

AniFaceDrawing: Anime Portrait Exploration during Your Sketching

ZHENGYU HUANG, Japan Advanced Institute of Science and Technology, Japan

HAORAN XIE, Japan Advanced Institute of Science and Technology, Japan

TSUKASA FUKUSATO, Waseda University, Japan

KAZUNORI MIYATA, Japan Advanced Institute of Science and Technology, Japan

In this paper, we focus on how artificial intelligence (AI) can be used to assist users in the creation of anime portraits, that is, converting rough sketches into anime portraits during their sketching process. The input is a sequence of incomplete freehand sketches that are gradually refined stroke by stroke, while the output is a sequence of high-quality anime portraits that correspond to the input sketches as guidance. Although recent GANs can generate high quality images, it is a challenging problem to maintain the high quality of generated images from sketches with a low degree of completion due to ill-posed problems in conditional image generation. Even with the latest sketch-to-image (S2I) technology, it is still difficult to create high-quality images from incomplete rough sketches for anime portraits since anime style tend to be more abstract than in realistic style. To address this issue, we adopt a latent space exploration of StyleGAN with a two-stage training strategy. We consider the input strokes of a freehand sketch to correspond to edge information-related attributes in the latent structural code of StyleGAN, and term the matching between strokes and these attributes “stroke-level disentanglement.” In the first stage, we trained an image encoder with the pre-trained StyleGAN model as a teacher encoder. In the second stage, we simulated the drawing process of the generated images without any additional data (labels) and trained the sketch encoder for incomplete progressive sketches to generate high-quality portrait images with feature alignment to the disentangled representations in the teacher encoder. We verified the proposed progressive S2I system with both qualitative and quantitative evaluations and achieved high-quality anime portraits from incomplete progressive sketches. Our user study proved its effectiveness in art creation assistance for the anime style.

CCS Concepts: • **Computing methodologies** → **Graphics systems and interfaces**; **Image manipulation**; **Machine learning**.

Additional Key Words and Phrases: Stroke-level Disentanglement, StyleGAN, Anime Portrait, Disentanglement Learning, Freehand Sketching

ACM Reference Format:

Zhengyu Huang, Haoran Xie, Tsukasa Fukusato, and Kazunori Miyata. 2023. AniFaceDrawing: Anime Portrait Exploration during Your Sketching . 1, 1 (June 2023), 11 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Using AI to assist the general user in creating a professional anime portrait is not a trivial task. As a popular drawing style, the anime

style is based on realism but has its own characteristics and exaggerations – line drawings of anime portraits are simpler and more abstract than real human faces. Furthermore, the input sketches that users make during the drawing process contain little detailed information and lack partial shape information. As a result, the high-quality synthesis of anime portraits from freehand sketches is challenging. The main problem is how to generate appropriate guidance images that match these abstract lines, based on a sequence of incomplete sketches, for the user during freehand sketching.

We also realized that classic S2I or style transfer techniques would not work for this task. Consider a scenario in which a novice tries to create an anime portrait with an ideal AI-assisted system. As the user draws stroke by stroke, the guidance image generated by this system should be able to locally match the sketch as the number of strokes increases. However, most S2I approaches tend to consider only complete sketches as input for image generation – in the case of incomplete sketches, especially those with sparser strokes, they cannot maintain the quality of the output. Taking sketch-to-anime-portrait generation with StyleGAN as an example, in Figure 1(a), the state-of-the-art Pixel2Style2Pixel (pSp) [Richardson et al. 2021] is an encoder for GAN (Generative Adversarial Network) inversion that can successfully reconstruct a complete line drawing into an anime portrait that can tolerate small missing areas (first row), but it gets poor outputs when the input is a sketch with large missing areas (second row) or a rough sketch with less details (third row). Therefore, conventional GAN inversion techniques perform poorly in the drawing process – they do not spontaneously implement stroke-level disentanglement during learning, nor do they naturally maintain partial matches. Similarly, the simple combination of a rough sketch-real-face approach with a further style transfer from realistic to anime style may not be a good idea. Figure 1(b) shows such an example: A sketch-real face pSp encoder is combined with a face-anime style transfer called DualStyleGAN [2022] for sketch-based anime portrait generation, using real faces as the intermediate results. Because the difference between the anime-style face and the real face is relatively large, results generated by this method are not consistent with the input sketches. For the same reason, DualStyleGAN itself tends to fall into “mode collapse,” no matter if the input is a sketch of a realistic style or an anime style.

Thus, we propose a new idea for anime-style portrait generation during sketching. Our solution involves sketch-based latent space exploration in a pre-trained StyleGAN [Karras et al. 2019]. The advent of StyleGAN made it possible to create high-quality images for many types of subjects, including anime portraits. In turn, this great success led to the rapid development of the GAN control for image editing. By applying linear regression to the disentangled latent space of a StyleGAN, users can control the various properties of the generated image by modifying the attribute parameters. In our case,

Authors' addresses: Zhengyu Huang, Japan Advanced Institute of Science and Technology, Nomi, Japan, huang.zhengyu@jaist.ac.jp; Haoran Xie, Japan Advanced Institute of Science and Technology, Nomi, Japan, xie@jaist.ac.jp; Tsukasa Fukusato, Waseda University, Tokyo, Japan, tsukasafukusato@waseda.jp; Kazunori Miyata, Japan Advanced Institute of Science and Technology, Nomi, Japan, miyata@jaist.ac.jp.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2023 Copyright held by the owner/author(s).

XXXX-XXXX/2023/6-ART

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>



Fig. 1. Comparison of different ideas for sketch-based anime portrait generation. In (a), an original pSp encoder, which works for line drawing with small areas missing (first row), cannot correctly recognize user sketches, even for a complete sketch (third row). In (b), sketch-anime portrait style transfer using real faces as intermediate results lead to an input/output mismatch and mode collapse. And lack of diversity in the results of different inputs fails to assist the user in drawing anime portraits. Realistic sketches (first three rows) are used to obtain better intermediate faces. In contrast, our method can generate more diverse anime portraits, even if the input is a realistic sketch after translation (first two rows) or flip (third row). (c) shows that our method can generate high-quality results that consistently match the input sketch throughout the sketching process. To make the matching of sketches and results of our method clear, the intermediate results disentangled most of the color information (second row) are stacked below the input strokes (blue strokes on the first row) once a new stroke (red) is added. The final results (third row) were generated using a random style-mixing technique. Note that all generated results with “near-white” hair are intermediate results, which are style mixing with a fixed “near-white” color latent code.

we are trying to implement sketch-based anime portrait generation control during the drawing process, because manipulating multiple shape-related attributes (e.g., pose, mouth shape, or nose position) with separate sliders is not intuitive enough. Compared with the above-mentioned approaches, our approach allows the generated results to be matched with the users’ rough sketches during their drawing process (see Figure 1(c)). To the best of our knowledge, our system is the first to provide anime portraits progressive drawing assistance. Our main contributions are summarized as follows:

- We present AniFaceDrawing, the first high-quality anime portrait drawing assistance system based on the S2I framework from freehand sketches throughout the entire drawing process.
- We propose an unsupervised stroke-level disentanglement training strategy for StyleGAN, so that rough sketches with sparse strokes can be automatically matched to the corresponding local parts in anime portraits without any semantic labels.
- A user study is conducted to prove the effectiveness and usability of AniFaceDrawing for users when creating anime portraits.

