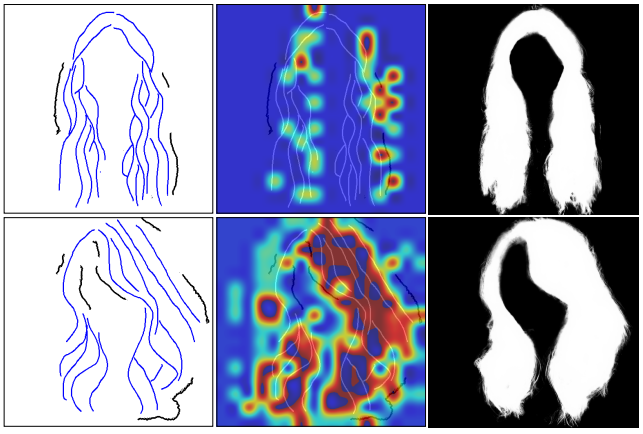
Supplemental Materials
SketchHairSalon: Deep Sketch-based Hair Image Synthesis

CHUFENG XIAO, School of Creative Media, City University of Hong Kong
DENG YU, School of Creative Media, City University of Hong Kong
XIAOGUANG HAN, SSE, The Chinese University of Hong Kong, Shenzhen
YOUYI ZHENG, State Key Lab of CAD&CG, Zhejiang University
HONGBO FU*, School of Creative Media, City University of Hong Kong

1 METHOD

1.1 Self-Attention Maps

Due to the relatively interpretable property of the attention modules, we can visualize the attention maps generated in S2M-Net to find out what the network learns. As shown in Figure 1, the most attention is assigned to the regions surrounding the input strokes. This is reasonable to encourage the network to produce plausible hair shapes.



(a) Input Sketch (b) Attention Map (c) Output Matte

Fig. 1. Visualization of attention maps generated at the third self-attention module in S2M-Net. The generated attention maps (b) are upsampled and placed over the input sketch (a), showing where they guide S2M-Net to focus on for generating the hair mattes (c).

1.2 Network Architecture and Parameter Settings

We adopt a two-stage framework to synthesize hair images directly from sketches. The two main networks in our framework, S2M-Net and S2I-Net, share a similar encoder-decoder architecture, as shown in Figure 2. For S2M-Net, the encoder consists of eight vanilla convolution layers, while the decoder has eight deconvolution layers with self-attention modules. S2I-Net extends from S2M-Net by adding one background encoder for blending features between the background and foreground regions, followed by one more convolution layer. We utilize skip connections between the encoder and the decoder.

We trained and tested our proposed system *SketchHairSalon* on a PC with Intel i7-8700 CPU, 32GB RAM and a single 2080Ti GPU. Since the matte and the hair image to be generated are the targets of different natures, we separately train S2M-Net (batch size = 20) and S2I-Net (batch size = 4) for 200 epochs on our dataset (augmented at each iteration) using the Adam solver with the fixed learning rate 1e-4 via the PyTorch framework. For S2M-Net, we train it on both of the unbraided and braided datasets. For S2I-Net, we first train the unbraided model on the unbraided dataset and then fine-tune the braided model (400K iterations) upon it on the braided dataset. The whole training process took around 3 days including the two stages, i.e., around 12 hours for training S2M-Net, around 2 days for training S2I-Net on unbraided hairstyles, and around 12 hours for fine-tuning S2I-Net on braided hairstyles.

