches and save them to users’ personal favorites. Our SketchMaker helps users obtain and reuse their interested object sketches from exisiting scene sketches in an interactive manner. Furthermore, our SketchMaker can be extended to support sharing the users’ personal favorites to obtain more sketch data, so that more information could be exchanged between different users.

Sketch-based research has received unprecedented attention in the last decade, among which one of the most popular topics is to use sketches as a query input to conduct image retrieval. Compared to traditional query forms of image retrieval, especially text, sketch-based image retrieval (SBIR) mainly has two advantages: i) sketches can describe fine-grained object characteristics intuitively, while text-based queries mainly focus on retrieving objects of the same category, and are often hard to describe the visual appearance (form, size, orientation, etc.). ii) scene sketches are more convenient in depicting the layout and object details (directions, relative relationships, sizes, poses) of images than text. Although sketching is an convenient way to search for images, one key issue of applying SBIR is how to prepare effective input sketches with necessary details in an efficient manner. Creating a scene sketch with appropriate object outlines and details is challenging for most non-professional users, which hinders the uptake of SBIR in a wide range of applications in real life.

In this paper, we present SketchMaker, an interactive scene sketch composition system that helps users with no drawing skills generate scene sketches in an efficient way by reusing existing sketch data. We focus on the sketch extraction and scene sketch composition problem. As a user browses a sketch storyboard or scene sketches, SketchMaker can help extract the sketch objects (and object-associated sketches) in the scene through sketch completion operation. Our SketchMaker allows the user to select object sketches from the database and fine-tune them via editing operations including shifting, zooming in / out, rotating, etc. when creating new scene sketches. Moreover, with sketch extraction and reuse, our SketchMaker helps users to composite scene sketches easily, and can promote applications of SBIR and sketch-based video localization.

Extracting and reusing hand-drawn sketch objects from existing scene sketches is challenging, due to the inherent characteristics of sketches: large internal and external variations exist between sketches, there is no exact alignment between sketch and the corresponding object outline, and it is also ambiguous to draw an appropriate sketch. For example, the same object can be drawn in diverse styles by different users, or even by the same user in different drawing environments. Sketches can also be drawn with a large range of abstractness with visual details varying according to the users' preferences. Moreover, since people tend to use sketches to express inaccurate concepts and ideas, even visually different sketches may convey similar information. Furthermore, object-level sketches are easier to be organized and manipulated than scene sketches. Benefiting from the progress of deep learning, we can leverage deep learning techniques to improve the capability of processing sketches, leading to an expansion of sketch applications. We cope with sketch completion procedure to help users to extract object sketches for future reuse, such as storyboard composition. As for sketch-based image retrieval task, a completed sketch containing more information can facilitate a more precise retrieval than a corrupted one, as pointed out in SketchGAN [33]. Overall, by adopting a generative adversarial network (GAN) to complete corrupted object sketches and reuse them via a series of scene sketch editing operations to produce scene sketches, our SketchMaker enables users to interactively select their interested sketch elements from existing scenes when reading a storyboard or watching sketch animations, and further use them for typical sketch-oriented applications, such as image retrieval and video localization.

Overall, in this paper, we propose an interactive scene sketch processing and composition system, SketchMaker, to extract and reuse existing sketch data to help non-professional users create new scene sketches, providing a convenient way to generate input sketch for sketch-oriented applications such as SBIR. The main contributions of this work can be summarized as follows:

(1) We design SketchMaker, an interactive scene sketch creation system for non-professional sketching users to composite scene sketches in an efficient way. Our system can help with the restoration and reuse of sketch data from existing sketch data.

(2) SketchMaker systematically integrates scene sketch creation into sketch-based image retrieval application, which can facilitate other sketch-oriented research and applications.
(3) We conducted two user studies and both the qualitative and quantitative results show that our SketchMaker can help users easily create scene sketches and make full use of existing sketch data. Moreover, the results show that our interactive SBIR has better performance than the baseline system.

2 RELATED WORK

The literature review is divided into three subsections: we first review recent sketch co-creation with AI works discussing how sketch-oriented applications are used in the AI era. Then, we focus on pioneering sketch assistance tools. At last, we narrow down to the scene sketch processing research since the creation and acquisition of scene sketch input data are even harder than object sketch elements.

Human-Computer Co-Creation with Sketches. Sketch-based user interfaces in this AI era span from traditional sketch-based applications (e.g. sketch recognition [3, 20], sketch-based video retrieval [39], illustrations creation [28], collaborative fabrication [4]), to newly proposed tasks (e.g. sketch-text generation [22], sketch animation [44], sketch-based user interface retrieval [23], etc.)

More related to our work are *DuetDraw* [37] and IMOTION [46]. *DuetDraw* presents an interface where AI and humans can draw pictures interactively to conduct further sketch colorization, showing great user experience and promising application prospects in human-AI co-creation. IMOTION is a video retrieval system that suggests a mid-sketch to users during their sketching process and then uses the mid-sketch to query videos. Although *DuetDraw* and IMOTION also integrate computer-aided sketch composition and follow-up sketch applications, they both aim at sketch drawing recommendations about creating object sketches, while our SketchMaker aims to help people composite more complex scene sketches that can enable more sketch-based user interactions.

Sketching Assistance tools. Although extensive algorithms have been proposed to improve the efficiency and effectiveness of sketch processing, most applications require users to personally draw a sketch as input. However, the sketch drawing skills vary dramatically from person to person, leading to the unstable performance of sketch algorithms. Various forms of systems have been designed for helping users to sketch [26, 27, 29, 35]. SketchTivity [19] is a stylus-based intelligent tutoring system that helps users improve design sketching by providing effective feedback to users. ILoveSketch [1] and EverybodyLovesSketch [2] are two 3D curve sketching systems which integrate many sketch-based interaction techniques to help users proficiently create 3D curve sketches.

In the aspect of computer-aided sketch creation, Lee et al. present ShadowDraw [30], a sketch drawing interface which guides users as they are drawing by offering suggestions for object contours. Drawing Apprentice [9] is a sketching system that applies creative sense-making cognitive framework to explore human collaboration with a co-creative AI agent. Similarly, iSketch&Fill [17] constructs a drawing system by making shape recommendation and image generation through deep learning-based methods. The objects in iSketch&Fill are limited to ten classes (moon, soccer, basketball, etc.) all with white background and simple circle shapes. EmoG [42] and Cali-Sketch [47] explore the problem of supporting the sketching of characters. EmoG uses an auto-encoder framework to generate emotional expressions for a given neutral face in storyboard, and Cali-Sketch integrates a stroke calibration and completion network for image generation from poorly-drawn face sketches. Both of them are limited to helping users produce a specific type of sketch face images, while our SketchMaker aims to handle multi-class objects and offer users with limited

Fig. 2. A snapshot of the user interface of Sketch Me *That* Shoe [51]. It has an object category selection button at the top, a sketching canvas on the right, and an image display panel at the bottom left. Users are allowed to draw a sketch in their minds to search for an image of shoe. The top 10 retrieved images are shown.

drawing skills a convenient way to conduct scene sketch compositions by reusing existing sketch data.

Scene Sketch Processing. Although recent advances in touch screen and deep learning techniques have made sketch processing and creation an easier task, most sketch datasets and research only study object-level sketch applications, i.e. only one instance in each sketch.

Scene250 [49] and Cross-Modal Places (CMPlaces) [5] are two scene sketch datasets. Scene250 is a scene sketch dataset consisting of 250 scene sketches in 10 categories, and it is designed for sketch scene recognition. CMPlaces is a cross-modal scene dataset covering five different modalities (natural images, line drawings, cartoons, text descriptions, and spatial text images) annotated with category labels. Recently, Zou *et al.* [52] present SketchyScene, a scene sketch database containing 7,000+ scene templates with instance-level segmentation annotations. All the scene sketches in SketchyScene are in art clip or cartoon style.

Regardless of whether there are object-level annotations in these scene sketch datasets, the object instance sketches are not reused for further applications. Our SkechMaker avoids requiring the user to draw a scene sketch for the follow-up applications, but instead offers an interactive pipeline to reuse sketch objects in existing scene sketches to conduct scene sketch composition.

3 NEEDFINDING STUDY

To understand the users' experience in current AI-collaboration sketching applications, we first conduct a survey with four non-professional sketching users with an average age of 25. Sketch Me *That* Shoe [51] is a recently famous mainstream sketch application, which retrieves color images using hand-drawn sketches as inputs. Fig. 2 shows a snapshot of the interface. Users draw a sketch shoe and the system returns a few pictures according to their similarities with the input. The sketches and images in Sketch Me *That* Shoe are all single objects with white backgrounds. We use Sketch Me *That* Shoe as a baseline system to investigate the users' experience when interacting with AI to query images using sketch.

We first showed the users an image as the query target for two seconds, and then we asked them to find out the image from the database via drawing a sketch with the impressions in their mind. After the test, we asked the users some questions:

(1) Compared with other conventional image retrieval methods, e.g. searching by category or keywords, what are the advantages and disadvantages of using sketch as the input form?
(2) Is the Sketch Me *That* Shoe system easy to use and what aspects of the system should be improved?

(3) For sketches as the query inputs, what is the most significant difficulty in use and the most urgent thing to improve?

Each user conducted 10 retrieval trials and the experiments lasted for about 1 hour (plus the interview time).