2 RELATED WORK

Latent space of StyleGAN. With the further development of GAN, how to use the latent space to manipulate outputs from pre-trained GANs has also become a hot research topic [Xia et al. 2023]. Among various pre-trained GANs, StyleGAN [Karras et al. 2021] is usually the most common choice. A typical StyleGAN generator usually involves three types of latent spaces: \mathcal{Z} , \mathcal{W} , and $\mathcal{W}+$. A random vector $z \in \mathcal{Z}$ is often a white noise belonging to a Gaussian distribution, which is the same as the original GAN. In StyleGAN, the z vector first passes through a mapping network, which is composed of eight fully-connected layers and is transformed to w embedding into an intermediate latent space \mathcal{W} . Note that both z and w are 512-dimensional vectors. Here, the introduction of this mapping network is to get rid of the influence of the input vector z by the

distribution of the input data set and to better disentangle the attributes. Each layer of the StyleGAN generator can receive a vector w of input via AdaIN (adaptive instance normalization). As there are 18 such layers in the StyleGAN generator, StyleGAN can input up to 18 mutually different w vectors. This different w can be concatenated into a new vector $w+$ with 18×512 dimensions and the corresponding latent space to $w+$ is called $\mathcal{W}+$. One application of $w+$ is style mixing, which can also be found in the inference step in Figure 4. In addition, we mapped incomplete progressive sketches into the latent space $\mathcal{W}+$ of StyleGAN for guidance generation.

Facial latent space manipulation. One of the most important applications of latent space manipulation is face attribute editing. Chiu et al. [2020] present a human-in-the-loop differential subspace search for exploring the high-dimensional latent space of GAN by letting the user perform searches in 1D subspaces. Härkönen et al. [2020] identify latent directions with principal components analysis (PCA), and created interpretable controls for image synthesis, such as viewpoint changing, lighting, and aging. By determining facial semantic boundaries with a trained linear SVM (support vector machine), Shen et al. [2022] is able to control the expression and pose of faces. An instance-aware latent-space search (IALS) is performed to find semantic directions for disentangled attribute editing [Han et al. 2021]. Instead of the tedious 1D adjustment of each face attribute, we directly use progressive rough sketches to control the shape attributes of the face and explore latent space.

GAN-control with encoder-based manipulation. The pSp encoder implements GAN inversion without optimization by using feature pyramids and mapping networks. Since this method does not need to compute losses between inputs and outputs of a GAN, it can also handle semantic layouts or line drawings as input. To improve the editability of the encoder-based approach, Tov et al. [2021] introduced regularization and adversarial losses for latent codes into encoder training. In addition, the ReStyle encoder [Alaluf et al. 2021]

has improved the reconstruction quality of inverted images by iteratively refining latent codes from the encoder. Unlike encoder-based approaches, which require many training pairs, our method automatically generates sketches directly from the GAN on the fly in the training step without additional pairwise data generation.

Interactive AI assistance. With the rapid development of deep learning, many efforts have been made to apply AI to interactively assist users in various fields, such as music creation [Frid et al. 2020], handwritten text editing [Aksan et al. 2018], and sketch colorization [Ren et al. 2020]. When it comes to sketch-based drawing assistance, the dominant idea has remained to adopt retrieval-based approaches [Choi et al. 2019; Collomosse et al. 2019] since the ShadowDraw [2011] have been proposed. For example, to improve users' final sketches, DeepFaceDrawing [2020] and DrawingInStyles [2022] adopts a shadow guidance which retrieves sketches from a database rather than using the generated images directly.

Sketch-based applications. As a high-level abstract representation, sketches can be used as conditional inputs to generative models. Sketch-based systems allow users to intuitively obtain results in various applications, such as image retrieval [Lee et al. 2011; Liu et al. 2017; Yu et al. 2016] and image manipulation [Dekel et al. 2018; Portenier et al. 2018; Tseng et al. 2020; Yang et al. 2020], simulation control [Hu et al. 2019], block arrangement [Peng et al. 2020], and 3D modeling [Fukusato et al. 2020; Igarashi and Hughes 2001; Igarashi et al. 1997]. As for applications such as iSketchNFill [2019], which consider the sketching process as input, the generation quality is still limited and cannot be applied to high-quality anime portrait generation. Although there have also been attempts to generate high-quality faces with sketches (e.g., DeepFacePencil [2020]), they do not take into account the case of sparse input sketches at the beginning of drawing process. On the other hand, the vast majority of S2I studies [Chen et al. 2020; Yang et al. 2021] target the generation of real images, but how these methods can be applied to the abstract artistic style for art drawing assistance has not been explored, which is the topic in this paper.

3 STROKE-LEVEL DISENTANGLEMENT

The use of sketches to control shape properties of an anime face to achieve latent space exploration is called stroke-level disentanglement. We first explain its concept with a simple example (Figure 2). Given an image generated by StyleGAN with a fixed color latent code, the left/right eyes (green box) are mapped to L and R in the disentangled latent space with GAN inversion. Stroke-level disentanglement means that there is a sketch-GAN inversion coder for the rough sketch that allows Stroke 1 and 2 (red box) to be mapped to the subset of the corresponding latent codes L and R , respectively. Note that the percentage of the latent code (intersection of the same structural information \div latent codes for a single facial part) of Stroke 1 to L is higher than that of the latent code of Strokes 2 to R , because Stroke 1 contains more details. In addition, there may be a one-to-many relationship between the strokes and the latent code of different facial parts; for example, if a stroke contains shape information of both the left and right eyes at the same time, it will correspond to a subset of both L and R after encoding.

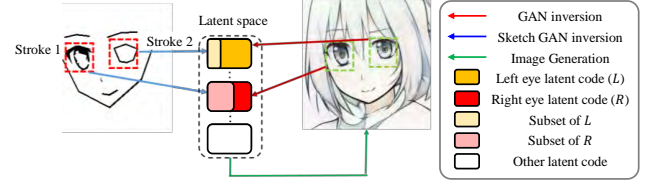


Fig. 2. Illustrating the stroke-level disentanglement.

We formally describe the problem as follows. Let P and S denote the anime portrait domain and the sketch domain, respectively. Q is a subset of P that separates most representations of color information from structural information and can form a one-to-one mapping with S . Our sketch encoder learns a mapping $F : S \rightarrow Q$ which can find the correct correspondence with increased drawing strokes. This mapping F is called “sketch GAN inversion” in this paper. The output during the drawing process should gradually converge and maintain high quality as the input strokes increase. Two main research questions need to be addressed:

- Q1. How does one learn a stroke-level disentangled mapping F that allows strokes to locally match to the generated image?
- Q2. How can the aforementioned mapping not be affected by the stroke order?

Given a sketch consisting of a series of strokes $\{s_1, s_2, \dots, s_n\}$, these two questions require that mapping F in ideal cases satisfies the following two conditions.

Stroke independence. Assume that an image encoder that converts an anime portrait to completely disentangled structural latent codes $\{d_1, d_2, \dots, d_n\}$ corresponding to strokes one by one, then:

$$F(s_i) = d_i \quad (1)$$

where i is the index of strokes ($i \leq n$). Note that each stroke can create a new partial sketch $\{s_i\}$ that contains only one facial part.