1.3 Braided Models

As describe in Section 5.3, we construct five parametric braid 3D models given users' control. We denote the control as follows:

$$\begin{cases} \triangle Y = (B_{Y0} + B_{Y1})/2 \\ \triangle X = (B_{X0} + B_{X1})/2 \\ a(t) = |B_{X0} - B_{X1}|/2 \\ t = [0, |\triangle Y|] \end{cases}, \tag{1}$$

where B_0 and B_1 are the two boundary strokes. We also provide a extra parameter w directly from users, which can control the number of braided knots and the knot direction. With the parameters from users, the braid models can be reshaped. We provide five braided hairstyle models, including fishtail, rope, three-strand, four-strand and five-stand. Their detailed definitions are given below:

- Fishtail:

$$\begin{cases} L_0 : x = a(t)\sin(wt) + \triangle x,\ y = \triangle y, z = b\sin(2wt) \\ L_1 : x = a(t)\sin(wt + 2\pi/5) + \triangle x, y = \triangle y, z = b\sin(2(wt + 2\pi/5)) \\ L_2 : x = a(t)\sin(wt + 4\pi/5) + \triangle x, y = \triangle y, z = b\sin(2(wt + 4\pi/5)) \\ L_3 : x = a(t)\sin(wt + 6\pi/5) + \triangle x, y = \triangle y, z = b\sin(2(wt + 6\pi/5)) \\ L_4 : x = a(t)\sin(wt + 8\pi/5) + \triangle x, y = \triangle y, z = b\sin(2(wt + 8\pi/5)) \end{cases}, \tag{2}$$

*Corresponding author.

Authors' addresses: Chufeng Xiao, School of Creative Media, City University of Hong Kong, chufeng.xiao@my.cityu.edu.hk; Deng Yu, School of Creative Media, City University of Hong Kong, deng.yu@my.cityu.edu.hk; Xiaoguang Han, SSE, The Chinese University of Hong Kong, Shenzhen, hanxiaoguang@cuhk.edu.cn; Youyi Zheng, State Key Lab of CAD&CG, Zhejiang University, youyizheng@zju.edu.cn; Hongbo Fu, School of Creative Media, City University of Hong Kong, hongbofu@cityu.edu.hk.

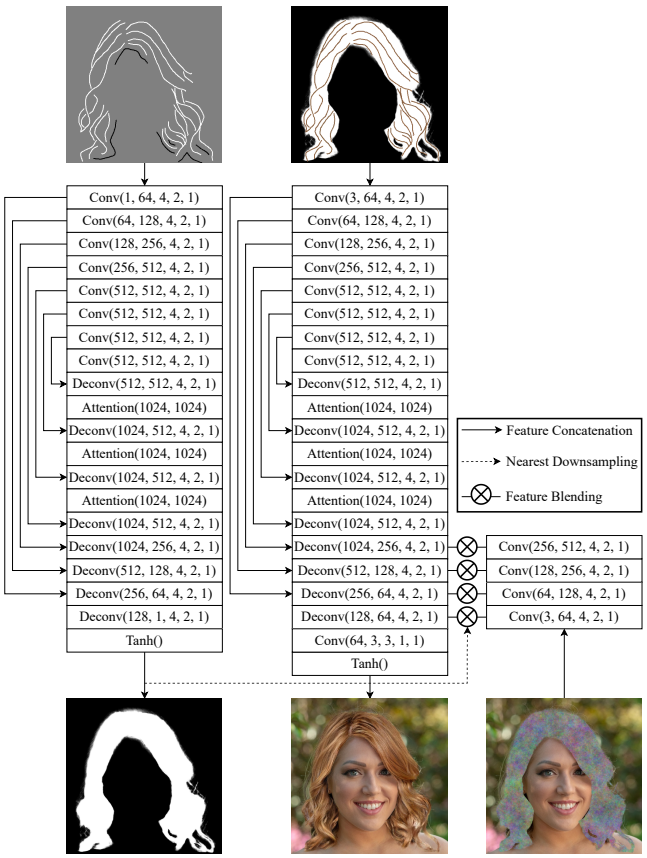


Fig. 2. Network architecture details. The left network is S2M-Net, while the right one is S2I-Net. The parameters of convolution and deconvolution layers are denoted as *Conv/Deconv(input_channel_number, output_channel_number, kernel_size, stride, padding_width)*, while those of the self-attention modules are *Attention(input_channel_number, output_channel_number)*. Original image courtesy of Charcharius.