3.1 Results.

Our user study reveals some advantages and disadvantages of this typical SBIR system, as well as the challenges about using sketch as the form of query input.
Finding 1. The most general result is that users believe that compared to conventional retrieval methods (e.g. based on the category or keywords), **the sketch-based retrieval interaction method supports more accurate and flexible searching, and drawing can provide more details compared to plain text.** For example, one user said, "*I can more clearly describe the shape I have in mind. This can include aspect ratio of the shape and other aspects that are difficult to describe in words.*" The domain gap between real photos and text is usually larger than that between sketch. Users can describe their desired objects in a more intuitive way by drawing a rough sketch of the object outline with some fine details, which is hard to describe by a simple category or keywords. Another user said, "*Under certain circumstances, you even do not know what it is or how to describe it properly, but you do know what it looks like.*"
Finding 2. The biggest obstacle is users' drawing ability. All of the users think the most urgent need for the SBIR system is to provide a few ways of drawing support. One user said, "*Just like we have predictive input method for text, we also need some help when drawing sketches.*" Another user said, "*It is difficult to start drawing from nowhere, especially for those guys who rarely draw. Hence I believe showing some sketch examples would be better, or adding a search engine based on keywords to find a similar picture first, and then finishing the sketching based on it.*"
Finding 3. Although the system is relatively easy to use, some improvements are needed. For example, the retrieval results under some conditions are not accurate enough, and users hope the system can provide more functions, like searching history, color selection. One user said, "*One improvement direction would be allowing users to edit their drawings, for example the aspect ratios, so that they do not need to redraw the shoe every time, instead they could warp the shape to allow for faster search.*" Another user believes "*The system is easy to understand and use. However, if the user makes a sketch mistake, they'll have to reset the whole sketch and start again, which is a terrible waste of time.*" Sketching an accurate drawing can be difficult in one trial. **Editing capability of the drawing could allow users to refine their input.** Moreover,although we conduct needfinding study with a single object retrieve system, 3/4 of users expressed it would be better to support multi-object retrieval during the interview.

3.2 Design Requirements

By extracting the challenges and urgent needs for improvements, we identified the following design requirements in designing interfaces when users and AI collaborate in sketch applications:

(1) Firstly, suggesting and reusing existing sketches instead of asking users to personally draw a sketch to provide help to those who are poor at drawing.
(2) Secondly, offering more sketch editing tools besides a pen for interactive sketching aids. Particularly, we have designed a "color panel" sketch editing tool, which allows users to draw colored sketches to provide more information for potential applications.
(3) Lastly, allowing more flexibility and multiple object sketch composition as users are drawing, thus specifying more detailed information (e.g. object direction and background).

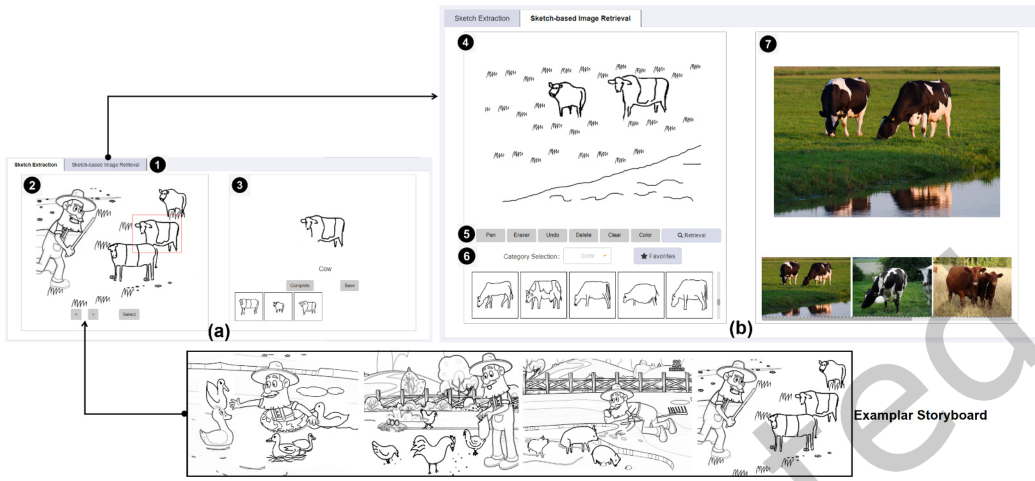

Fig. 3. The user interface of our SketchMaker contains two pages, (left) the Sketch Data Extraction Page and (right) the Scene Sketch Composition and Image Retrieval Page. The Sketch Data Extraction Page contains: ❶ the page switching panel, ❷ the scene sketch display panel, and ❸ the sketch object extraction tools panel. The Scene Sketch Composition and Image Retrieval Page contains: ❹ the scene sketch composition canvas, ❺ the scene sketch editing tools panel, ❻ the sketch object browsing and selection panel, ❼ the image retrieval result display panel.

4 SKETCHMAKER SYSTEM DESCRIPTION

4.1 User Interface

The user interface of our SketchMaker contains two pages (see Fig. 3), the Sketch Data Extraction Page and the Scene Sketch Composition and Image Retrieval Page. Users are able to conduct sketch extraction or scene sketch composition by switching between the two pages (Fig. 3 ❶). The interface is based on a website which can be adapted to devices such as mobile phone, laptop, digital whiteboard. SketchMaker supports inputs with fingers or pen on touch screens, as well as mouse on desktops.

In the Sketch Data Extraction Page: when users are reading a list of scene sketches in a storyboard, the scene sketch display panel displays the current scene sketch, and users are able to select their interested sketch object elements via drawing a rectangular box surrounding the object (Fig. 3 ❷). Sketch objects selected by users may exist occlusion or overlapping, then the sketch object extraction tools panel (Fig. 3 ❸) can automatically complete the sketch and save it to users' personal favorites.

In the Scene Sketch Composition and Image Retrieval Page: the scene sketch composition canvas (Fig. 3 ❹) supports user operations of compositing scene sketches, where dragging, moving, zooming in /out and rotating operations of sketch elements are offered. The scene sketch editing tools panel (Fig. 3 ❺) contains Pen, Eraser, Undo, Delete, Clear, Color and Retrieval (from left to right), supporting creation and editing of sketch elements. Users can query an image with the created scene sketch by clicking the Retrieval button. The sketch object browsing and selection panel lists sketch object elements selected or suggested by our system (Fig. 3 ❻). The image retrieval result display panel (Fig. 3 ❼) shows the retrieved images of our system, where the Top 1 result is shown at the top and other similar images are listed at the bottom.

The main functions of our SketchMaker system are as follows:

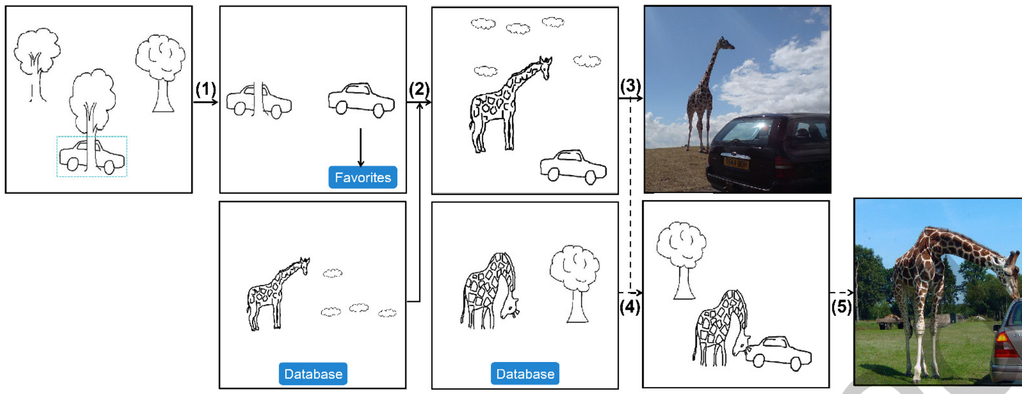

Fig. 4. An example drawing scenario of using our SketchMaker. When the user browses a scene sketch, our SketchMaker enables sketch object extraction via sketch completion (1) operations. Then the user can composite a sketch scene by selecting from personal favorites or database of sketch elements (2). The user can further use the created scene sketch as query input to retrieve an image (3). Moreover, users are able to continuously adjust the sketch composition via selecting different sketch elements from the database for further image retrieval ((4)-(5)).

(1) *Extracting objects from existing scene sketches:* This function enables our SketchMaker to help users obtain and save their interested object sketches from scene sketches in an interactive manner. This is done by user selecting an interested sketch element from an existing scene sketch and then our system completes the corrupted sketch object.
(2) *Scene sketch composition:* This function enables users to create new scene sketches in a fast and convenient way. Users are allowed to reuse existing sketch objects through dragging, moving, zooming in /out and rotating operations etc. Several drawing tools, e.g. pen, eraser, undo, are also offered for drawing new objects.
(3) *Image retrieval using scene sketch inputs:* This function allows users to query images using their composited scene sketches as inputs. Users can also continuously adjust the scene sketch to get a more appropriate image consistent with that in their mind.

Fig. 4 briefly illustrates a drawing scenario of using our SketchMaker system. When the algorithm occasionally works poorly, our system also enables users to manually adjust the sketch extraction process (e.g. improving the sketch completion through pen and eraser tools to get more accurate or ideal results).

When a user browses a scene sketch, he gets his interested sketch element (the *car* at the middle bottom of the scene sketch) via drawing a rectangular box surrounding it. Then our SketchMaker can automatically complete the corrupted or overlapping sketch object and save it to the user's personal favorites (see Fig. 4 (1)). When the algorithm occasionally works poorly, our system also enables users to manually adjust the sketch extraction process (e.g. improving the sketch completion through pen and eraser tools to get more accurate or ideal results). Next, the user creates a scene sketch by selecting sketch elements (the *car* and the *giraffe*) from his personal favorites or the sketch database and modifying them via dragging, rotation and zooming operations (see Fig. 4 (2)). Then, the user uses the created scene sketch to search an image (see Fig. 4 (3)). Moreover, users are able to continuously adjust the composited sketch via selecting different sketch elements from the database to obtain a more desired image (see Fig. 4 (4)-(5)).

