SketchGAN: Joint Sketch Completion and Recognition with Generative Adversarial Network

Fang Liu\textsuperscript{1,2} Xiaoming Deng\textsuperscript{1} Yu-Kun Lai\textsuperscript{3} Yong-Jin Liu\textsuperscript{4} Cuixia Ma\textsuperscript{1,2} Hongan Wang\textsuperscript{1,2} \textsuperscript{*}
\textsuperscript{1} State Key Laboratory of Computer Science and Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences
\textsuperscript{2} University of Chinese Academy of Sciences \textsuperscript{3} Cardiff University \textsuperscript{4} Tsinghua University

Abstract

Hand-drawn sketch recognition is a fundamental problem in computer vision, widely used in sketch-based image and video retrieval, editing, and reorganization. Previous methods often assume that a complete sketch is used as input; however, hand-drawn sketches in common application scenarios are often incomplete, which makes sketch recognition a challenging problem. In this paper, we propose SketchGAN, a new generative adversarial network (GAN) based approach that jointly completes and recognizes a sketch, boosting the performance of both tasks. Specifically, we use a cascade Encode-Decoder network to complete the input sketch in an iterative manner, and employ an auxiliary sketch recognition task to recognize the completed sketch. Experiments on the Sketchy database benchmark demonstrate that our joint learning approach achieves competitive sketch completion and recognition performance compared with the state-of-the-art methods. Further experiments using several sketch-based applications also validate the performance of our method.

1. Introduction

Sketch is a natural symbol to express abstract ideas in a straightforward way, which is widely used in computer vision \cite{deng2018sketchy, sun2017sketch}, multimedia reorganization \cite{liu2019sketch, li2020sketch} and human computer interactions \cite{li2018sketch}. Hand-drawn sketch recognition is a fundamental problem in many sketch-based applications. However, they are often incomplete, e.g., quick interim sketches, and sketches of multiple overlapping objects where some objects are partially occluded. In this paper, we explore the sketch completion problem, which will benefit related research such as sketch recognition, sketch editing and sketch-based image retrieval (SBIR).

Sketch completion aims to infer reasonable lines to fill in missing strokes in a sketch. An ideal sketch contains perceptually closure contour and uses a sparse set of lines to capture main shape characteristics. Although sketch recognition \cite{liu2019sketch} and classification \cite{li2018sketch, li2019sketch} have achieved much progress in the last decade, little attention is paid to sketch completion. Hand-drawn sketches lack texture and contextual information, and are generally known to be more ambiguous than natural images. Therefore, image completion methods designed for color images do not work well for sketches directly. Moreover, the sketches of the same object may be drawn in diverse styles due to the nature of freehand sketches, which makes sketch completion very challenging.

Sketch is closely related to contour, both of which are visually closed outlines of objects using black and white pixels. The major difference between sketch and contour is that contour always keeps consistent with the corresponding real image, while sketch has more diversity \cite{liu2018sketch} and abstraction levels \cite{liu2019sketch} in appearance. A lot of effort has been made on contour completion \cite{liu2018sketch, li2020contour, zhao2020contour}, aiming to either extract perceptually-salient contours from images or to find the boundaries of objects and surfaces in a scene. The contour completion methods cannot be applied to sketch completion directly since contour completion enforces accurate alignment with object boundaries in original images, while sketches are not associated with specific images.
Another related problem is image completion. However, even the state-of-the-art convolutional neural network (CNN) based methods \cite{18,20} designed for image completion have poor performance on sketch completion due to the lack of color and texture context.

In this paper, we propose a generative adversarial network (GAN) for sketch completion, called SketchGAN. Our method is not category specific, and can complete input sketches of different categories. Our key idea is to jointly conduct sketch completion and recognition tasks. The generator of our SketchGAN consists of a multi-stage cascade network, where each stage consists of a conditional GAN \cite{21}, and the cascade structure can further boost the performance of a single stage. The first stage uses an incomplete sketch as input, and transforms it to a roughly completed output. A later stage concatenates the original input and all the output sketches of the previous stages, and feeds them to the generator network in the current stage. Since we aim to design a general sketch completion framework suitable for multi-category sketch completion, motivated by multi-task learning in which auxiliary tasks can often improve the performance of the main task, we adopt sketch recognition as the auxiliary task. Conceptually, the high level recognition task benefits from the global context structure of the sketch completion results, and in turn training them together promotes better sketch completion.