- Rope:

$$\begin{cases} L_0 : x = a(t)\sin(wt) + \triangle x, y = \triangle y, z = b \cdot \sin(wt) \\ L_1 : x = a(t)\sin(wt + \pi) + \triangle x, y = \triangle y, z = b \cdot \sin(wt + 3\pi/4) \end{cases}, \tag{3}$$

- Three-strand:

$$\begin{cases} L_0 : x = a(t)\sin(wt) + \triangle x, \ y = \triangle y, z = b\sin(2wt) \\ L_1 : x = a(t)\sin(wt + 2\pi/3) + \triangle x, y = \triangle y, z = b\sin(2(wt + 2\pi/3)) \\ L_2 : x = a(t)\sin(wt + 4\pi/3) + \triangle x, y = \triangle y, z = b\sin(2(wt + 4\pi/3)) \end{cases}, \tag{4}$$

- Four-strand:

$$\begin{cases} L_0 : x = a(t)\sin(wt) + \triangle x, \ y = \triangle y, z = b \cdot f(t) \\ L_1 : x = a(t)\sin(wt + \pi/2) + \triangle x, y = \triangle y, z = b \cdot f(wt + \pi/2) \\ L_2 : x = a(t)\sin(wt + \pi) + \triangle x, y = \triangle y, z = b \cdot f(wt + \pi) \\ L_3 : x = a(t)\sin(wt + 3\pi/2) + \triangle x, y = \triangle y, z = b\sin(4(wt + 3\pi/2)) \end{cases}, \tag{5}$$

where

$$f(t) = \begin{cases} \sin(2t), \text{if } 2n\pi \le \mathrm{t} < 2\mathrm{n}\pi + \pi, \mathrm{n} \in \mathbb{Z} \\ \sin(4t), otherwise \end{cases} \tag{6}$$

- Five-strand:

$$\begin{cases} L_0 : x = a(t)\sin(wt) + \triangle x, \ y = \triangle y, z = b\sin(2wt) \\ L_1 : x = a(t)\sin(wt + 2\pi/5) + \triangle x, y = \triangle y, z = b\sin(4(wt + 2\pi/5)) \\ L_2 : x = a(t)\sin(wt + 4\pi/5) + \triangle x, y = \triangle y, z = b\sin(4(wt + 4\pi/5)) \\ L_3 : x = a(t)\sin(wt + 6\pi/5) + \triangle x, y = \triangle y, z = b\sin(4(wt + 6\pi/5)) \\ L_4 : x = a(t)\sin(wt + 8\pi/5) + \triangle x, y = \triangle y, z = b\sin(4(wt + 8\pi/5)) \end{cases}, \tag{7}$$

2 DATASET

Our dataset includes long (60%) and moderately short (40%) hair images on straight (1K), wavy (2K), and braided (1K) hairstyles on male (5%) and female (95%). We have many more female images since the female hairstyles exhibit significantly larger variety. Our system works for these hairstyles and their combinations on diverse structure and appearance. Figure 3 shows some representative examples in our dataset. Note that all of them are the testing results produced by our system given the sketches as input, including the generated mattes and the synthesized hair images. The original images are not shown due to the copyright issue.

3 EXPERIMENT

3.1 Matte VS. Mask

To show the effectiveness of hair mattes, we compare synthesized hair images based on the generated mattes with those based on the generated masks. We train two more networks with the same architectures (including the background blending module) as S2M-Net and S2I-Net, taking as input hair sketches to predict binary hair masks and further hair images. Figure 4 shows two examples of the matte-based and mask-based generation results. It is obvious that the generated mattes (Figure 4 (c)) can provide a more natural blending between hair and background regions, with fewer artifacts around their boundaries in the synthesized hair results (Figure 4 (e)), compared to the binary masks (Figure 4 (b)) and mask-based generated results (Figure 4 (d)), respectively.