4.2 Optional Editing Tool: the Color Panel

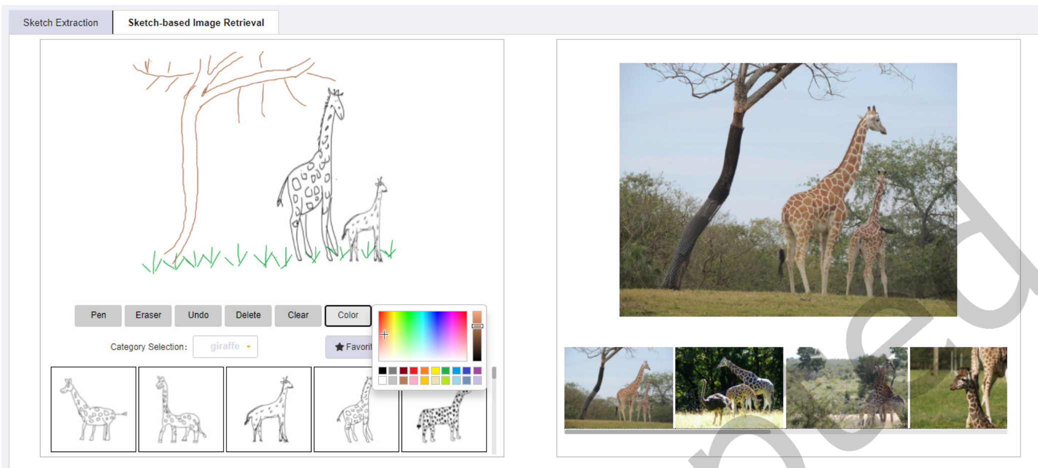

Fig. 5. The user interface of our color panel. We show the colored scene sketch composited by users and the result of sketch-based image retrieval. Users can draw colored sketches to express more information.

Although sketch has the advantage of conveying users' key ideas and objects' main characteristics in a fast and intuitive way, it it can not express color or texture-related information. Considering color information may improve the performance of some sketch-based applications. As a result, we have added a color panel in our SketchMaker system, which allows users to draw colored sketches (See Fig. 5). Although in the current retrieval mode, the introduced SBIR algorithm treats sketch as black and white pixels, we are going to combine sketch with extra user input from other modalities, such as color and texture information, in the future. We may also consider providing a flexible and hybrid query interface which integrates sketch as well as other modality input for post sketch-based multimedia applications.

4.3 SketchMaker Algorithms and Implementions

Our framework is built on the recently developed deep learning sketch processing methods, including sketch completion and sketch-based image retrieval.

4.3.1 Sketch Completion. Sketches obtained from scene sketch extraction are often incomplete, e.g., sketches of multiple overlapping objects where some objects are partially occluded. Since incompleted objects make direct reuse difficult, we design a sketch completion network to repair the corrupted sketch objects for further restoration and reuse (see Fig. 6). To point out, sketch completion [17, 40] and generation [7, 18, 43] are new but highly potential research topics. There are also many researches dedicated to improving the effect of sketch completion. Our system could be compatible with different sketch completion models in the future. Here we built our completion network based on SketchGAN because it was the first sketch completion network, and it has stable performance. Given a corrupted sketch x, the aim of sketch completion is to produce reasonable lines to fill in the missing regions and output a completed sketch y. Our network improves the state-of-the-art sketch completion model from SketchGAN [33], which applies a cascaded generative adversarial network (GAN) to jointly recognize and complete a sketch. SketchGAN mainly consists of three components: a cascaded generator, a discriminator, and an auxiliary sketch classifier.

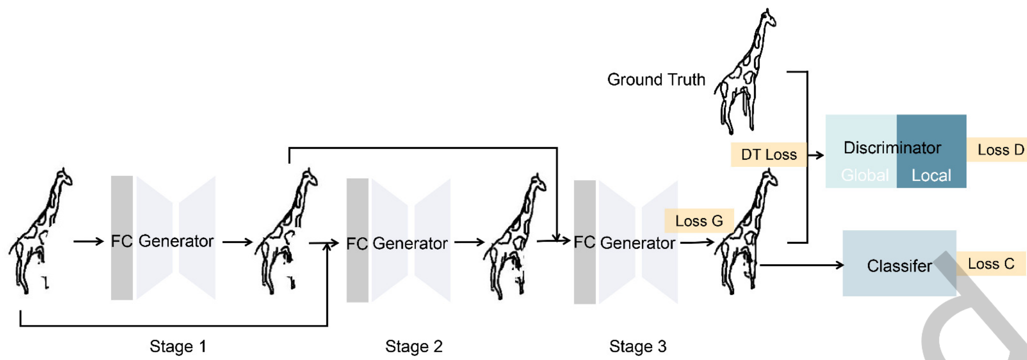

Fig. 6. Overview of our sketch completion network. The network mainly consists of three components: a cascaded generator, a discriminator, and an auxiliary sketch classifier. We built our sketch object completion model based on SketchGAN [33]. Differently, while the sketch generator in SketchGAN [33] makes use of outputs of all the previous stages, we concatenate the original input sketch only with the output of the last stage to feed to the next stage, which can improve the inference efficiency. And we adopt an effective line-promoting distance transform (DT) loss [50] to better measure the similarity between sketch strokes.

Network Architecture. We adopt the discriminator and classifier from SketchGAN. As for the generator, we make two major improvements on the original SketchGAN. The generator of SketchGAN consists of a three-stage cascaded network to iteratively refine the completion result by using the fused information of the original input and the outputs of all the previous stages. Specifically, the input of the second stage of SketchGAN is the concatenation of the original input x and the output of the first stage y_1, and the input of the third stage is the concatenation of the original input x and the output of all the previous stages y_1 and y_2. This rough concatenation of outputs of all the previous stages in SketchGAN leads to significant increase of network computing time cost. Specifically, as the number of cascaded stages of the network increases, the channel dimension of input increases at the same time, leading to higher computational cost. In this paper, we notice that in SketchGAN the output sketch of the later cascaded stage contains more fine-grained information than that of the former stage, so we improve the generator of sketch completion in our SketchMaker by concatenating the original input sketch only with the output of the last stage to feed to the next stage instead. Thus the input sketch size is fixed for all the cascaded stages, which improves the computational efficiency of the original SketchGAN. To further compare the performance of our sketch completion network with original SketchGAN [33], we show the sketch completion performance and time-efficiency on the Sketchy database on 11 object categories in Table 1. These subcategories are consistent with those in the SketchGAN. We can observe that the performance of our completion network is almost close to SketchGAN, but the running time of the algorithm drops by about 7.5%.

Table 1. Time-efficiency and completion performance on the Sketchy database on 11 object categories using our completion model and SketchGAN [33]. Our completion method achieves comparable completion performance to the original SketchGAN, but the running time drops by about 7.5%.

	Average Time-cost (ms)	Completion Performance			
		Precision(%)	Recall(%)	F-Measure(%)	Accuracy (%)
SketchGAN [33]	850	75.62	52.07	61.67	91.09
Our Completion Model	790	74.59	51.31	60.80	90.77

Loss Function. Our model is a GAN model where the generator learns to generate a completed sketch y for the input x, and the discriminator tries to distinguish the generated sketch from the real ones. Specifically, following Pix2Pix [25] the loss function of our conditional GAN can be formulated as:

$$L_{cGAN}(G, D) = \mathbb{E}_{(y)\sim p_{\text{data}}(y)}[\log(D(x, y)] + \mathbb{E}_{(x)\sim p_{\text{data}}(x),(z)\sim p_{\text{data}}(z)}[\log(1 - D(x, G(x, z))],$$

where G and D are the generator and the discriminator, and z is a Gaussian noise. We also adopt the $L1$ loss and the auxiliary sketch classification loss in SketchGAN:

$$L_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x, z)||_1]$$

$$L_{ac}(C) = \mathbb{E}[\log P(C = c|y)]$$

Furthermore, we introduce a line-promoting distance transform loss L_{DT} from [50] to our SketchMaker to better measure the similarity between sketch strokes. Compared to $L1$ loss that enforces pixel-to-pixel alignment between input sketch and the completed result, L_{DT} is able to tolerate small offsets of sketches.

The overall loss function of our model can be formulated as:

$$\min_G \max_D L(G, D) = L_{cGAN}(G, D) + \lambda_1 L_{\mathcal{L}_1}(G) + \lambda_2 L_{\text{ac}}(G) + \lambda_3 L_{DT}(G, D)$$

In our experiments, we set $\lambda_1 = 100$, $\lambda_2 = 0.5$ and $\lambda_3 = 0.1$. Compared to SketchGAN, the sketch completion of our SketchMaker can gain a completion performance increase of about $1.8\% - 2.7\%$ in recall on the same dataset used in SketchGAN on 11 classes of sketch objects. Overall, our network is more efficient than SketchGAN. SketchGAN iteratively refines a sketch by using the fused information of the input sketch and the outputs of all the previous stages, while our method only concatenates the input sketch with the output of the last stage. So, the input size of SketchGAN will become larger as stages increase, leading to inefficiency. However, the input size of our network is fixed, and time cost is not influenced by more used stages.

Generalization ability. As for the generalization ability of our completion model, we have conducted two experiments to verify that our SketchGAN [33] is able to complete sketch objects which are unseen in the training stage and sketches from different sketch datasets.

We firstly explore the generalisation ability of our completion model on unseen object categories. In particular, we train the completion model using the 11 sketch sub-categories (car (sedan), cow, horse, cat, dog, sheep, airplane, motorcycle, bicycle, songbird, pickup truck) of Sketchy, and test the model using other 50 sketch sub-categories. Fig. 7 shows some completion results, in which these sketch categories (banana, wheelchair, chair, window, mouse, gun, strawberry, table) have not been used during the training stage. Experiments show that our sketch completion method is able to repair sketches whose categories are unseen during the network's training stage.

We also conducted an experiment to explore the generalization ability of our completion model on sketches from unseen sketch databases. We train the completion model using object sketches from the Sketchy database [41], and test the model using sketches from TU-Berlin [11]. Fig. 8 shows examples of sketch completion results. Experiments show that our sketch completion method has good generalization ability to repair sketches from different sketch databases.

