FaceSketching: Interactive Realistic Face Image Creation from Free-hand Sketches

Anonymous Author(s)
Submission Id: 1234

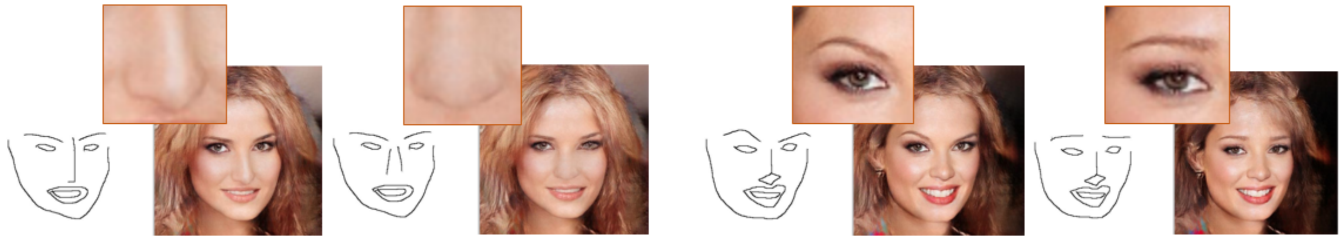

Figure 1: This is a teaser

ABSTRACT

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Image synthesis, sketch-based interface, face editing, deep neural network

ACM Reference Format:
Anonymous Author(s). 2020. FaceSketching: Interactive Realistic Face Image Creation from Free-hand Sketches . *ACM Trans. Graph.* 1, 1 (April 2020), 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Flexibly creating new content is one of the most important goals in both computer graphics and computer-human interaction. While sketching is an efficient and natural way for common users to express their ideas for designing and editing new content, sketch-based interaction techniques have been extensively studied since the very early stage of computer graphics [Chen et al. [n.d.], 2008; Igarashi et al. 1999; Sutherland 1964; Zeleznik et al. 1996]. Imagery content is the most ubiquitous media with a large variety of display devices everywhere in our daily life. Creating new imagery content is one way to show people's creativity and communicate smart ideas. In this paper, we target portrait imagery, which is inextricably bound to our life, and present a sketch-based system that allows common users to create new face imagery and edit them by specifying the desired facial shapes via free-hand strokes.

0730-0301/2020/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Deep learning techniques bring significant improvement on the realism of virtual images. Recently, a large number of studies have been conducted on face image editing. (Add papers.) All these methods require a real image as the start points. When creating a non-existed face, sketching is the most natural way to describe the desired structure and shapes. Treating free-hand sketches as a painting style, image translation techniques can be directly applied. (Cite image translation and style transfer papers.) However, the challenges in image translation is the edge alignment problem. The ambiguities and large discrepancy from real face shapes existed in imperfect free-hand sketches poses significant challenges for image translation networks which have strong power on realistic texture synthesis but perform poorly on geometric shape creation. Moreover, local editing is not supported in these uninterpretable end-to-end networks.

(Space and scale varying ambiguity in sketches) In order to support user's control on the creation and local editing from imperfect sketches, the ambiguities of hand-drawn strokes which vary in different facial regions at different scales should be solved for generating geometrically correct guidance. (Spatially-varying.) A user may draw the face contour in a casual way or depict the eye or eyebrow shape carefully in little ambiguity. (Temporally-varying.) As immediately see the generated face image, the user may add or edit local strokes with more careful control. Therefore, the ambiguity and distortion in different stage of the sketching process vary from the beginning to the end.

(Is speed an issue for sketch-base interface?)

(Contributions) In summary, our contribution in this paper is three-fold.

- Based on comprehensive analysis on the edge alignment issue in image translation DNNs, we propose an sketch-to-face system that allows users to control the desired facial shapes in adaptive precision.
- A spatial varying feature normalization in deep neural network to support global creation and local editing in multiple scales.

Figure 2: Generated face images from a sketch that does not follow the aligned face layout.

- Robust to various styles of hand-drawn strokes and allows progressively refinement to generate desired shapes and facial details.
- An easy-to-use system for face creation and editing which requires hand-drawn sketches only as input without any other specification of region masks.