In addition, we conducted one more perception study to evaluate the effectiveness of hair mattes for synthesizing hair images, compared to binary hair masks. We randomly picked 20 pairs of matte-based and mask-based generated hair images corresponding to the same sketch inputs, and invited 23 participants, each of which was asked to choose the better one in terms of visual naturalness from the two results in each pair (with the randomized presentation order of matte-based and mask-based results) . Finally, we got 23 (participants) × 20 (pairs) = 460 subjective evaluations. The voting results show our matte-based method received significantly more votes (95.8% voting rate) than the mask-based one (4.2% voting rate). This is consistent with our findings on the qualitative results (Figure 4).

3.2 Comparisons on Sketch-based Hair Image Synthesis

We compare S2I-Net with the state-of-the-art methods for image synthesis conditioned on hair sketches, including pix2pix, HIS and MichiGAN in terms of the visual quality of hair image generation. For a fair comparison, the inputs to these methods are the same as those to S2I-Net, i.e., hair mattes, hair sketches and background

Fig. 3. Diverse hairstyles in our dataset. The first two rows are a set of unbraided hairstyles, while the last two rows are a set of braided hairstyles. In each set, the top row contains the input sketches and the predicted mattes, while the bottom row gives the generated hair images by our system. Due to the copyright issue, we show our generated hairstyles, which are very close to those in the original images.

(a) Sketch (b) Mask (c) Matte (d) Mask-based (e) Matte-based

Fig. 4. Comparison of mask-based (d) and matte-based (e) generated hair images. Note that both of the masks (b) and the mattes (c) are auto-generated given the same hair sketches (a). Please zoom in to examine the differences. Original images courtesy of The Cooperative Trust and Tim Reckmann.

regions, though the ways to control the appearance are slightly different between MichiGAN and the other methods.

To be specific, the binary hair masks of MichiGAN are replaced with our hair mattes. We find that the appearance encoder of MichiGAN could not be re-trained well from scratch due to our not very large-scale dataset. Figure 5 shows the testing results from MichiGANs trained on our dataset with different strategies, including trained from scratch and fine-tuned on the given pretrained model, given the ground truth as the reference. We also show the results generated by the pretrained model of MichiGAN and ours for reference. It shows that the pretrained model can perform better for restoring the appearance of the ground truth than that trained from scratch, due to the large-scale dataset for training the pretrained model. Thus, to achieve the best performance and mainly compare the quality of hair structure generation, we fine-tune the pre-trained MichiGAN on our dataset. Note that since MichiGAN actually controls local structures via orientation maps, we also re-train their sketch-to-orientation network on our dataset, of which the orientation map is extracted from the hair images like MichiGAN does. In addition, since MichiGAN requires a reference image as an appearance condition, we set the ground-truth images as the appearance inputs for training and testing.

Pix2pix and HIS take the totally same inputs as those to our S2I-Net, with the colored hair sketches and background regions. To remain their original network architectures, the two inputs are concatenated and fed together into their networks. HIS consists of two stages: its first stage is similar to pix2pix, i.e., colored sketch-to-image translation, while the second stage takes as inputs the colored sketches and the orientation maps as well as texture maps extracted from the first-stage output images.

Note that we separately train the unbraided model and braided model for all the compared methods (the same way to train our models) until convergence. Besides the qualitative results shown in our main paper, we provide more results for reference, as shown in Figure 6.

Fig. 5. Comparisons of the results generated by MichiGAN trained with different strategies, as well as the provided pretrained model, given the ground truth as the reference. We also show ours result for reference. Original images courtesy of Oscar Campos-Morales, Tony Au, Kerry Goodwin, and Allef Vinicius.

Fig. 6. Comparisons of structure reconstruction with the state-of-the-art methods given the same input sketches (Top Row) and their generated mattes S2M-Net (Second Row). The sketches contains hair strokes and non-hair strokes (in black), of which the non-hair strokes are only used for generating hair mattes but removed for hair image synthesis. The left four columns are unbraided hairstyles, while the rest are braided hairstyles. Please zoom in to better examine the quality of synthesized results by the compared methods against the ground truth. Original images courtesy of AFGE, Pawel Loj, Noah Diamond, Bigashb and Stilfehler.