4.3.2 Image Retrieval using Scene Sketch. Due to the lack of scene sketch data and the limited drawing ability for novice users to sketch scenes, existing SBIR researches usually conduct object-level image retrieval, where only one object exists in each sketch or image. One exception is the recently proposed SceneSketcher [34], which uses scene sketches as query inputs to search images. We adopt the graph convolutional networks (GCNs) in SceneSketcher to explicitly model the scene sketches and images, where the objects are seen as nodes and the relationships between objects are

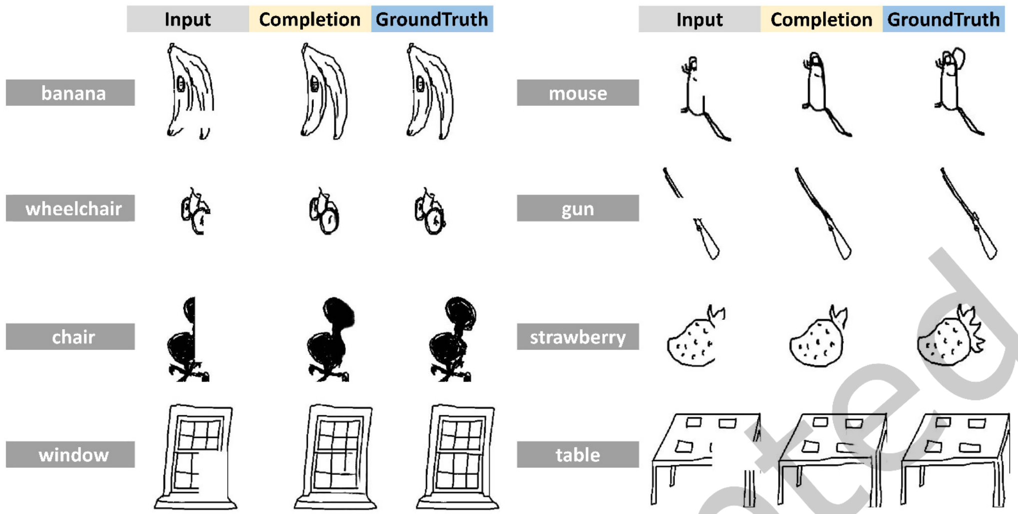

Fig. 7. Results on the Sketchy database [41] of our sketch completion model. We train the completion model using 11 sketch categories, and test the model on 50 unseen categories. Experiments show that our method is able to repair sketch categories which are unseen during the network's training stage.

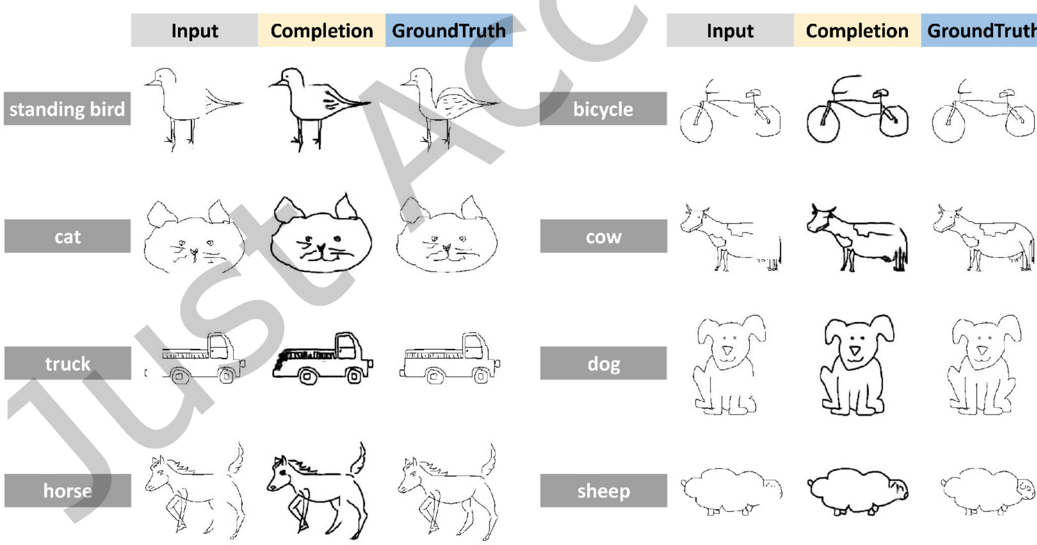

Fig. 8. Results on the Tu-berlin [11] using our sketch completion model trained on the Sketchy database [41]. Experiments show that our completion model is able to repair object sketches from different databases.

represented by edges in graphs. We use three types of features as the input of the SBIR algorithm, including object category, object visual appearance and global layout of scenes. Our SketchMaker integrates the scene sketch composition framework with the scene-level SBIR network, allowing users to continuously create and adjust the scene sketches and view the corresponding image retrieval results.

4.3.3 Data. For sketch completion, we refined SketchGAN [33] network using sketch objects from Sketchy [41] (75,471 sketches objects of 125 categories). For image retrieval, we construct our image gallery using images from SketchyCOCO dataset [15] (14,081 scene images). We also put the object-level sketch from Sketchy [41] in our sketch gallery. In user study, users browse existing scene sketches and extract their interested sketches to save in their favorites. To composite scene sketches, they can either draw a sketch element, or select it from our sketch database or their favorites.

5 USER STUDY

To evaluate the performance of our SketchMaker, we design a baseline system and a user study consisting of a series of scene sketch composition tasks, questionnaires and interviews.

5.1 Baseline System

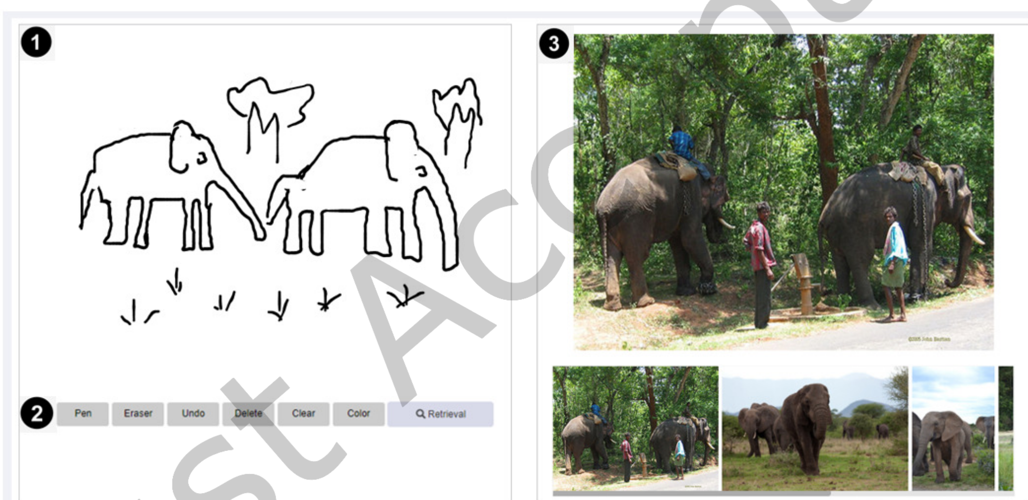

Fig. 9. The user interface of the baseline system. ❶ the scene sketch composition canvas, ❷ the scene sketch editing tools panel (Pen, Eraser, Undo, Delete, Clear, Retrieval (from left to right)), ❸ the image retrieval result display panel.

We implement a baseline system without scene sketch processing and reusing functionalities to compare with our SketchMaker. As described in Section 2, typical sketch applications require users to draw complete sketches as inputs. In our baseline system, users can draw sketch objects in scenes one-by-one for image retrieval task, but with no reuse of existing sketch elements. In other word, compared with our SketchMaker, there is no sketch extraction and reuse function in the baseline system.

The user interface of the baseline system is shown in Fig. 9. The baseline system provides: (1) the scene sketch composition canvas for users to draw sketch for their desired image, (2) the scene sketch editing tools panel (Pen, Eraser, Undo, Delete, Clear, Retrieval (from left to right)) for selecting tools for sketch creation and editing, and (3) the image retrieval result display panel.

5.2 Participants

We recruited participants via an online workgroup of our institute. We recruited 16 participants (8 males and 8 females). Their mean age was 26.313, and the SD was 3.260 (M: Mean = 27.250, SD = 3.845, F: Mean = 25.375, SD = 2.446). Before the experiment, we explained the purpose and procedures to the participants, and we let the participants try out the system a few times. Each experiment lasted about half an hour, and all of the participants reported that they have no professional drawing skills.

5.3 Tasks and Procedures

The user study consisted of two tasks, each of which involved one of the two systems: SketchMaker and the baseline system. We employ the strategy of prompting the creation of sketches from particular photos in [41]. In each task, a color image was shown to the participants as the retrieval target for 2 seconds (about 10 percent of short-term memory), and then they were asked to composite a sketch according to their rough impression to retrieve the target image. SketchMaker allows users to select and modify sketch objects from the database and use the editing tools to create a new sketch, while the baseline system only offers pencil, eraser, and undo tools to create sketches. Each participant did 6 trials (4 with given target images to retrieve and 2 free trials), and we collected 96 scene sketches composited by the 16 participants in each system.

5.4 Survey

We conducted a survey to evaluate the user experience of SketchMaker quantitatively. At the end of each task, the participants filled out the questionnaires about the experience of the two systems. The survey covers four criteria including the usability of the system (Usability Metric for User Experience, UMUX) [13], the technology acceptance model (TAM) [8] (the Perceived usefulness (PU) and Perceived ease-of-use (PEOU)), and the AI interface issue.

The UMUX consists of four items: (1) Effectiveness: [This system's] capabilities meet my requirements. (2) Satisfaction: Using [this system] is a frustrating experience. (3) Overall: [This system] is easy to use. (4) Efficiency: I have to spend too much time correcting things with [this system].

The TAM evaluates how users come to accept and use a technology. We test TAM indicators in the creation phase and the retrieval phase respectively, containing two aspects: Perceived usefulness (PU) and Perceived ease-of-use (PEOU) metrics. There are four items in PEOU: (5) Easy to learn (composition / retrieval): At this stage the operation of the system is easy to understand. (6) Easy to use (composition / retrieval): At this stage the system is easy to use. (7) Comfortable (composition / retrieval): At this stage the system does not require a lot of mental load to use. (8) Communicative (composition / retrieval): At this stage the system knows what I want. And three items in PU: (9) Useful (composition / retrieval): At this stage the system can help me (composite scene sketches / find pictures). (10) Time-saving (composition / retrieval): At this stage, the system saved me time to (composite scene sketches / find pictures). (11) Fulfilling (composition / retrieval): At this stage, the system (composites scene sketches / finds pictures) very accurately.