Experiments on a widely-used dataset show that our method is superior to state-of-the-art methods, and the sketch completion and recognition tasks benefit each other. An example of our joint sketch completion and recognition is illustrated in Figure 1, which demonstrates the mutual benefits of addressing both tasks simultaneously. We further demonstrate that our method is very helpful to improve the performance of downstream applications such as incomplete sketch recognition and sketch editing.

Our contributions are summarized as follows.

1. To the best of our knowledge, we are the first to solve the problem of sketch completion, which can inspire further sketch-based research.

2. We propose a new network architecture for sketch completion, namely SketchGAN, which handles sketches of different categories. Our method jointly conducts sketch completion and an auxiliary sketch recognition task, and we find that sketch completion and recognition tasks benefit each other.

3. Experiments demonstrate that the output of SketchGAN can enhance the performance of typical sketch applications such as incomplete sketch recognition and sketch editing.

2. Related Work

Contour completion. Contour completion plays an important role in visual perception, and it aims to group fragmented low-level edge elements into perceptually coherent and salient contours \cite{34}. Several learning based methods for edge detection are proposed recently \cite{2,57}. Xie et al. \cite{43} use a convolutional network to pool information from the whole image. These local \cite{23} or mid-level contour detection models \cite{34,42,50} often ignore the important problem of contour closure, which has been widely studied more recently \cite{55}. Instead of using segmentation, they propose a novel approach for contour completion that results in closure contour in the contour domain.

Our paper focuses on sketch completion, which aims to inpaint corrupted sketches or hand-drawn lines with missing lines instead of extracting edges from natural images in contour completion. Sketch completion is more challenging than contour completion, due to the ambiguity nature, large variations and different abstraction levels among sketches.

Image completion. Another related problem to sketch completion is image completion, both of which aim to fill in missing strokes or regions. Previous image completion methods can be classified into diffusion-based image synthesis and patch-based methods. Diffusion-based methods apply filters to propagate the local image appearance near the target regions to fill them \cite{4,25}. Patch-based approaches are proposed for larger missing areas and complicated image completion \cite{12,18,24}. Unfortunately, depending on the hand-crafted features, these traditional image completion approaches can only repair small corrupted areas, and cannot generate new objects which do not exist in the original corrupted images. Recently, convolutional networks have been applied to image completion \cite{37,51}. A joint multiplanar autoregressive and low-rank based approach has been proposed for image completion from random sampling \cite{26}. Beak et al. \cite{3} present a multiview image completion method that provides geometric consistency among different views by propagating spatial structures.

Sketch completion and image completion both belong to generative models. Compared to image completion, sketch completion is more challenging due to the fact that hand-drawn sketches lack color and contextual information for sketch understanding. Although these image completion methods work very well on incomplete natural images, they are not suitable for the sketch completion task due to the style and content gap between natural images and sketches.

Generative adversarial networks (GANs). GANs \cite{15} achieved impressive results for image generation \cite{5}, image completion \cite{20}, and image editing \cite{38}. GAN trains two networks, a generative model $G$ and a discriminative model $D$. Upon convergence, $D$ can reject images that look fake, and $G$ can produce high-quality images, which can fool $D$. A variety of GANs have been proposed for im-
Figure 2. Overview of our SketchGAN architecture. Our network is built upon conditional GANs with a cascade encode-decoder architecture. The network uses an incomplete sketch as input, and predicts a completed sketch and its sketch classification label. We illustrate a three-stage network which is the chosen architecture. In the first stage, we use the original incomplete sketch $x$ as input and get a roughly completed sketch output $y_1$. In the next stage, we feed $x$ and $y_1$ to the generator and get the output of the second cascade stage, $y_2$. Finally, the outputs of all the previous cascade stages $y_1$ and $y_2$, together with the original $x$, are fed to the generator. We feed the output of the final cascade stage $y_3$ to the discriminator and the sketch classifier.