Stroke order invariance. For any different stroke index $i, j \leq n$:

$$\begin{aligned} F(s_i | s_1, s_2, \dots, s_{i-1}, s_{i+1}, \dots, s_n) \\ = F(s_j | s_1, s_2, \dots, s_{j-1}, s_{j+1}, \dots, s_n) \\ = F(s_1, s_2, \dots, s_n) \end{aligned} \quad (2)$$

where $s_i | s_1, s_2, \dots, s_{i-1}, s_{i+1}, \dots, s_n$ means add stroke s_i to a sketch consisting of strokes $\{s_1, s_2, \dots, s_{i-1}, s_{i+1}, \dots, s_n\}$. Note that we do not use any semantic label, and the inputs are monochrome sketches.

Figure 3 shows our core idea of simulating the drawing process and making the sketch with a higher degree of completion, closer to the original sketch, in the latent space limited to a neighboring region that can provide the answers to the above two questions. Given an image generated by StyleGAN, the point computed by GAN inversion in the latent space P is p , and the point in the image fixed color latent code (first row in the gray dashed box) projected into the latent subspace Q is q . Our drawing process simulation generates a sequence of simulated sketches (second row in the gray dashed box) from simple to complex, whose positions in Q space are denoted as S_1 to S_n . The core idea is to learn a spatial neighborhood in P whose projection in subspace Q can make the sequence of points S_1 to S_n gradually approximate the point q .

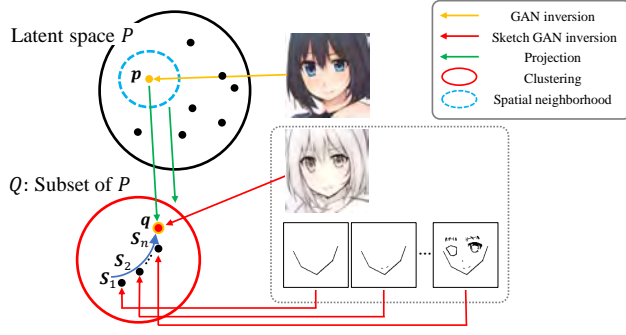


Fig. 3. Illustrating the drawing process in the latent space.

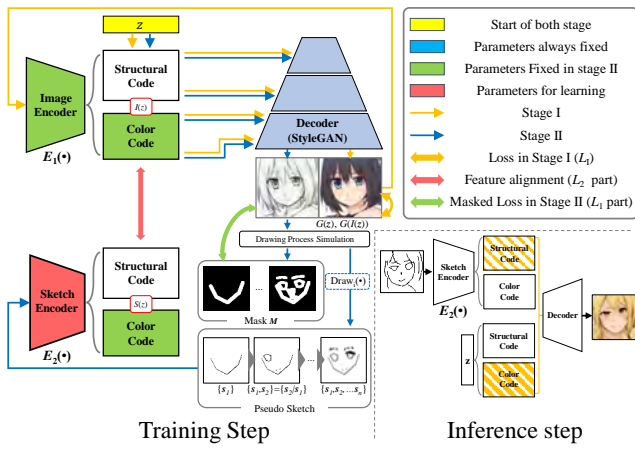


Fig. 4. Framework of the proposed method. The training step consists of two stages: (Stage I) an image encoder for GAN inversion and (Stage II) a sketch encoder for sketch GAN inversion where the image encoder works as a teacher.

4 PROPOSED FRAMEWORK

An overview of the framework is shown in Figure 4. In the training step, we first trained an image encoder using the randomly-generated images from the decoder, which correctly projected the anime portraits back into the latent space (Stage I). Then, we rearranged the latent space vectors in this image encoder by simulating the drawing process, so that sketches with similar strokes retained more rational distribution when projected into Q (Stage II). In the inference step, we concatenated the structural codes derived from the sketch encoder with the color codes from the random Gaussian noise z , which is known as style-mixing. Note that once the decoder is determined, all data are derived from the randomly-generated images of that decoder, and no additional auxiliary database is required. Thus, this is an unsupervised learning approach.

The training in Stage I is similar to previous work [Richardson et al. 2021]. The difference is that we simply adopted the $L2$ loss between the original images from StyleGAN and the reconstructed images encoded by our image encoder.

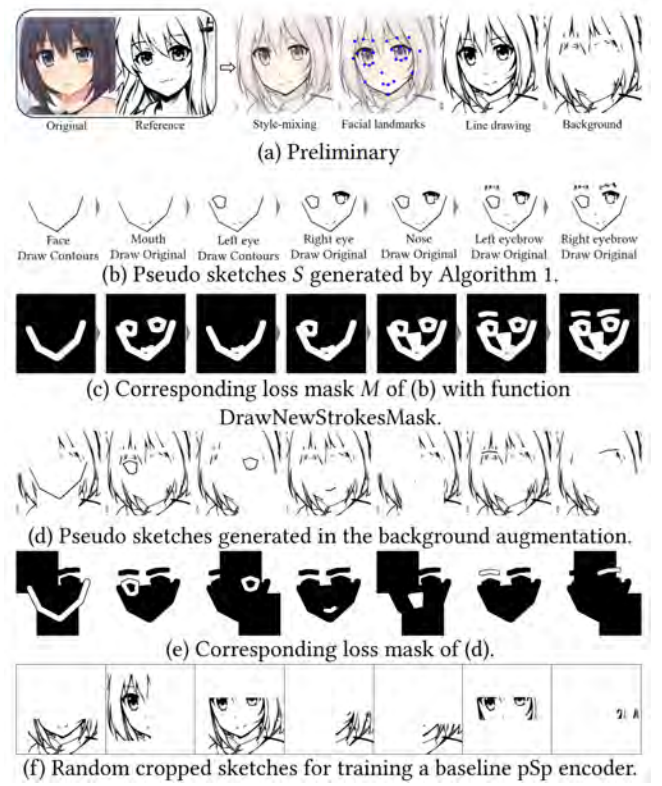


Fig. 5. An example of (b, c) the drawing process simulation and (d, e) background augmentation after (a) a series of pre-processing for a randomly generated image (original) from StyleGAN. The corresponding choices of *RandomSelectOneStroke* and *RandomDrawing* functions for each stroke are shown in (b). In (c), *DrawNewStrokesMask*(m_t, s) executes *DrawContours* on the previous loss mask m_t for each stroke s to get a new loss mask, except for the nose (the nose area is determined by the landmarks in the center of eyes and the corners of mouth). For comparison, we also trained a baseline encoder with a random cropping strategy from the input line drawing in (f).

4.1 Drawing Process Simulation

In Stage II, the drawing process simulation automatically generates sketch-image pairs from StyleGAN. Before the drawing process simulation, we should first get a complete line drawing of the original anime portrait generated by StyleGAN as the simulation input. As shown in the first line of Figure 5, we conducted a style-mixing between the original and the reference image so that most of the color information could be removed and obtained a complete line drawing from the style-mixing result with xDoG [2011]. We then used a landmark detection technique for anime faces [Hysts 2021] to acquire information about the contours of each face part. We simulated the intermediate results of the sketch process stroke by stroke using Algorithm 1. Facial landmarks are divided into n types ($n = 7$), and for each new stroke in k iterations, a facial part is randomly selected as a part stroke $s \in C$ with *RandomProcess* and *RandomDrawing* as Figure 5(b) shown. In the same iteration,

the corresponding cumulative loss mask is drawn by the function DrawNewStrokesMask (Figure 5(c)).

Background augmentation. Since the hair and other parts could not be extracted by the anime face detection algorithm, we treated them as background. To increase stability, random cropping of the background image and random selection from facial contours were combined as augmentation data. The effects of this method are discussed in Section 6.2. At this point, we had a series of pseudo sketches for training in Stage II which is called “feature alignment.”