We introduce three indicators commonly used in the AI interface issue and AI-collaboration systems [21, 38, 42]: (12) Controllability, (13) Creativity, and (14) Degree of Freedom.

Finally, we investigate the (15) NPS (Net Promoter Score): Would you like to recommend this system to others?

At the end of each task, users completed a survey with a 7-point Likert scale ranging from highly disagree to highly agree. For the NPS, users scored between 0-10 to show whether they are willing to recommend the system to others.

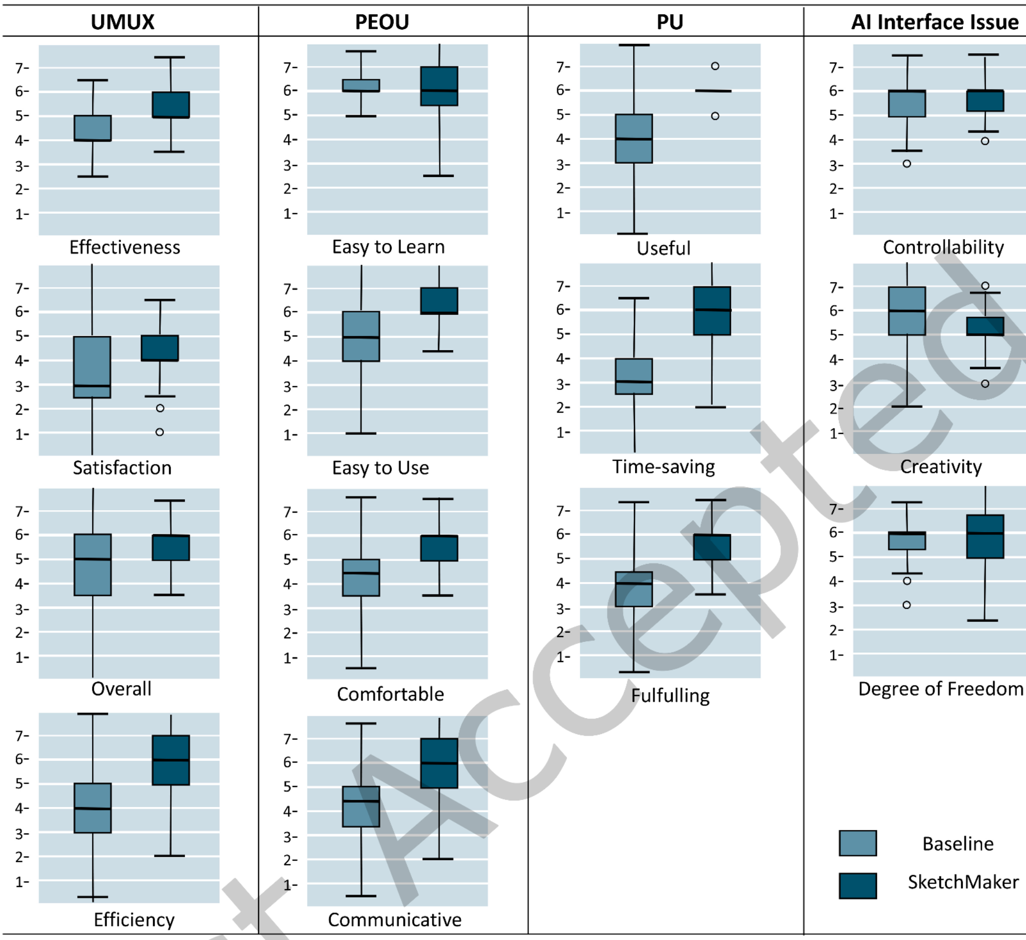

Fig. 10. Boxplots of each item in baseline system and our SketchMaker. Except for Satisfaction, Easy to Learn, Controllability and Degree of Freedom, all other items show significant differences.

5.5 Interview

Compared with usual survey such as questionnaire, interview has the advantage of allowing interviewees to express their attitudes or opinions freely. We also conducted semi-structured interviews to gain a deeper and more detailed understanding of user experience in our SketchMaker framework. In the interviews, the participants were asked about their overall impressions of SketchMaker, their thoughts on the two different sketch processing systems, and the experience of using sketch as the human-computer interaction approach. Each interview lasted for about half an hour.

6 RESULTS

From the study, we used quantitative data from the questionnaires and qualitative data from the interviews to conduct analysis for our SketchMaker.

6.1 Quantitative Results

We applied paired t-test to find whether there are significant differences between the two results of the questionnaire. We set significant level as 0.05.

Usability Metric for User Experience

Statistical significance was found in effectiveness (t=3.60, p<.01). Furthermore, *SketchMaker* (M=5.38, SD=.72) gets higher scores than *Baseline* (M=4.31, SD=.95). There was also statistical significance between *SketchMaker* and *Baseline* in efficiency (t=5.16, p<.01). The participants gave a higher score for *SketchMaker* (M=6.06, SD=.85) than *Baseline* (M=4.06, SD=1.12). Scores of satisfaction between *SketchMaker* (M=4.25, SD=1.48) and *Baseline* (M=3.50, SD=1.79) had no significant difference (t=1.44, p>.05), while significant difference existed in overall rating (t=-3.06, p<.01), where *SketchMaker* (M=5.75, SD=.58) performed significantly better than *Baseline* (M=4.69, SD=1.40).

Perceived ease-of-use

There was no statistical significance between the score of whether the system is easy to learn (t=-0.22, p>.05). Both *SketchMaker* (M=6.00, SD=1.10) and *Baseline* (M=6.06, SD=.68) do not occupy too much time to learn. We find a significant difference in ease of use (t=3.88, p<.01). Participants graded higher scores for *SketchMaker* (M=6.19, SD=.98) than *Baseline* (M=5.00, SD=1.32). Statistical significance was detected in comfortability (t=3.67,p<.01). *SketchMaker* (M=5.75, SD=.77) got a higher score than *Baseline* (M=4.38, SD=1.20). As for communicative ability, there was also statistical significance between *SketchMaker* and *Baseline* (t=4.58, p<.01). Participants thought *SketchMaker* (M=5.81, SD=.98) is more communicative than *Baseline* (M=4.25, SD=1.06).

Perceived usefulness

In terms of usefulness, we found statistical significance between the scores of the two systems (t=5.23, p<.01). The score of *SketchMaker* (M=5.97, SD=.57) is higher than *Baseline* (M=4.00, SD=1.21). We also found that *SketchMaker* (M=6.13, SD=.89) is more time-saving than *Baseline* (M=3.25, SD=1.13) (t=7.67,p<.01). Similarly, there was statistical significance in fulfilling (t=5.57, p<.01), where *SketchMaker* (M=5.81, SD=.66) was rated higher than *Baseline*(M=3.75, SD=1.29).

AI interface issue and AI-collaboration systems

The results of creativity show statistical significance between the two systems (t=-4.14, p<.01). The score of *SketchMaker* (M=5.13, SD=.89) is lower than *Baseline* (M=6.13, SD=.89). There was no statistical significance in controllability between the two systems (t=1.86, p>.05), the scores of *SketchMaker* and *Baseline* are almost the same. Similarly, there was no statistical significance in degree of freedom (t=0.64, p>.05), where the scores of *SketchMaker* (M=5.88, SD=1.01) and *Baseline* (M=5.69, SD=1.01) are almost the same.

Net Promoter Score (NPS)

The NPS of *SketchMaker* is 93.75% that fifteen participants gave ten or nine points. Moreover, the NPS of *Baseline* is -37.5% that one participant gave ten or nine points, eight participants gave eight or seven points.

6.2 Qualitative Results

In the qualitative analysis, we aim to understand the users' thoughts, comments, and suggestions beyond the quantitative results. We show some examples of scene sketches created in the study and the corresponding image query targets in Fig. 11.

Easy to Use and Simple Interface

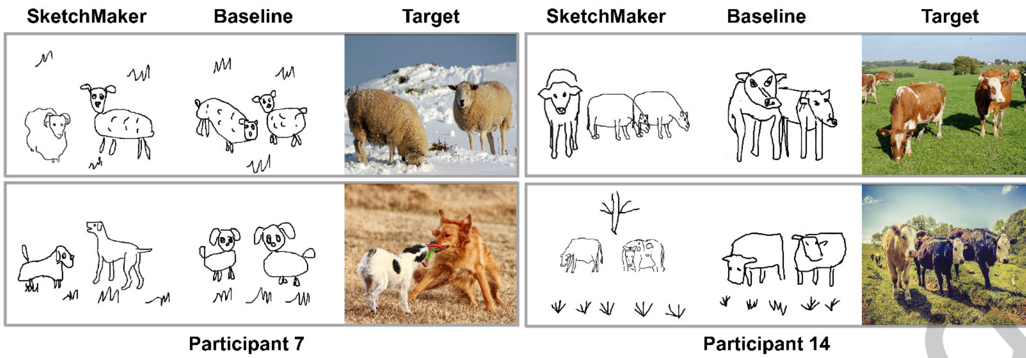

Fig. 11. Sketches from the user study. We show two sketches drawn with our SketchMaker and the baseline system for each participant. Participants 7 and 14 drew better sketches under the guidance of SketchMaker. For the users with little drawing skills, SketchMaker can provide them with inspirations and guidelines to draw better sketches.