3. Methodology

The architecture of our sketch completion network SketchGAN is shown in Figure 2. Briefly speaking, given an incomplete sketch $x$ as input, our architecture iteratively refines it through multiple stages (3 stages in the illustration). At each stage, the generator $G$ takes the input $x$ and outputs from the previous stages (if applicable) and produces a better completed sketch $G(x)$. The generated sketch image $G(x)$ of the last stage is judged by the discriminator network $D$ to be real or fake. Finally, we set a sketch recognition network as an auxiliary network to improve the main target of sketch completion.

3.1. GAN and Recognition Losses

Our approach is based on generative adversarial networks (GANs) [15]. A GAN is a generative model that learns a mapping from a random noise vector $z$ to an output image $y : G : z \rightarrow y$. In contrast, a conditional GAN learns a mapping from observed image $x$ and random noise vector $z$ to an output image $y : G : \{x, z\} \rightarrow y$. Denote
by \( p_z \) and \( p_{\text{data}} \) the prior distributions of \( z \) and the real data \( x \). The objective of a conditional GAN can be formulated as:

\[
\mathcal{L}_{\text{cGAN}}(G, D) = E_{x,y \sim p_{\text{data}}(x, y)}[\log D(x, y)] + E_{x \sim p_{\text{data}}(x), z \sim p_z(z)}[\log(1 - D(x, G(x, y)))] ,
\]

where a generator \( G \) is trained to minimize this objective against an adversarial \( D \) that tries to maximize it, i.e.

\[
G^* \in \arg \min \max_{D} \mathcal{L}_{\text{cGAN}}(G, D).
\]

We further utilize the improved loss proposed by Pix2Pix \[21\] in which the discriminator does not observe \( x \) and an \( l_1 \) distance is mixed with the GAN objective to encourage less blurring:

\[
\mathcal{L}_{\text{cGAN}}(G, D) = E_{y \sim p_{\text{data}}(y)}[\log D(x, y)] + E_{x \sim p_{\text{data}}(x), z \sim p_z(z)}[\log(1 - D(x, G(x, y)))] , \quad \mathcal{L}_{L1}(G) = E_{x,y,z}|||y - G(x, z)|||_1.
\]

The auxiliary sketch recognition loss maximizes the log-likelihood between predicted and ground-truth labels:

\[
\mathcal{L}_{\text{ac}}(C) = E[\log P(C = c|y)]
\]

and the generator maximizes the same log-likelihood \( \mathcal{L}_{\text{ac}}(G) \) with the discriminator and classifier fixed.

Our total loss function is defined as follows:

\[
G^* \in \arg \min \max_{D} \mathcal{L}_{\text{cGAN}}(G, D) + \lambda_1 \mathcal{L}_{L1}(G) + \lambda_2 \mathcal{L}_{\text{ac}}(G),
\]

where \( \lambda_1 \) and \( \lambda_2 \) are weights to balance different loss terms. We empirically set \( \lambda_1 = 100 \) and \( \lambda_2 = 0.5 \), which give good performance. In addition, we make our final network learn a mapping from \( x \) to \( y \) without \( z \), which is consistent with Pix2Pix \[21\]. We only apply noise on several layers of the generator in the form of dropout.

3.2. Cascade Strategy

We propose a cascade module to further refine the contour closure of the completed sketches and boost the baseline performance, which can be trained in a multi-stage manner and keeps a good balance between accuracy and efficiency. Our method uses sequential iterations for refinement, starting from an initial model output, and then refining the output by iteratively using original input and all the outputs of the previous stages.