ALGORITHM 1: Drawing process simulation

Input: Portrait image P
Output: List of pseudo sketch S , List of loss mask M
Landmarks of portrait $L \leftarrow \text{FaceDetect}(P)$;
Strokes of facial parts $C \leftarrow \text{Resort}(L)$;
Number of $n \leftarrow \text{len}(C)$;
Temporary white image $p_t \leftarrow \text{ones}(P.\text{shape}) \times 255$;
Temporary loss mask $m_t \leftarrow \text{zeros}(P.\text{shape})$;
RandomProcess=[GaussianBlur(kernel size= 3×3); Dilate; Erode, KeepOriginal(None)];
RandomDrawing=[DrawOriginal, DrawContours];
 $S \leftarrow \emptyset$;
 $M \leftarrow \emptyset$;
for $k=1:n$ **do**
 Index $i = \text{RandomSelectOneStroke}(C)$;
 Part stroke $s=C.\text{pop}(i)$;
 $p_t = \text{RandomDrawing}(\text{RandomProcess}(p_t, s))$;
 $S.\text{push}(p_t)$;
 $m_t = \text{DrawNewStrokesMask}(m_t, s)$;
 $M.\text{push}(m_t)$;
end
return S, M

4.2 Feature Alignment

Given Gaussian noise z , the input image of our encoder is $x = G(z)$ and the output latent code $I(z)$, a special implementation of a point p in P (Figure 3), is then defined as:

$$I(z) := E_1(G(z)) \quad (3)$$

where $E_1(\cdot)$ and $G(\cdot)$ denote the image encoder and StyleGAN generator, respectively. Then, our method for training an image encoder in Stage I followed the usual GAN inversion method. The loss function L_I we used in Stage I is as follows:

$$L_I = L_2(G(I(z)), G(z)) \quad (4)$$

Just by calculating the L_2 distance between the input image and the reconstructed image, the image encoder can already learn inverse mapping very well. Similarly, we defined the output latent code of our sketch encoder as follows:

$$S(z) := E_2(\text{Draw}_i(G(z))) \quad (5)$$

where $E_2(\cdot)$ and $\text{Draw}_i(\cdot)$ denote our sketch encoder and our drawing process simulation as described in Algorithm 1, which can convert the image x to a series of intermediate sketches of the drawing process and select the i -th sketch from among them.

In each iteration of training in Stage II, we can generate sketches S and corresponding loss masks M after our drawing process simulation. Then, the loss function is:

$$L_S = L_1(G(S(z)) * M, G(z) * M) + L_2(I(z), S(z)) \quad (6)$$

$L_2(I(z), S(z))$ ensures that the sketch with a higher degree of completion is closer to the projection of the original in the latent subspace, while $L_1(G(S(z)) * M, G(z) * M)$ with a loss mask ensures the local similarity between the originals and the generated results.

5 USER INTERFACE

Figure 6 shows our drawing assistance system. The system automatically records all the vertices of strokes and the stroke order, and converts the strokes into a raster image and corresponding guidance display on the sketch panel in real-time. Similar to ShadowDraw [2011], this system provides two types of guidance: “rough guidance” and “detailed guidance,” which users can switch at any time. Different colors denote semantics of the generated line drawing and we calculated the colorized guidance with a combination of few-shot semantic segmentation [Wang et al. 2019] and one-shot learning for StyleGAN controlling [Endo and Kanamori 2022] (Section 2 of the supplementary material contains more details). Detailed guidance shows the full-face portrait to the user as a prompt, while rough guidance shows the user a part of the face that has been drawn roughly or will be drawn soon as a prompt by predicting the user’s drawing progress. In rough guidance mode, only a single semantic part (color) will be shown according to the moving mouse point. Both are useful and high-quality: detailed guidance allows the user to understand the overall layout of the face to draw, and rough guidance allows the user to focus on drawing the local parts of the face. If the user is satisfied with the current guidance and does not want to change it any further for a sketch trace, he/she can press the “Pin” button to realize this purpose. When the sketch is completed, users can generate the final color image by clicking on the “reference image selection” button (face icon) to select the coloring style from the reference images. In contrast, with the “Eraser” tool, users right-click on a stroke and the system erases it. In addition, the “Undo” tool can delete the last stroke from the stroke list. Note that our system can also load (or export) user-drawn strokes by clicking the “Load” (or “Save”) buttons.

6 EXPERIMENTS AND RESULTS

6.1 Implementation Details

The image encoder and sketch encoder (Figure 4) adopted the pSp architecture [2021]. We chose layers 1-8 in \mathcal{W}^+ space as the structural code and layers 9-18 as the color code. We adopted a Ranger optimizer and set the learning rate to 0.0001. As a training environment, NVIDIA RTX3090 GPU was used to train our encoders on the Linux platform. Then, a workstation with Intel Core i7 8700, 3.20 GHz, NVIDIA RTX1070 GPU, and 64GB RAM on the Windows 10 platform was used as the testing environment. The input and output image resolution was 256×256 and 512×512 , respectively; 35,480 iterations with batch size 8 in Stage I and 3,808 iterations with batch size 16 (7+9 for Figure 5(b, d)) in Stage II. Because data were randomly generated by z , “max epochs” is replaced by “iterations.”

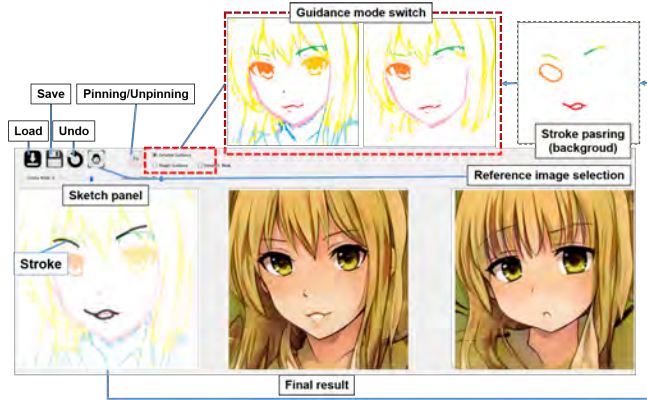


Fig. 6. User interface of the proposed AniFaceDrawing drawing assistance system. User guidance is generated using our sketch-based latent space exploration approach.

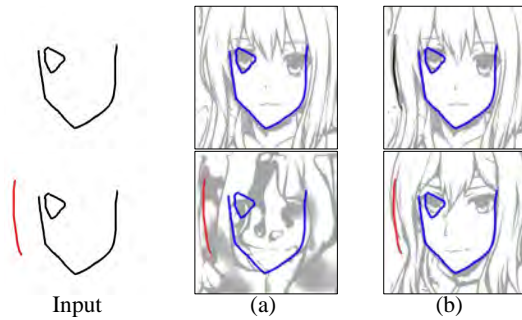


Fig. 7. Comparison of training (a) without background augmentation and (b) our training strategy in Stage II. When a red stroke is added, the result from the sketch encoder training without background augmentation is highly degraded.

6.2 Stability Testing

To test the stability of our sketch encoder during the freehand sketching process and verify the stroke-level disentanglement conditions, we first performed the following experiments.

Without background augmentation. The effect of background augmentation is shown in Figure 7. If sketches in the drawing process (Figure 5(b, c)) are trained without considering the background, the sketch encoder cannot correctly understand strokes associated with the background and project them near the correct position.