Almost all the participants believe that SketchMaker is easy to use. For example, P6 said, "*It is convenient to assist my sketching skills. It provides sketch object extraction and completion to expand the sketch library. Thus I can use the sketch instance directly for my scene-level sketch-based image retrieval.*" P9 noted, "*I have no doubt that this software can greatly simplify the process of sketch creation and sketch-based image retrieval.*" P11 said, "*For the Baseline system, the manual drawing by the mouse is not easy for users who have insufficient experiences interacting with such devices, making the searching process almost painful; but SketchMaker eliminates the difficulty by providing sketch objects to choose from, and huge loads of work were reduced. It is the elimination of manual drawing that saved the most of time and provided most fun.*"

SketchMaker is more efficient and user-friendly

During the interview, the participants prefer our SketchMaker to the baseline system. P1 said, "*The SketchMaker is more efficient and user-friendly than the baseline. It really helps users with poor sketching skills to generate their scene sketch.*" Participants tend to like our interface. P2 said, "*Well, I have found 3 target images out of 4. I can create the scene sketch by adding existing sketch objects from my personal favorites or the sketch database, and modify them according to my intents, then move, rotate, and organize the sketches by dragging, rotating, and clicking operations.*" Fig. 11 shows some sketches drawn by the participants. Some participants like the "scene sketch processing - sketch composition - image retrieval framework", believing that it is the elimination of manual drawing that saved the most of time. P12 pointed out, "*Sometimes finding the target image by sketch can save time comparing to tag and text. Sketches can contain location, distribution, and texture information while tag and text cannot.*" Also the sketch-based image retrieval process is improved by our SketchMaker. P1 said, "*It is a concise and efficient way for the image retrieval task. I can get satisfactory retrieval results by drawing and continuously adjusting different sketches.*"

Inspiration and Creativity

The most interesting thing we have observed is that participants have inconsistent views on whether the system can enhance creativity. Compared with the baseline system, some (7 out of 16) participants think SketchMaker can give them more inspirations. That is to say, for the users with little drawing skills, the sketch data extracted from existing scene sketches offers inspirations and guidelines to help them to learn how to draw. "*The extracted sketch objects are seen as drawing*

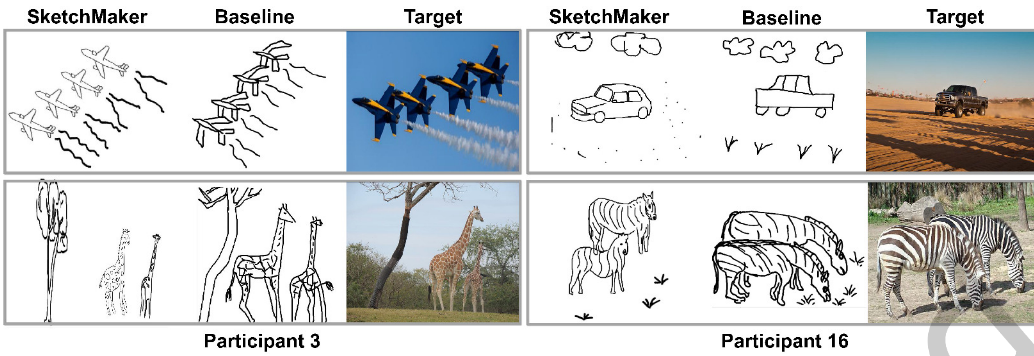

Fig. 12. Sketches from the user study. For some participants, since SketchMaker provides various drawing examples, they do not bother to create new sketches. As a result, sketches drawn by SketchMaker may have less creativity than those drawn by the baseline system.

examples. When I create a drawing, I choose to use SketchMaker because it will provide me with various materials." P10 said. And P5 said, "*SketchMaker surprised me and fully met my needs in the process of drawing creation. In addition, I can get inspiration for drawing from the objects already in the sketch gallery. I like the creative process very much.*"

On the contrary, P3 said, "*Compared to the baseline system, SketchMaker provides me with more sketch objects to select from and use, providing me with a lot of convenience. But it also limited my creativity. Since there are sketch objects available on site, I do not bother to create new sketches.*" We show sketches drawn by P3 in Fig. 12. What P3 pointed out indeed makes sense to some extent. We have found in our SketchMaker which offers lots of sketch data, even though some participants tend to not use the sketch objects in the database and create sketches by themselves, their sketch drawings are inadvertently similar to the sketches in the sketch database. The participants' drawing style is obviously affected by the style of the sketches in the database. Sketches drawn with the baseline system have more style diversities in aspects of smoothness of strokes, abstractness (more / less details), degrees of alignments between sketches and real objects, etc.

I will use it when searching for complex images

During the interview, when asking the participants if they will use this system and recommend it to others, answers vary from user to user. All the participants indicate that they will recommend our system to other people, and most of them state that they will use our SketchMaker under certain circumstances. P3 said, "*When I know exactly how the scene sketch in my mind looks like, or which picture I am looking for, I will use SketchMaker to quickly composite a scene or use it to query pictures.*" Different from P3, P2 said, "*Maybe when I have an impression on the target picture, I have seen the picture before, then I want to retrieve the picture by quickly constructing a rough image layout with sketch objects in my favorites and databases.*"

Very few users (2 out of 16) said that they would not use our system frequently, but they would also recommend it to others to experience. P13 said, "*I will probably not use this system currently because searching images is not an urging need for me. But I would like to recommend this system to others and let them experience this interesting technique.*" P7 said, "*This system to me is more like a 'marked but not used' type, I will recommend it to my friends, but I don't think I will frequently use this system. Maybe children and UI designers will use it. Anyway, it is very fun.*"

7 QUANTITATIVE STUDY: IMPROVEMENTS OF SBIR WITH SKETCHMAKER

We have conducted another user study to verify the effectiveness of our SketchMaker to help non-professionals to composite scene sketches and improve sketch-based image retrieval. In the study, we have collected an image gallery containing 210 test images from SketchyCOCO [14] dataset. During the user study, we analysis the results from the perspectives of time effectiveness and the retrieval accuracy.

7.1 Tasks and Procedures

We recruited 4 participants. Before the experiment, we explained the purpose and procedures to participants, and we let the them try out the system a few times. Similar to the previous user study in Section 5, the quantitative study also consists two tasks, each of which involved one of the two systems: SketchMaker and the baseline system. We selected 4 scene-level images and showed these images to the participants as the retrieval target for 2 seconds. Then we asked them to composite a sketch according to their rough impression to retrieve the target image. We finally collected 16 scene sketches composited by the 4 participants using each system.

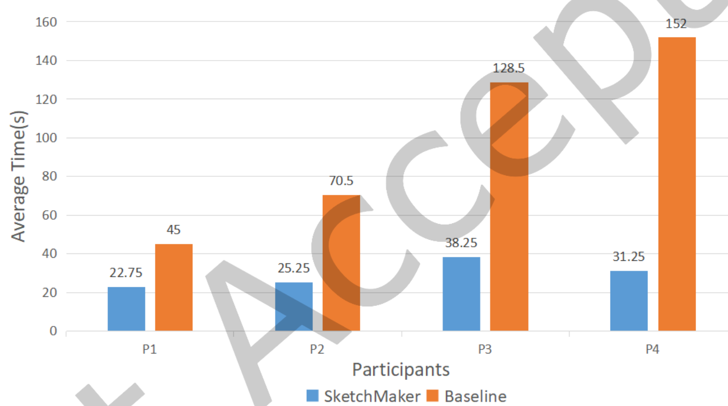

Fig. 13. The average time of compositing one scene sketch using SketchMaker and baseline system.

7.2 Result

During the user study, we have recorded the time of user compositing each scene sketch. Users' average composition time is shown in Fig.13. Average time spending on each retrieval trial (including scene sketch composition and sketch-based image retrieval) and the Top-K image retrieval results are shown in Tab. 2. In addition, we display four scene sketch composition and image retrieval examples in Fig. 14.

Users need to draw sketch objects one-by-one in the baseline system, while they can either draw sketches or reuse the existing sketch elements pre-stored in the sketch gallery in our SketchMaker, which may speed up the drawing process and improve the image retrieval performance. On the other hand, one of the most important steps in the scene sketch retrieval process is to identify **the category of each object sketch** in the input scene sketch. The categories of the sketch elements in our SketchMaker are known, while the categories of object sketches drawn by users need to be inferred by the retrieval algorithm. Thus, there is a sketch object recognition loss, which further leads to a decrease in the accuracy of sketch-based image retrieval in the baseline system. Therefore, the retrieval performance of SketchMaker outperforms is significantly better than the baseline system. According to the result of the user study, we show that SketchMaker significantly shortens

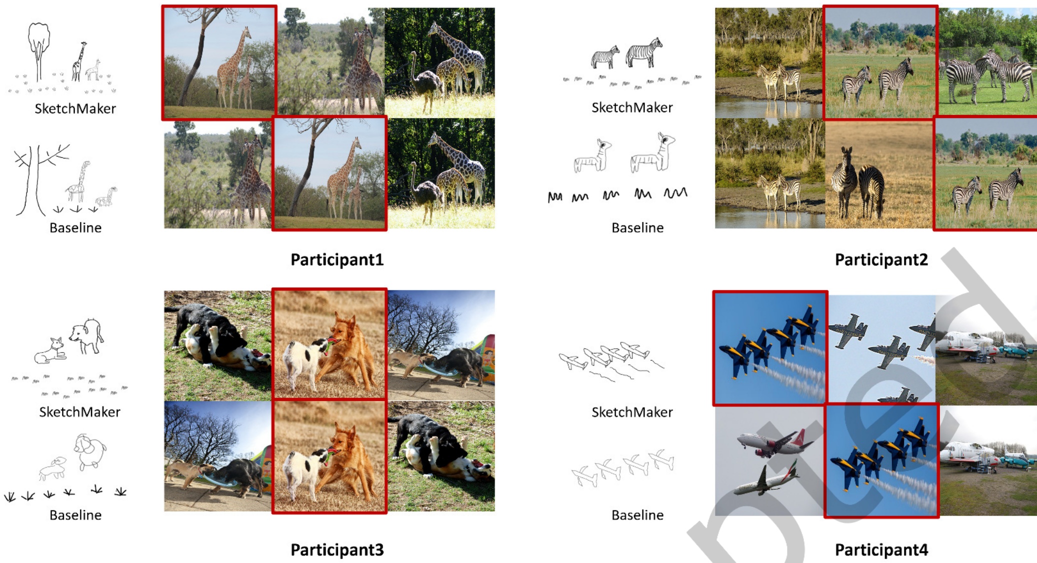

Fig. 14. Scene sketch compositions and their top-3 retrieved images. For each participant, we display two trails (one with SketchMaker and one with Baseline system). The ground truth matches are highlighted with red rectangles.

the time of user compositing scene sketches (from 99s to 29.375s on average). Moreover, the top-1, top-3 and top-5 image retrieval accuracy of using SketchMaker are all higher than those using the baseline system, which demonstrates that our SketchMaker could speed up the scene sketch composition process and improves sketch-based image retrieval performance at the same time.