Each cascade stage is built upon a conditional GAN network, but does not share network parameters for different cascade stages. More specifically, the output of the first cascade stage \( y_1 \) is fed to the next cascade stage, together with the original corrupted sketch image \( x \). The output of the second cascade stage \( y_2 \) is fed to the third cascade stage, together with the original corrupted sketch image \( x \) and the output of the first cascade stage \( y_1 \). Experiments of the proposed network on the Sketchy database \[44\] over the number of cascade stages have been conducted, and the performance of the network is almost reaching a steady state after three cascade iterations. This cascade network can be regarded as a recurrent completion process since both the original input \( x \) and the intermediate outputs are repeatedly fed to the generator. The outputs are not fed to the discriminator or the classification network until the last cascade stage.

3.3. Sketch Recognition

We adopt a sketch recognition network \( C \) as an auxiliary network for sketch completion. The architecture for \( C \) is shown within the top-right box in Figure 2. The motivation to use sketch recognition as an auxiliary task is as follows. On the one hand, completed sketches help improve sketch recognition performance. Although sketch recognition has been widely studied, the problem of corrupted sketch recognition is hardly researched. As we will show later in Section 4, the output of our sketch completion method improves the performance of existing sketch recognition algorithms. On the other hand, since correctly completed sketches are easier to classify, this in turn promotes better sketch completion. Experiments show that the completion results with the auxiliary sketch recognition task are better than those without the sketch recognition task.

3.4. Network Architecture

**Generator.** Following the generator architecture from those in \[21\], skip connections between mirrored layers in the encoder and decoder stacks have been added to the generator, following the general shape of a “U-Net” \[43\]. Our inputs are the fused information of the original input and the outputs of the previous stages. The inputs are fed to a pre-processing network consisting of two convolutional layers. We use eight up-convolutional layers with kernel size 4 and stride 2. A batch normalization layer and a rectified linear unit (ReLU) activation is used in all layers. Instead of a sigmoid activation used in the output layer in Pix2Pix, we use a \( \tanh \) activation and then map it to \([0, 1]\) in the output.

**Discriminator.** Inspired by the work \[20\], we use two components in the discriminator network. The local discriminator tells the fake sketch image from the real one only concentrating on the completed local regions, while the global discriminator focuses on the coherence of the entire image and the contour closure effect. Outputs of the networks are fused together by a concatenation layer, followed by a fully connected layer and activated by a ReLU function. Finally, we use a \( \tanh \) activation and then map the result to \([0, 1]\) in the output, in which 0 means fake and 1 means real.

**Classifier.** We have adopted Sketch-a-Net \[53\] as our
sketch recognition auxiliary network $C$. Sketch-a-Net is a state-of-the-art CNN model for freehand sketch recognition. The network architecture of $C$ is shown in the top-right box in Figure 2. The sketch recognition auxiliary network is trained using a cross-entropy loss in an end-to-end manner along with the rest of SketchGAN.

4. Experiments and Results

4.1. Dataset and Evaluation Metrics

Dataset. The Sketchy database (Sketchy) [44] is the first large-scale collection of sketch-photo pairs. Particular photographic objects sampled from 125 categories have been sketched by crowd workers and totally 75,471 sketches of 12,500 objects have been acquired. This database is widely used in fine-grained sketch-related applications.

Data Preprocessing. We conduct data augmentation on the Sketchy database by making randomly corrupted sketches for each category. The dataset is randomly split into training and testing sets, containing 80% and 20% sketches, respectively. For each sketch, we generate a blank rectangle mask of random size and random position to erase the original sketch and get a corrupted sketch. This corrupted sketch is saved in the database if the difference between the original sketch and get a corrupted sketch is within $10\%$ ~ $40\%$, to simulate typical missing content on sketches. We generate one corrupted sketch for each sketch and finally obtain about 500 corrupted sketches for each category.