The influence of stroke order and multiple strokes on one facial part. Figure 8 compares the intermediate process of the same sketch with different stroke orders. It can be seen that the final results are not very different, but the intermediate processes maintain some diversity. This figure also shows that even if only one stroke is used for each part of the face during training, the generated guidance matches well when the user uses multiple strokes for the same part (e.g., the left eye and mouth).

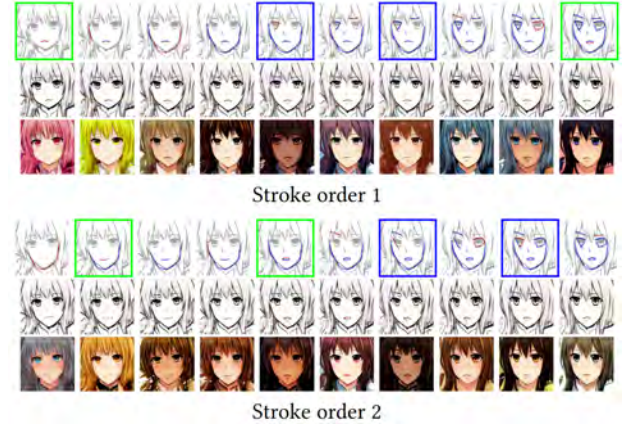


Fig. 8. The influence of stroke order and multiple strokes for one facial part (e.g., mouth (green) and left eye (blue)) in AniFaceDrawing.

Table 1. FID scores of baseline and our approaches. As a reference, the FID between Decoder1k and Danbooru1k is 70.86.

DBName	Ours	Decoder1k	Danbooru1k
Baseline	74.75	106.03	151.42
Ours	-	74.14	125.19

“Bad” stroke. If only some parts of a stroke contain valid information, then the stroke is considered as “bad.” In freehand sketching, “bad” strokes are not uncommon. The results generated by our method provide a suitable match to the valid part of such “bad” strokes. For example, the strokes representing the left eye in Figure 8 form a triangle, a shape that is not natural for representing the eye contour, while the generated result is still reasonable. Another example is the first stroke, which partially matches the normal face contour (see Figure 10), but our approach still manages to capture this information and ignore the meaningless part of this stroke.

6.3 Qualitative Results

We found that there is no S2I synthesis technique for anime portraits due to the lack of paired data. Therefore, we trained an additional sketch encoder for the complete sketch using a random cropping strategy with the state-of-the-art pSp encoder [2021] as a baseline for a fair comparison. Some examples of input sketches in the training step for this baseline encoder are shown in Figure 5(f). Except for this random cropping training strategy, which is used in iSketchN-Fill [2019], the hyperparameters and the architecture of the baseline network are the same as those in our sketch encoder. From the comparison results (Figure 10), we verified that our method can provide consistently high-quality guidance that better matches the input during the sketching process. Meanwhile, when the input contains “bad” strokes, the generated guidance will provide a reasonable result rather than complete matching guidance.

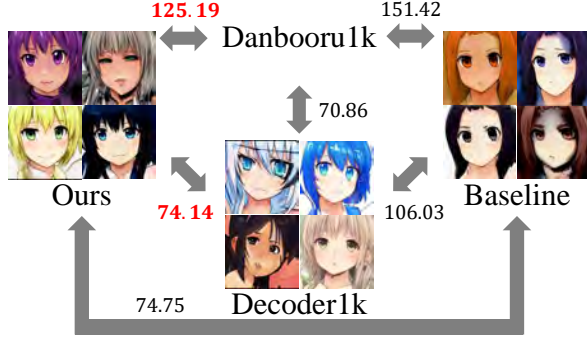


Fig. 9. Samples from the different datasets or approaches and the FIDs of each.

6.4 Quantitative Results

Our approach was evaluated from two aspects: the quality of the generated images, and the match between input and output. The quality of image generation affects both the quality of the guidance received by the user and the evaluation of the final generated result, so it was necessary to measure this indicator quantitatively in addition to the subjective evaluation of the user. For similar reasons, the match between the input sketch and the guide needed to be measured quantitatively to ensure that the validity of our approach was subjectively and objectively consistent. To evaluate usability and satisfaction, a user study was conducted for the overall system.

Quality of generated images. Unlike normal S2I synthesis, this work was dedicated to the stability of matching rough sketches and intermediate results throughout the drawing process. To evaluate the degree of matching between strokes and hints during the drawing process, this work used Fréchet Inception Distance (FID) to measure the gap between the generated images: First, users were asked to draw 10 sketches and record 177 images generated by our method as a database “Ours,” the results generated by the baseline method with the same input as a database “Baseline,” 1,000 randomly-generated images using StyleGAN in our decoder as a database “Decoder1k,” and 1,000 randomly-selected images from the Danbooru database [Branwen et al. 2019] as a database “Danbooru1k.” The FIDs between them are shown in Table 1. It can be seen that, in line with the observation in the qualitative results, our method generated better-quality images – similar to the images generated by the decoder in Decoder1k as well as to the real images in Danbooru1k. Figure 9 shows samples from each dataset or approach mentioned above, which makes results more intuitive.

Matching the generated image to the input sketch. To evaluate the match between the input sketch and the generated guidance, sketch-guidance matching can be thought of as a prediction problem. Although neither sketch S nor the generated line drawing L is reliable enough as ground truth, the input sketches are regarded as ground truth here because what we are concerned about is how the system will cater to the input with guidance. The evaluation indicators precision p , recall r , and $F1$ score for sketch-guidance matching are defined in Section 1 of our supplementary material.

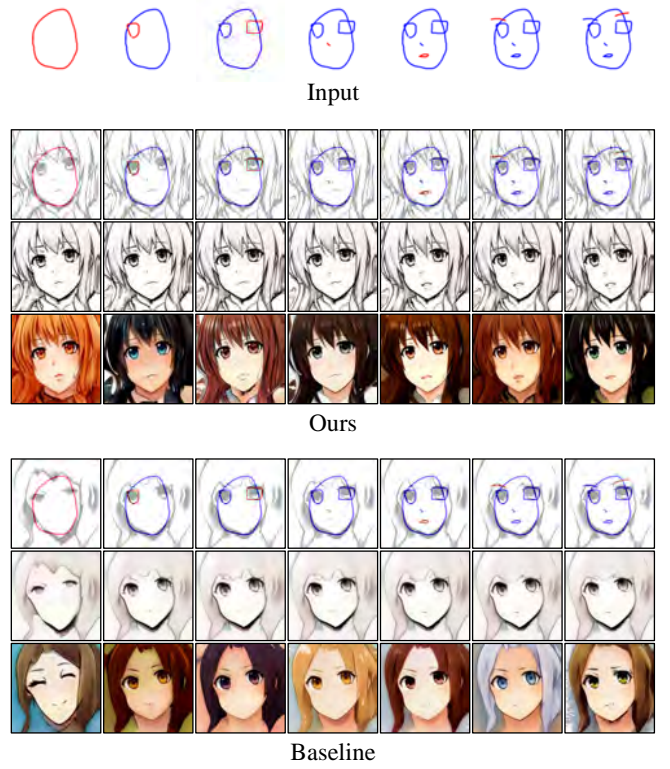


Fig. 10. Qualitative comparison with the same input sketch sequences. A red stroke represents the last stroke in a sketch.

Table 2. The average of different metrics from the proposed method and the baseline method (ours / baseline). At the beginning of the drawing process, the input sketches are usually sparser, which makes it difficult to generate matching results. Thus, the average recall score r of the first k strokes is more important.