Table 2. The result of average scene sketch composition time and retrieval accuracy using our SketchMaker and the baseline system.

	Average Time (s)	Accuracy (%)		
		Top-1	Top-3	Top-5
SketchMaker	29.375	37.50	81.25	100.00
Baseline	99.000	18.75	75.00	93.75

8 OTHER APPLICATION: SKETCH-BASED VIDEO LOCALIZATION

Our SketchMaker system aims to offer users with no sketching skills a user-friendly way of preparing input sketches, and thus promotes the practical application of sketch-oriented applications. We have shown how our interactive sketch composition framework can facilitate sketch-based image retrieval in the previous sections, and here we will present another application, sketch-based video localization (SBVL), to verify the effectiveness of our scene sketch composition workflow.

Temporal activity localization in video is a recently emerging research problem, and it gains lots of attention in the last several years [16]. While traditional video retrieval task aims at finding the particular pre-processed video clip, which is inconsistent with users' query input in terms of video targets, the goal of video localization is to search the start/end time of the period from untrimmed videos that match the input. The existing video localization methods usually use text as query

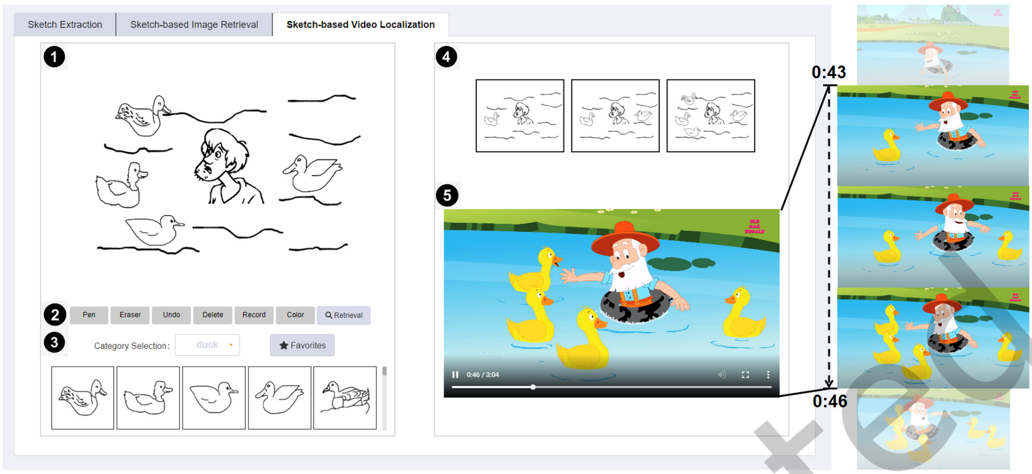

Fig. 15. The user interface of our sketch-based video localization (SBVL) Page. The SBVL Page has: ❶ the scene sketch composition canvas, ❷ the scene sketch editing tools panel, ❸ the sketch object browsing and selection panel, ❹ the sketch sequence display panel, and ❺ the video localization result display panel.

input, which has the characteristic of ambiguity for video content. Compared to text, sketch can capture the semantic information and the fine-grained visual appearance of the target video clip in a way. In this section, we design a video localization scheme which enables finding the target fine-grained video clip via using the composited scene sketch of our SketchMaker as query input.

Specifically, we propose a three-stage workflow for SBVL as follows:

(1) **Step 1. Scene sketch composition:** This function enables users to create new scene sketches in a simple and fast way. Users are allowed to reuse existing sketch objects through dragging, moving, zooming in / out and rotating operations etc. Several drawing tools, e.g. Pen, Eraser, Undo, Clear are also offered for drawing new objects.
(2) **Step 2. Sketch sequence generation:** This function allows users to create a sequence of scene sketches as the query input to find their desired moment in a video by repeating **Step 1.** The sketch elements represent the corresponding characters or objects in the video and the movements are depicted by the variation in the sketch sequence.
(3) **Step 3. Video localization:** This function allows users to find their desired moment in the video by using their composited sketch sequences as input.

Fig. 15 shows the user interface of our SketchMaker for SBVL. The same as the Sketch-based Image Retrieval Page, the SBIV Page also has the scene sketch composition canvas (Fig. 15 ❶), the scene sketch editing tools panel, (Fig. 15 ❷), the sketch object browsing and selection panel (Fig. 15 ❸). Particularly, there is a sketch sequence display panel (Fig. 15 ❹) and a video localization result display panel (Fig. 15 ❺) in our SBVL Page, supporting users to create a sequence of scene sketches as input to localize a moment in the video.

9 DISCUSSION, LIMITATIONS AND FUTURE WORK

In this section, we discuss the design implications from our study and shortcomings in our current work.

The sketch object extraction is useful

From the quantitative investigation and interviews, most of the participants believe our sketch object extraction framework is of great significance and effectiveness for sketch data restoration and reuse. P1 said. "*I can get the sketch object by sketch object completion. I will reuse these data when I want to retrieve images. This interactive and convenient way of sketch composition makes me more willing to use sketches to search for images. I hope more sketch-related applications could be integrated in this framework besides image retrieval.*"

SketchMaker is fun

Surprisingly, more than half of the participants have mentioned that our SketchMaker is interesting or fun. "*It is great fun. Seeing the poorly drawn sketches is already great fun, using such pictures to search 'real' images provides even more fun.*" P13 said. Just as P13 pointed out in Section 6.2, even though he does not have the need to frequently search images, he will recommend SketchMaker to others to experience this interesting system. P4 said, "*I think it is absolutely fantastic. I love the object extraction and scene sketch composition process itself. Despite that I do not have the image retrieval task, I will also play with the interesting sketches. I am poor in drawing, but I can make beautiful sketches with the help of SketchMaker. I really enjoyed it.*"

Sharing the favorites and sketch data

In our SketchMaker, users are able to composite the scene sketch in their mind quickly by reusing the sketch object data from their personal favorites and sketch database. The participants suggested that more sketch objects of different categories and drawing styles should be added to the database to offer more diverse compositing materials. "*The storage of sketch objects in the system needs to be categorized more carefully and enlarged, in order to help the users find the most desired one.*" P14 said. Furthermore, users hope their personal favorites could be shared with each other to obtain more sketch data. Moreover, the participants noted that the real-time scene sketches composited by them should also be stored and shared by category, so that more information could be exchanged between them.

User Interface Functionality

Most participants think that the user interface of our SketchMaker is user-friendly, but there are still some other interface suggestions. P13 said, "*The storage of the demos needs to be categorized and enlarged, in order to help the users find the needed one.*" Two out of the 16 participants pointed out that SketchMaker should be able to keep user history for future re-editing. There are also some other interface suggestions. For example, P9 said, "*When I make a mistake, I hope I can undo the current stroke instead of deleting the whole object or clearing the picture.*", and P16 said, "*It would be better if I have a pen instead of using a mouse on computers or using my fingers on touch screens.*"

Limitations and Future Work

Although our SketchMaker is successful in promoting sketch data reuse and helping users with little drawing skills composite scene sketches in an interactive and efficient way, We have identified three limitations of our system:

First, in the Sketch Extraction Page, although promising performance has been achieved, there are still some failure cases. For example, the completion model may fails to cope with different styles of sketches and plenty of sketch details. In addition, when the missing area of sketch objects is too large, the sketch completion method may produce poor results. We will improve the user interface and sketch processing algorithms to strengthen systems' functions and offer better user experiences in our future work.

Second, we use image retrieval and video localization with scene sketches as applications to verify our sketch composition workflow, but there are many sketch-oriented applications besides SBIR. We will investigate other popular sketch applications (such as sketch-based image synthesis, sketch-based 3D retrieval, sketch-based text generation, etc.) in the future, and integrate more sketch-based applications with our sketch composition system to extend the practical applications of sketch in daily life.

Third, our system can store the newly composited sketches, however, the relationships between objects in scene sketches have not been analyzed. We will establish relationships between various scene sketches and offer more flexible sketch composition modes by promoting the restoration and reuse of the whole or partial scene sketches in future.

As for the study design, we also identified two limitations in our user study:

First, sketches exist in different forms and styles (e.g. storybooks, comics, storyboards, animations, etc.), users may have different sketch process needs under different circumstances. In future work, we will investigate in-depth users' needs and experience in intelligence sketch processing and applications to improve our SketchMaker system.

Second, participants' experience may be influenced by the initial impressions and freshness of using a new system. We will study users' long-term experience of using our SketchMaker to obtain more improved proposals.

10 CONCLUSION

This paper presents SketchMaker, a sketch processing system that extracts sketch objects from existing scene sketches, helps users with no drawing skills to create scene sketches, and integrates sketch composition with sketch-based applications, e.g. image retrieval and video localization. Our user study revealed that our SketchMaker (1) is useful and easy to use, (2) gives inspirations and guidelines to people with no drawing skills on how to draw a good sketch efficiently, and (3) provides much fun during the exploration process. We have also conduct a pilot study of integrating our SketchMaker with video localization task, and we will further investigate the effect of our SketchMaker system on video localization in the future work. The SketchMaker system can also improve the restoration and reuse of existing sketch data, reduce the difficulty of drawing for non-professionals, and inspire related sketch-based applications.

Future work includes comprehensive study on the needs and requirements of users in sketch processing, and integration of our SketchMaker with more sketch-based applications.

REFERENCES

[1] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2008. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST)*. ACM, Monterey CA USA, 151–160.

[2] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2009. EverybodyLovesSketch: 3D sketching for a broader audience. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology (UIST)*. ACM, Victoria BC Canada, 59–68.

[3] Jared N Bott and Joseph J Laviola Jr. 2015. The WOZ recognizer: A wizard of Oz sketch recognition system. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 3 (2015), 1–38.

[4] Adrien Bousseau, Theophanis Tsandilas, Lora Oehlberg, and Wendy E Mackay. 2016. How novices sketch and prototype hand-fabricated objects. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 397–408.

[5] Lluis Castrejon, Yusuf Aytar, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Learning aligned cross-modal representations from weakly aligned data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Caesars Palace, 2940–2949.