Our sketch completion network design is flexible to cope with arbitrary number of sketch categories. We perform ablation experiments and comparisons with the state-of-the-art methods on 11 sketch sub-categories (car, (sedan), cow, horse, cat, dog, sheep, airplane, motorcycle, bicycle, songbird, pickup_truck) of Sketchy [44]. These sub-categories are consistent with those in the SKETCHPARSE network [45] designed for general sketch parsing work. Moreover, we also conduct sketch completion on larger numbers of sketch categories, and experiments show that our model has stable performance even with all the 125 categories from the Sketchy database.

Evaluation Metrics. To begin with, we adopt a straightforward pixel-to-pixel comparison method. We only compute the metrics within the corrupt mask. Four widely used metrics, i.e., the precision of completed black pixels Precision, the recall of completed black pixels Recall, the F-measure F-Measure, and the accuracy Accuracy, are computed. This pixel-to-pixel comparison method is not perfect for measuring the performance of sketch completion due to the inevitable variation of appropriate sketches. Ultimately, the goal of sketch completion is to make the generated sketches plausible to humans. Generated sketches that do not match the original sketches used to make them, have relatively low Precision or Recall, but they may still be good (see Figure 3). Thus, we also visually inspect the results and conduct two user studies (refer to Section 4.5).

4.2. Sketch Completion Results

Comparative Results. In this section, we show sketch completion results of the mentioned 11 object categories. Table 1 shows the quantitative performance of different categories in the Sketchy database. Each category contains about 550 sketches, and we make one corrupted sketch for each original sketch. Thus, the total numbers of training and testing corrupted sketches for all the 11 categories are 6,196 and 1,184, respectively.

Table 1. Results on the Sketchy database. We conduct the experiment using the 11 categories simultaneously, and report the performance on individual categories. Experiments show that our method is effective in sketch completion tasks of various sketch categories, leading to stable performance.

<table>
<thead>
<tr>
<th>Sketch Category</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F-Measure (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cow</td>
<td>76.74</td>
<td>49.77</td>
<td>60.38</td>
<td>92.30</td>
</tr>
<tr>
<td>horse</td>
<td>72.77</td>
<td>47.22</td>
<td>57.27</td>
<td>90.61</td>
</tr>
<tr>
<td>cat</td>
<td>75.56</td>
<td>48.23</td>
<td>58.87</td>
<td>91.25</td>
</tr>
<tr>
<td>dog</td>
<td>73.24</td>
<td>44.93</td>
<td>55.89</td>
<td>91.26</td>
</tr>
<tr>
<td>sheep</td>
<td>77.75</td>
<td>50.23</td>
<td>61.28</td>
<td>92.35</td>
</tr>
<tr>
<td>pickup_truck</td>
<td>74.86</td>
<td>51.64</td>
<td>61.12</td>
<td>89.82</td>
</tr>
<tr>
<td>car</td>
<td>75.33</td>
<td>51.90</td>
<td>61.46</td>
<td>91.45</td>
</tr>
<tr>
<td>bicycle</td>
<td>76.47</td>
<td>60.47</td>
<td>67.53</td>
<td>88.32</td>
</tr>
<tr>
<td>motorcycle</td>
<td>79.56</td>
<td>61.68</td>
<td>69.49</td>
<td>89.87</td>
</tr>
<tr>
<td>airplane</td>
<td>77.41</td>
<td>55.85</td>
<td>64.88</td>
<td>92.85</td>
</tr>
<tr>
<td>songbird</td>
<td>70.75</td>
<td>48.32</td>
<td>57.42</td>
<td>93.00</td>
</tr>
</tbody>
</table>

Ablation study. Compared to conventional photo-oriented DNNs such as Pix2Pix and DCGANs, our SketchGAN has two distinct features (see Section 3): (i) the cascade network architecture, (ii) the strategy of utilizing the auxiliary sketch...
recognition network. In this section we evaluate the contributions of each feature. Specifically, we examine three ablative versions of our full model: SketchGAN with all features but no cascade (No Cascade), SketchGAN with all features but no auxiliary sketch recognition network (No Classifier), i.e. there are no class labels and the data of 11 categories are trained and tested together, and SketchGAN with all features. We show the sketch completion results in Figure 4 for visual analysis. And the results in Table 2 show that both strategies contribute to the final strong performance of SketchGAN. In particular, (i) from the visual results, more broken lines and intermediate completed sketches are likely to be produced without the proposed cascade strategy, and the cascade strategy leads to better completion results on large corrupted regions. (ii) The auxiliary sketch recognition network improves the sketch completion greatly when multi-class objects exist by offering strong category priors. Experiments demonstrate that our sketch completion method can achieve reliable sketch completion results. The cascade strategy helps to make the completed sketches maintain contour closure, and the completion results tend to be distorted without the sketch recognition auxiliary network.