Metrics \ k	1	3	6	9	whole process
p	0.04/0.03	0.04/0.05	0.05/0.07	0.07/0.08	0.12/0.12
r	0.48 /0.40	0.46 /0.39	0.45 /0.38	0.43 /0.37	0.39 /0.31
$F1$	0.07/0.05	0.07/0.09	0.09/0.11	0.11/0.13	0.17/0.16

Table 2 shows the comparison results between the proposed method and the baseline method (described in Section 6.3). It is even more important to provide high-matching guidance in the early stages of drawing when strokes are sparse. Therefore, the average recall rate was calculated for the first 1, 3, 6 and 9 strokes and for the whole sketch process (∞). We believe that the average recall is the best numerical description of sketch-guidance matching, as it is the proportion of overlapping areas of monochrome sketches and guidance over all current input sketches, and our results consistently outperformed and agreed with the baseline method in this metric throughout the drawing process, as shown in Figure 11. This result was also consistent with the observation of the qualitative comparison (see Figure 10). However, slightly lower values of our

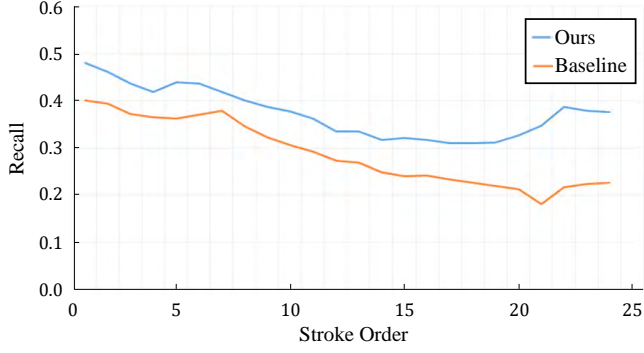


Fig. 11. Recall comparison as strokes increase. The average score of our method was higher than that of the baseline method.

method to the evaluation metrics p and $F1$ compared to those of the baseline, indicate that our method provides more details in the guidance generation. This experiment demonstrated that the guidance generated by this system can better match the input sketch, both at the beginning and throughout the drawing process. Based on the above results, recall can be considered a valid metric for measuring the match between the input sketch and output guidance.

7 USER STUDY

To verify the effectiveness of our anime-style drawing assistance system, we invited 15 participants (graduate students) to participate in user study. All participants were asked to draw anime-style portraits online using a remote mouse control. They were asked to draw the portraits freely and aimlessly, trying to draw as much detail as possible. Each user conducted anime portrait drawing twice: the first time to experience the whole process of drawing creation to familiarize themselves with the operation until they got used to this system and felt comfortable; and the second time, the participants completed the whole process independently. We instructed all participants on how to use AniFaceDrawing with user manuals (in the supplementary video). Before the hands-on experiments, they were asked to watch a tutorial video. All participants were required to carefully draw and select the most anticipated references for local guidance from several generated candidates after completing the global stage. If the generated guidance met their wishes and expectations, participants were required to press the “Pin” button to draw carefully to refine the input sketch. Participants could select a reference image for color portrait generation at any time during their drawing until they were satisfied with the results. Finally, they completed the questionnaire after finishing the second drawing.

7.1 Questionnaire Design

Our questionnaire consists of three parts: the system usability scale (SUS) [Bangor et al. 2008], creativity-support index (CSI) [Carroll et al. 2009], and a set of custom questions shown in Table 3 to investigate the relationship between user satisfaction and guidance matching. In the SUS, 10 questionnaire items were set up to capture subjective evaluations of the system usability. A five-point Likert scale was used in the evaluation experiment.

Table 3. Custom questions in our user study.

#	Question	Mean	SD
Q0	How would you rate your drawing skills for anime/real faces ?	2.07	1.10
Q1	Does the guidance match your sketch overall ?	4.07	0.26
Q2	Does the guidance match your sketch when drawing the mouth ?	3.40	1.12
Q3	Does the guidance match your sketch when drawing the left eye ?	3.93	0.88
Q4	Does the guidance match your sketch when drawing the right eye ?	4.00	0.76
Q5	Does the guidance match your sketch when drawing the nose ?	4.07	0.80
Q6	Does the guidance match your sketch when drawing the hair ?	3.47	1.51
Q7	Does the guidance match your sketch when drawing the facial contours ?	4.07	0.80
Q8	For your sketch and guidance, which facial balance is more reasonable ?	3.87	1.06
Q9	What is the quality of the guidance ?	4.00	0.53
Q10	Does the guidance maintain high quality in your sketching process ?	4.27	0.46
Q11	Is the rough semantics guidance mode helpful for your drawing ?	3.93	0.70
Q12	Is the detailed semantics guidance mode helpful for your drawing ?	4.13	0.74
Q13	Are you satisfied with the final coloring results ?	3.93	0.59
Q14	Does the guidance follow your will ?	3.87	0.35
Q15	Are you satisfied with the final sketch result ?	3.87	0.35

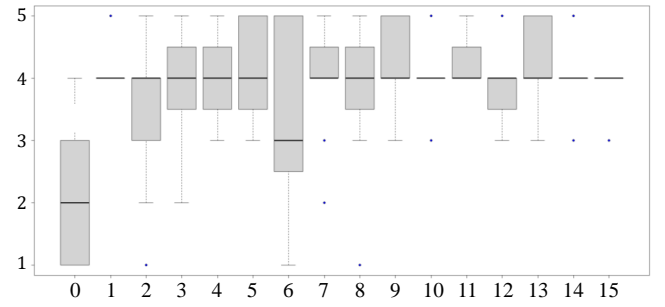


Fig. 12. Boxplots of custom questions in our user study. Questions Q0 to Q15 correspond to those in Table 3.

Since the purpose of this work is to support user drawing creativity, the CSI is used to quantitatively evaluate the effectiveness of the proposed method. The CSI score defines the creativity of the tool with six factors: collaboration, enjoyment, exploration, expressiveness, immersion, “results worth effort,” and is scored with a maximum of 100 points. Here, the “Collaboration” factor was set to 0 (not applicable) because there is no collaboration with another user in our task, as users completed the art drawing independently.

7.2 Results

This section discusses the visual results of AniFaceDrawing from users and user feedback from the user study. Table 3 shows the mean and SD of each question in our customized questionnaire, while Figure 12 shows the corresponding boxplots for these questions.

Visual results.

Figure 13 shows some examples of the results generated in the user study. Our system could successfully transform the user’s rough sketches into high-quality anime portraits. According to Q0 in our custom questions, 86.66% of participants thought their drawing skills were not good enough (less than or equal to 3) for anime portrait drawing. As shown in Figure 13, it can be concluded that even novices can make reasonable sketches with the help of the system and end up with high-quality color art drawings.