[6] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. 2009. Sketch2Photo: internet image montage. *ACM Transactions on Graphics* 28, 5 (2009), 124:1–10.

[7] Yajing Chen, Shikui Tu, Yuqi Yi, and Lei Xu. 2017. Sketch-pix2seq: a model to generate sketches of multiple categories. *arXiv preprint arXiv:1709.04121* (2017).

[8] Fred D Davis, Richard P Bagozzi, and Paul R Warshaw. 1989. User acceptance of computer technology: a comparison of two theoretical models. *Management science* 35, 8 (1989), 982–1003.

[9] N Davis, C Hsiao, Kunwar Yashraj Singh, Brenda Lin, and Brian Magerko. 2017. Quantifying collaboration with a co-creative drawing agent. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7, 4 (2017), 1–25.

[10] Daniel Dixon, Manoj Prasad, and Tracy Hammond. 2009. iCanDraw?: A Methodology for Using Assistive Sketch Recognition to Improve a User's Drawing Ability. In *ACM Symposium on User Interface Software and Technology (UIST) Posters*, Vol. 10. ACM, Vancouver Canada.

[11] Mathias Eitz, James Hays, and Marc Alexa. 2012. How do humans sketch objects? *ACM Transactions on Graphics* 31, 4 (2012), 1–10.

[12] Jennifer Fernquist, Tovi Grossman, and George Fitzmaurice. 2011. Sketch-sketch revolution: an engaging tutorial system for guided sketching and application learning. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST)*. ACM, Santa Barbara California USA, 373–382.

[13] Kraig Finstad. 2010. The usability metric for user experience. *Interacting with Computers* 22, 5 (2010), 323–327.

[14] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. 2020. SketchyCOCO: Image Generation From Freehand Scene Sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Seattle Washington USA, 5174–5183.

[15] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. 2020. SketchyCOCO: Image Generation From Freehand Scene Sketches. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Seattle Washington USA, 5174–5183.

[16] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*. IEEE, Venice Italy, 5267–5275.

[17] Arnab Ghosh, Richard Zhang, Puneet K Dokania, Oliver Wang, Alexei A Efros, Philip HS Torr, and Eli Shechtman. 2019. Interactive sketch & fill: Multiclass sketch-to-image translation. In *Proceedings of the IEEE international conference on computer vision*. IEEE, Seoul Korea, 1171–1180.

[18] David Ha and Douglas Eck. 2017. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477* (2017).

[19] Tracy Hammond, Shalini Priya Ashok Kumar, Matthew Runyon, Josh Cherian, Blake Williford, Swarna Keshavabhotla, Stephanie Valentine, Wayne Li, and Julie Linsey. 2018. It's Not Just about Accuracy: Metrics That Matter When Modeling Expert Sketching Ability. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 8, 3 (2018), 1–47.

[20] Tracy Hammond and Brandon Paulson. 2011. Recognizing sketched multistroke primitives. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 1, 1 (2011), 1–34.

[21] Melanie Hartmann. 2009. Challenges in Developing User-Adaptive Intelligent User Interfaces (IUI). In *LWA*. Citeseer, ABIS–6.

[22] Forrest Huang and John F Canny. 2019. Sketchforme: Composing sketched scenes from text descriptions for interactive applications. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST)*. ACM, New Orleans LA USA, 209–220.

[23] Forrest Huang, John F Canny, and Jeffrey Nichols. 2019. Swire: Sketch-based user interface retrieval. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–10.

[24] Emmanuel Iarussi, Adrien Bousseau, and Theophanis Tsandilas. 2013. The drawing assistant: Automated drawing guidance and feedback from photographs. In *ACM Symposium on User Interface Software and Technology (UIST)*. ACM, ACM, ST ANDREWS UK.

[25] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Honolulu Hawaii, 1125–1134.

[26] Gabe Johnson, Mark Gross, Ellen Yi-Luen Do, and Jason Hong. 2012. Sketch it, make it: sketching precise drawings for laser cutting. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. ACM, Aubtin Texas USA, 1079–1082.

[27] Levent Burak Kara and Thomas F Stahovich. 2004. Hierarchical parsing and recognition of hand-sketched diagrams. In *Proceedings of the 17th annual ACM symposium on User interface software and technology (UIST)*. ACM, Santa Fe NM, 13–22.

[28] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014. Kitty: sketching dynamic and interactive illustrations. In *Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST)*. ACM, Honolulu Hawaii USA, 395–405.

[29] Yongkwan Kim and Seok-Hyung Bae. 2016. SketchingWithHands: 3D sketching handheld products with first-person hand posture. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, Tokyo Japan, 797–808.

[30] Yong Jae Lee, C Lawrence Zitnick, and Michael F Cohen. 2011. Shadowdraw: real-time user guidance for freehand drawing. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 1–10.

[31] Yuwei Li, Xi Luo, Youyi Zheng, Pengfei Xu, and Hongbo Fu. 2017. SweepCanvas: Sketch-based 3D prototyping on an RGB-D image. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST)*. ACM, Québec City QC Canada, 387–399.

[32] Yuyu Lin, Jiahao Guo, Yang Chen, Cheng Yao, and Fangtian Ying. 2020. It Is Your Turn: Collaborative Ideation With a Co-Creative Robot through Sketch. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–14.

[33] Fang Liu, Xiaoming Deng, Yu-Kun Lai, Yong-Jin Liu, Cuixia Ma, and Hongan Wang. 2019. Sketchgan: Joint sketch completion and recognition with generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Long Beach CA USA, 5830–5839.

[34] Fang Liu, Changqing Zou, Xiaoming Deng, Ran Zuo, Yu-Kun Lai, Cuixia Ma, Yong-Jin Liu, and Hongan Wang. 2020. SceneSketcher: Fine-Grained Image Retrieval with Scene Sketches. In *European Conference on Computer Vision*. Springer, Springer, Glasgow UK, 718–734.

[35] Yusuke Matsui, Takaaki Shiratori, and Kiyoharu Aizawa. 2016. DrawFromDrawings: 2D drawing assistance via stroke interpolation with a sketch database. *IEEE transactions on visualization and computer graphics* 23, 7 (2016), 1852–1862.

[36] Winsor McCay. 1914. Gertie the Dinosaur. https://www.youtube.com/watch?v=TGXC8gXOPoUl.

[37] Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, and Bongwon Suh. 2018. I lead, you help but only with enough details: Understanding user experience of co-creation with artificial intelligence. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montréal Canada, 1–13.

[38] Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, and Bongwon Suh. 2018. I lead, you help but only with enough details: Understanding user experience of co-creation with artificial intelligence. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montréal Canada, 1–13.

[39] Lukas Probst, Ihab Al Kabary, Rufus Lobo, Fabian Rauschenbach, Heiko Schuldt, Philipp Seidenschwarz, and Martin Rumo. 2018. SportSense: User Interface for Sketch-Based Spatio-Temporal Team Sports Video Scene Retrieval.. In *IUI Workshops*. ACM, Tokyo Japan.

[40] Yonggang Qi, Guoyao Su, Pinaki Nath Chowdhury, Mingkang Li, and Yi-Zhe Song. 2021. SketchLattice: Latticed Representation for Sketch Manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, virtual, 953–961.

[41] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. 2016. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics* 35, 4 (2016), 119.

[42] Yang Shi, Nan Cao, Xiaojuan Ma, Siji Chen, and Pei Liu. 2020. EmoG: Supporting the Sketching of Emotional Expressions for Storyboarding. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–12.

[43] Guoyao Su, Yonggang Qi, Kaiyue Pang, Jie Yang, Yi-Zhe Song, and CVSSP SketchX. 2020. SketchHealer: A Graph-to-Sequence Network for Recreating Partial Human Sketches.. In *BMVC*. Springer, UK.

[44] Qingkun Su, Xue Bai, Hongbo Fu, Chiew-Lan Tai, and Jue Wang. 2018. Live sketch: Video-driven dynamic deformation of static drawings. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montréal Canada, 1–12.

[45] Sarah Suleri, Vinoth Pandian Sermuga Pandian, Svetlana Shishkovets, and Matthias Jarke. 2019. Eve: A sketch-based software prototyping workbench. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow UK, 1–6.

[46] Claudiu Tanase, Ivan Giangreco, Luca Rossetto, Heiko Schuldt, Omar Seddati, Stephane Dupont, Ozan Can Altiok, and Metin Sezgin. 2016. Semantic sketch-based video retrieval with autocompletion. In *Companion Publication of the 21st International Conference on Intelligent User Interfaces (IUI)*. ACM, Sonoma California USA, 97–101.

[47] Weihao Xia, Yujiu Yang, and Jing-Hao Xue. 2019. Cali-sketch: Stroke calibration and completion for high-quality face image generation from poorly-drawn sketches. *arXiv preprint arXiv:1911.00426* (2019).

[48] Jun Xie, Aaron Hertzmann, Wilmot Li, and Holger Winnemöller. 2014. PortraitSketch: Face sketching assistance for novices. In *Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST)*. ACM, Honolulu Hawaii USA, 407–417.

[49] Yuxiang Ye, Yijuan Lu, and Hao Jiang. 2016. Human's scene sketch understanding. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ACM, New York USA, 355–358.

[50] Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L. Rosin. 2019. APDrawingGAN: Generating Artistic Portrait Drawings From Face Photos With Hierarchical GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Long Beach CA, 355–358.

[51] Qian Yu, Feng Liu, Yi Zhe Song, Tao Xiang, Timothy M. Hospedales, and Change Loy Chen. 2016. Sketch Me That Shoe. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Caesars Palace Las Vegas USA, 799–807.

[52] Changqing Zou, Haoran Mo, Chengying Gao, Ruofei Du, and Hongbo Fu. 2019. Language-based colorization of scene sketches. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–16.

[53] Changqing Zou, Qian Yu, Ruofei Du, Haoran Mo, Yi-Zhe Song, Tao Xiang, Chengying Gao, Baoquan Chen, and Hao Zhang. 2018. Sketchyscene: Richly-annotated scene sketches. In *Proceedings of the European Conference on Computer Vision*. Springer, Munich Germany, 421–436.