Since the network directly outputs the inpainted sketches instead of filling holes, the output sketches may be different from the input on non-missing regions. We conduct evaluations on non-missing regions. Experiments of our SketchGAN show that non-missing regions are well preserved, with 93.46% precision and 85.72% recall.

**Network performance analysis.** We conduct experiments to evaluate the performance with respect to different incomplete ratios of corrupted sketches. In this experiment, we conduct a series of comparisons under the same conditions but with different incomplete ratios. Table 3 shows that our algorithm has good sketch completion performance under different incomplete levels. If the incomplete ratio is up to 40%, our sketch completion method performs well. When the incomplete ratio is above 40%, which is quite extreme in practical applications, the performance of our model drops.

Moreover, we also conduct experiments to evaluate the performance with respect to different numbers of sketch categories. In this experiment, we conduct a series of comparisons under the same conditions but with gradually increased sketch categories. Table 4 shows our algorithm has good sketch completion performance under different numbers of sketch categories, and our method behaves well even for all the 125 sketch categories in the Sketchy Database. Therefore, our method also has good generalization power.

**4.3. Comparison with state-of-the-art methods**

We compare our method with state-of-the-art methods: Pix2Pix [21], DCGAN [40], GlobalLocalImageCompletion [20], Generative Image Inpainting [52]. Pix2Pix and
Table 5. Comparison with state-of-the-art methods. Our method achieves better results than state-of-the-art methods: Pix2Pix [21], DCGAN [40], Globally and Locally Consistent Image Completion [20], and Generative Image Inpainting [52].

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F-Measure (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>75.62</td>
<td>52.07</td>
<td>61.67</td>
<td>91.09</td>
</tr>
<tr>
<td>Pix2Pix [21]</td>
<td>33.17</td>
<td>11.99</td>
<td>17.61</td>
<td>85.52</td>
</tr>
<tr>
<td>DCGAN [40]</td>
<td>14.96</td>
<td>24.27</td>
<td>18.51</td>
<td>74.34</td>
</tr>
<tr>
<td>Globally and Locally Consistent Image Completion [20]</td>
<td>22.49</td>
<td>18.25</td>
<td>78.48</td>
<td>20.15</td>
</tr>
<tr>
<td>Generative Image Inpainting [52]</td>
<td>64.25</td>
<td>11.41</td>
<td>19.38</td>
<td>89.17</td>
</tr>
</tbody>
</table>

DCGAN are GAN-based methods, and Globally and Locally Consistent Image Completion and Generative Image Inpainting are image inpainting methods.

Table 5 shows the quantitative evaluation of our method and the state-of-the-art methods. Figure 5 shows examples for visual comparison. Our network is superior to the baselines in sketch completion in terms of completion performance and visual results. Compared with Pix2Pix [21], our method produces better results in terms of contour connectedness. DCGAN [40] usually produces blurry sketches. For the two state-of-the-art inpainting methods designed for images, they often produce wrong strokes.

4.4. Applications

Incomplete Sketch Recognition. We set our sketch completion as an effective intermediate step to solve the problem of incomplete or corrupted sketch recognition. Sketch completion is conducted before sketch recognition. As mentioned in Section 1, Figure 1 shows the incomplete sketch recognition process with and without our sketch completion as an intermediate step.