System usability. The average score for SUS drawing assistance for anime style was 73.84 ($SD = 20.04$). The upper and lower limits were 90 and 65, respectively. In addition, participants stated that “Overall

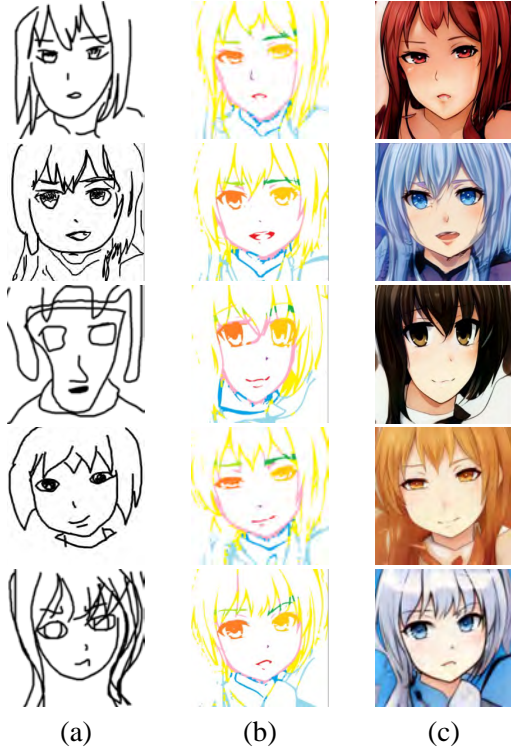


Fig. 13. Visual results from the user study. (a) the final user sketches, (b) the guidance in detail mode, and (c) the generated color drawings from (a) after the final reference image selection.

Table 4. CSI Questionnaire results in the user study.

Terms	Mean	SD
Collaboration	-	-
Enjoyment	27.93	10.09
Exploration	29.11	11.27
Expressiveness	23.50	6.30
Immersion	13.46	10.86
Results Worth Effort	22.54	11.65
CSI Score	77.69	

it's a good tool for those like me who do not have much drawing skills, and it's easy to use in terms of guidance generation and color selection" and *"I was not familiar with the operation when I first experimented, but I got an amazing generated result in the second experiment."* From these result, the usability of our drawing assistance system could be considered "good" for the anime style.

Creative support capability. As shown in Table 4, the average scores on CSI for anime style is 77.69. Although there is still room for improvement in terms of immersion and expressiveness, the system can be used to create sketch-based art drawings.

Time cost. After the user draws a stroke, AniFaceDrawing provides an average response time of 1.65 seconds for guidance generation. This response time appears to be too long for the user. Someone

said, *"One small problem is the not-so-short wait time after each stroke is completed."* This also affected the immersion score in Table 4. To improve the user experience, the calculation time needs to be further reduced. Although there was no time limit on our experiments, the average time to complete an experiment was about 9 minutes.

User-perception match degree. According to the results from Q1 to Q7 in Table 3, the average scores ranged from 3.40 to 4.07, which illustrates that our system can output relatively matching guidance to the input sketches during the drawing process. In these questions, a score of 5 indicates a *"complete match"* and 1 point means a *"complete mismatch."* Although the statistics showed that the input and output matched relatively well, users disputed whether the hair in the input sketch and the generated image matched. The proponents commented, *"The drawing assistance system performs better on hair and eyes, and can match well with the drawing person's draft to generate (anime portrait)."* Critics said, *"I tried to draw a double ponytail character, but couldn't achieve it"* and *"Some special hairstyles cannot be generated by this system."* The reason is that the stroke-level disentanglement is focused on facial contour features in the training step, and hair is trained with a random cropping strategy. Even so, most participants still tended to think that the sketch-hairs match is positive, which shows the generalization capability of our system at a certain level because we did not split the hair part into strokes to train in our training step.

User-perception quality. According to the results from Q8 to Q10 in Table 3, users believed that the system consistently produced high-quality and reasonably balanced facial guidance throughout the drawing process in anime-style drawing assistance. This result is consistent with the results of the qualitative experiments in Figure 10, and they corroborate each other.

User satisfaction with guidance. The results from Q11 to Q15 show that users generally agreed that our system provides good support for creating anime-style portraits, improving both the user's own sketches and producing a desirable final color image according to their expectations. Considering Q1 to Q7, the consistency of these scores illustrates that our approach achieved the optimal match between sketch input and guidance output so that users are satisfied with our drawing assistance during the drawing process.

8 CONCLUSION

We successfully re-ordered the feature vectors in latent space at the stroke-level by unsupervised learning with a drawing process simulation. The experiments demonstrated the stability and effectiveness of the proposed method. The experimental results show that our method can stably and consistently obtain high-quality generation results during freehand sketching, independent of stroke order and "bad" strokes. With our user study, AniFaceDrawing was proven to be effective and was able to create an anime portrait according to the users' intentions. As a limitation, the matching of the input sketch for the hair part could be improved due to the training strategy. As the results generated by our method are completely dependent on the decoder—that is, the pre-trained StyleGAN—the decoder, in turn, restricts the types of images generated (refer to supplementary material for more details). For example, since our pre-trained model is trained on an anime portrait database selected from Danbooru,

the generated results are all female. In addition, the current style is relatively constant: how to extract other styles from StyleGAN and make anime style more diverse and controllable will be explored in follow-up research. Meanwhile, how to expand the results with more styles, such as Ukiyo-e and painting, while keeping the strokes matching, is a promising topic for future work.

ACKNOWLEDGMENTS

This research was supported by the JAIST Research Fund, Kayamori Foundation of Informational Science Advancement, JSPS KAKENHI JP20K19845, and JP19K20316.

REFERENCES

- Emre Aksan, Fabrizio Pece, and Otmar Hilliges. 2018. DeepWriting: Making Digital Ink Editable via Deep Generative Modeling. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, New York, USA, 205:1–205:14. <https://doi.org/10.1145/3173574.3173779>
- Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. 2021. ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. In *Proceedings of IEEE/CVF International Conference on Computer Vision*. IEEE, Montreal, QC, Canada, 6691–6700. <https://doi.org/10.1109/ICCV48922.2021.00664>
- Aaron Bangor, Philip T. Kortum, and James T. Miller. 2008. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction* 24, 6 (2008), 574–594. <https://doi.org/10.1080/10447310802205776>
- Gwern Branwen, Anonymous, and Danbooru Community. 2019. Danbooru2019 Portraits: A Large-Scale Anime Head Illustration Dataset. <https://www.gwern.net/Crops#danbooru2019-portraits>. Accessed: 2023-01-25.
- Erin A. Carroll, Celine Latulipe, Richard Y. K. Fung, and Michael A. Terry. 2009. Creativity factor evaluation: towards a standardized survey metric for creativity support. In *Proceedings of the 7th Conference on Creativity and Cognition*. ACM, New York, USA, 127–136. <https://doi.org/10.1145/1640233.1640255>
- Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. 2020. DeepFace-Drawing: deep generation of face images from sketches. *ACM Transactions on Graphics* 39, 4 (2020), 72:1–72:16. <https://doi.org/10.1145/3386569.3392386>
- Chia-Hsing Chiu, Yuki Koyama, Yu-Chi Lai, Takeo Igarashi, and Yonghao Yue. 2020. Human-in-the-loop differential subspace search in high-dimensional latent space. *ACM Transactions on Graphics* 39, 4 (2020), 85:1–85:15. <https://doi.org/10.1145/3386569.3392409>
- Jungwoo Choi, Heeryon Cho, Jinjoo Song, and Sang Min Yoon. 2019. SketchHelper: Real-time stroke guidance for freehand sketch retrieval. *IEEE Transactions on Multimedia* 21, 8 (2019), 2083–2092. <https://doi.org/10.1109/TMM.2019.2892301>
- John P. Collomosse, Tu Bui, and Hailin Jin. 2019. LiveSketch: Query Perturbations for Guided Sketch-Based Visual Search. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Long Beach, CA, USA, 2879–2887. <https://doi.org/10.1109/CVPR.2019.00299>
- Tali Dekel, Chuang Gan, Dilip Krishnan, Ce Liu, and William T. Freeman. 2018. Sparse, Smart Contours to Represent and Edit Images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, UT, USA, 3511–3520. <https://doi.org/10.1109/CVPR.2018.00370>
- Yuki Endo and Yoshihiro Kanamori. 2022. Controlling StyleGANs using rough scribbles via one-shot learning. *Computer Animation and Virtual Worlds* 33, 5 (2022), e2102. <https://doi.org/10.1002/cav.2102>
- Emma Frid, Celso Gomes, and Zeyu Jin. 2020. Music Creation by Example. In *Proceedings of CHI Conference on Human Factors in Computing Systems*. ACM, New York, USA, 1–13. <https://doi.org/10.1145/3313831.3376514>
- Tsukasa Fukusato, Seung-Tak Noh, Takeo Igarashi, and Daichi Ito. 2020. Interactive Meshing of User-Defined Point Sets. *Journal of Computer Graphics Techniques* 9, 3 (2020), 39–58. <http://jcgt.org/published/0009/03/03/>
- Arnab Ghosh, Richard Zhang, Puneet K. Dokania, Oliver Wang, Alexei A. Efros, Philip H. S. Torr, and Eli Shechtman. 2019. Interactive Sketch & Fill: Multiclass Sketch-to-Image Translation. In *Proceedings of IEEE/CVF International Conference on Computer Vision*. IEEE, Seoul, Korea, 1171–1180. <https://doi.org/10.1109/ICCV.2019.00126>
- Yuxuan Han, Jialong Yang, and Ying Fu. 2021. Disentangled Face Attribute Editing via Instance-Aware Latent Space Search. In *Proceedings of International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Virtual, Montreal, 715–721. <https://doi.org/10.24963/ijcai.2021/99>
- Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. *Proceedings of Advances in Neural Information Processing Systems* 33 (2020), 9841–9850.
- Zhongyuan Hu, Haoran Xie, Tsukasa Fukusato, Takahiro Sato, and Takeo Igarashi. 2019. Sketch2VF: Sketch-based flow design with conditional generative adversarial network. *Computer Animation and Virtual Worlds* 30, 3–4 (2019), e1889:1–e1889:11. <https://doi.org/10.1002/cav.1889>
- Hysts. 2021. Anime Face Detector. <https://github.com/hysts/anime-face-detector>.
- Takeo Igarashi and John F. Hughes. 2001. A Suggestive Interface for 3D Drawing. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, USA, 173–181. <https://doi.org/10.1145/502348.502379>
- Takeo Igarashi, Satoshi Matsuoka, Sachiko Kawachiya, and Hidehiko Tanaka. 1997. Interactive beautification: a technique for rapid geometric design. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, USA, 105–114. <https://doi.org/10.1145/263407.263525>
- Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. CVF / IEEE, Long Beach, CA, USA, 4401–4410. <https://doi.org/10.1109/CVPR.2019.00453>
- Tero Karras, Samuli Laine, and Timo Aila. 2021. A Style-Based Generator Architecture for Generative Adversarial Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 12 (2021), 4217–4228. <https://doi.org/10.1109/TPAMI.2020.2970919>
- Yong Jae Lee, C Lawrence Zitnick, and Michael F Cohen. 2011. Shadowdraw: real-time user guidance for freehand drawing. *ACM Transactions on Graphics* 30, 4 (2011), 27:1–27:10. <https://doi.org/10.1145/2010324.1964922>
- Yuhang Li, Xuejin Chen, Binxin Yang, Zihan Chen, Zhihua Cheng, and Zheng-Jun Zha. 2020. DeepFacePencil: Creating Face Images from Freehand Sketches. In *Proceedings of the 28th ACM International Conference on Multimedia*. ACM, New York, USA, 991–999. <https://doi.org/10.1145/3394171.3413684>
- Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. 2017. Deep Sketch Hashing: Fast Free-Hand Sketch-Based Image Retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Honolulu, HI, USA, 2298–2307. <https://doi.org/10.1109/CVPR.2017.247>
- Y. Peng, Y. Mishima, Y. Igarashi, R. Miyauchi, M. Okawa, H. Xie, and K. Miyata. 2020. Sketch2Domino: Interactive Chain Reaction Design and Guidance. In *2020 Nicograph International*. IEEE, Tokyo, Japan, 32–38. <https://doi.org/10.1109/NicoInt50878.2020.00013>
- Tiziano Portenier, Qiyang Hu, Attila Szabó, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. 2018. FaceShop: Deep Sketch-based Face Image Editing. *ACM Transactions on Graphics* 37, 4 (2018), 99:1–99:13. <https://doi.org/10.1145/3197517.3201393>
- Hui Ren, Jia Li, and Nan Gao. 2020. Two-Stage Sketch Colorization With Color Parsing. *IEEE Access* 8 (2020), 44599–44610. <https://doi.org/10.1109/ACCESS.2019.2962579>
- Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2021. Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, virtual, 2287–2296. <https://doi.org/10.1109/CVPR46437.2021.00232>
- Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. 2022. InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 4 (2022), 2004–2018. <https://doi.org/10.1109/TPAMI.2020.3034267>
- Wanchao Su, Hui Ye, Shu-Yu Chen, Lin Gao, and Hongbo Fu. 2022. DrawingInStyles: Portrait Image Generation and Editing with Spatially Conditioned StyleGAN. <https://doi.org/10.48550/arXiv.2203.02762> arXiv:2203.02762
- Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an encoder for StyleGAN image manipulation. *ACM Transactions on Graphics* 40, 4 (2021), 133:1–133:14. <https://doi.org/10.1145/3450626.3459838>
- Hung-Yu Tseng, Matthew Fisher, Jingwan Lu, Yijun Li, Vladimir G. Kim, and Ming-Hsuan Yang. 2020. Modeling Artistic Workflows for Image Generation and Editing. In *Proceedings of 16th European Conference on Computer Vision*. Springer, Springer, Cham, 158–174. https://doi.org/10.1007/978-3-030-58523-5_10
- Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. 2019. PANet: Few-Shot Image Semantic Segmentation With Prototype Alignment. In *Proceedings of IEEE/CVF International Conference on Computer Vision*. IEEE, Seoul, Korea, 9196–9205. <https://doi.org/10.1109/ICCV.2019.00929>
- Holger Winnemöller. 2011. XDoG: advanced image stylization with eXtended Difference-of-Gaussians. In *Proceedings of 9th International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*. ACM, New York, USA, 147–156. <https://doi.org/10.1145/2024676.2024700>
- Weihao Xia, Yulun Zhang, Yujun Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. 2023. GAN Inverse: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 3 (2023), 3121–3138. <https://doi.org/10.1109/TPAMI.2022.3181070>
- Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. 2022. Pastiche Master: Exemplar-Based High-Resolution Portrait Style Transfer. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, New Orleans, LA, USA, 7683–7692. <https://doi.org/10.1109/CVPR52688.2022.00754>
- Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. 2020. Deep Plastic Surgery: Robust and Controllable Image Editing with Human-Drawn Sketches. In *Proceedings of 16th European Conference on Computer Vision*. Springer, Springer,

- Cham, 601–617. https://doi.org/10.1007/978-3-030-58555-6_36
- Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. 2021. Controllable Sketch-to-Image Translation for Robust Face Synthesis. *IEEE Transactions on Image Processing* 30 (2021), 8797–8810. <https://doi.org/10.1109/TIP.2021.3120669>
- Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M. Hospedales, and Chen Change Loy. 2016. Sketch Me That Shoe. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Las Vegas, NV, USA, 799–807. <https://doi.org/10.1109/CVPR.2016.93>