Moreover, we have found that the improvement of sketch recognition rate is largely influenced by the incomplete ratio of the corrupted data. When the incomplete ratio is relatively low, the current sketch recognition methods can cope with such input data quite robustly, but these methods cannot maintain the same level of performance as the incomplete ratio increases. In such cases, our sketch completion helps improve the recognition performance significantly. Here we conduct an incomplete sketch recognition experiment using two well-known sketch recognition methods: Sketch-a-net [53] and Sketch-object-recognition [46]. Figure 6 shows the sketch recognition performance w.r.t. varying incomplete ratio of sketches, indicating that our incomplete sketch recognition approach is universal for im-
proving general sketch recognition methods. Using our sketch completion method as an intermediate step of the incomplete sketch recognition task can enhance the recognition accuracy significantly, especially for sketches corrupted seriously.

**Figure 6.** Improvement of sketch recognition rates with respect to the incomplete ratio of sketches. The sketch recognition methods used here are Sketch-a-net [53] and Sketch-object-recognition [46], respectively.

**Sketch Editing.** Hand-drawn sketches are usually incomplete in scenarios such as overlapping sketches for a scene with multiple objects, interim sketches, or corrupted sketches due to image segmentation. Neither the well-known sketch2photo problem [7] nor the current sketch recognition studies have considered or handled the situations where corrupted sketches are provided, which brings huge obstacles and limitations to post sketching applications. Here we use our sketch completion method as an intermediate step which offers a new mode of sketch applications (See Figure 7).

**Figure 7.** Sketch editing. Corrupted sketches appear when parsing or segmenting sketch scenes. With sketch completion as a key step, we can handle post sketching applications more easily, including sketch reorganization, colorization, SBIR, sketch based image generation, etc.

### 4.5. User Study

Since pixel level comparisons do not always give meaningful measure of sketch completion quality, we further design user studies to evaluate our sketch completion method. **User Study I.** We ask 15 users (8 males and 7 females) to evaluate the naturalness of our sketch completion, since there do not exist standardized criteria to evaluate sketch quality [9]. The mean age of the participants is 24.5 years. Before the experiment, we explained the purpose and procedures to the participants. Each experiment lasted about 15 minutes. In our test, users are shown with randomly selected sketches, either original sketches from the dataset or corrupted sketches completed by our method, and they are asked to evaluate whether the sketch is an actual hand-drawn sketch or a completed one. The results show that 38% of our sketch completion results fool the people on average, and 56% of hand-drawn sketches are regarded as hand-drawn. This demonstrates that our completed sketches are perceptually close to real hand-drawn sketches from a user perspective.

**User Study II.** Next, we analyze subjective quality judgment of the completed sketches processed by our method and by other image completion methods. We adopt a two alternative force choice (2AFC) scheme which is widely used in psychological studies as it is simple and reliable. 20 participants are asked to pick a better one between the results of our method and those of Pix2Pix or GlobalLocalImageCompletion. Images are shown in random order to avoid bias. The results show that more than 76% and 84% users believe the completion results of our method are better than those completed by Pix2Pix and GlobalLocalImageCompletion, respectively, proving that our method is superior to state-of-the-art methods from a user perspective.

### 5. Conclusions

In this paper, a novel solution to joint sketch completion and recognition is proposed, and a completion model fusing cascade network and conditional GANs is used to capture the characteristics of sketches. The experiments demonstrate that the proposed model performs very well in various sketch completion tasks, and also demonstrate that sketch completion is useful in various sketch-based applications. Although promising performance has been achieved, there are still many issues open to be addressed in future work, such as coping with different styles of sketches and multiple objects at the same time, etc. In addition, the missing areas of sketches are limited to 50% in our current experiment settings. However, users may draw very rough sketches with fewer and inaccurate strokes on touch screen devices. We will extend the framework to complete sketches with large missing ratio in future.

**Acknowledgments:** This work was supported by the National Key Research and Development Plan (2016YFB1001200), Natural Science Foundation of China (61872346, 61725204, 61473276), Natural Science Foundation of Beijing, China (L182052), and Royal Society-Newton Advanced Fellowship (NA150431).
References