2 RELATED WORK

(Image Translation)

(Face Generation and Editing)

All existing methods for local face editing require the user to provide masks and strokes manually. In comparison, strokes are only the input to indicate the desired shape. Our system automatically interprets the intended edit and produces the local change accurately. This greatly reduces users' burden and preserves the fluency of user interaction.

Only local edit in a relatively small area is supported. As reported in SC-FEGAN [], artifacts appear when complete a large region in FaceShop [].

3 DEEP NETWORK FOR SKETCH-PHOTO TRANSLATION

3.1 Edge alignment in baseline Model

Generating a realistic photo from sketch can be considered as an image translation problem. We use the state-of-the-art Pix2PixHd [] as our baseline.

We use the CelebA-HD dataset [] which contains xxx face images in WxH. All the face images are globally aligned according to their eye positions. For each photo, we generate sketch samples by xxx method. XX for training and xx for testing. Using this paired sketch-photo dataset, we trained our method and other state-of-the-art approaches for comparison.

3.1.1 Global alignment. While all the face images in the CelebA dataset [] are globally aligned with their eye positions, the learned generator implicitly embeds the global layout. Once the drawn sketches deviate from this implicitly embedded layout, the learned translation model generate awkward results, as Fig. 2 shows.

3.1.2 Data augmentation with geometric translation. A straightforward way is to augment the training set by random geometric transformation of input sketches. However, large interval/range of the transform yields un-convergence of the network training. We limit the transformation range to $(-\frac{\pi}{10}, \frac{\pi}{10})$ rotation, $(-\frac{W}{20}, \frac{W}{20})$ translation, and (scale?) in order to increase the tolerance of the trained model on the distortion and roughness of hand-drawn sketches. Moreover, as an interactive system, we also simply provide a reference sketch, like ShadowDrawing []. We place an averaged face contour image on the canvas to provide a reference coordinate system for user to draw their strokes. Therefore, the drawn sketches under this geometric reference will be located in or close to space of the training sketches.

Figure 3: Comparison between a sketch generated from edge detection and from semantic boundary.

3.1.3 Face Sketches and Data Augmentation. Paired face sketch-photo dataset is required for supervised sketch-to-face generation methods. Since there exits no large-scale face sketch datasets, the training face sketches used by existing methods are generated from face image dataset, e.g. CelebA-HQ face dataset, using edge detection algorithm such as HED [?]. However, the sketches generated by edge detection algorithm are sometimes incomplete or other problems. Although CSAGAN [?] applied self-attention mechanism to alleviate the incompleteness problem, the others remains. Therefore we use another method to generate clearer and complete sketches from face images. The CelebAMask-HQ dataset [?] provides a face semantic map for each face image in CelebA-HQ dataset. We basically use the boundary map of the semantic map as the sketch of the corresponding face image. Figure 3 shows an example of comparison between a sketch generated by edge detector and from semantic boundary.

A shortcut of sketches generated from semantic boundary (and those generated by edge detector) is the lines of sketches are perfectly aligned to the edges of the corresponding face images. In order to break the edge-alignment and mimic the hand-drawn sketches, we apply a deform to the lines. Specifically, we vectorize the lines of each sketches using AutoTrace algorithm [?], and add an offset randomly selected from $[-d, d]^2$ to the control points and end points of the vectorized lines, where d is the maximum offset and we set $d = 11$ in our experiments.

3.1.4 pix2pixHD without instance normalization. The baseline model, as well as many existing stylization DNNs, uses spatial normalization to extract style statistics, treating them as spatially uniform on the entire image. However, the shallow convolution layers typically extract texture or brightness statistics, which are varying dramastically on different local regions of a drand-drawn sketch. For example, there might be a large number strokes around hair regions or mouth regions to describe details. These heavy strokes bring significant difference while instance normalization at each convolution layer normalizes local patch features with a global factor. Therefore, the extracted features for identical eye strokes could be very different.

We think that strokes mainly describe the shape features, without little texture information.

3.1.5 Spatial Attention Pooling. When the input hand-drawn sketch is not well-drawn, it is a trade-off between the realism of the output face image and the alignment between the input sketch and the output face image. In order to alleviate the edge alignment between the input sketch and the output face image, we should relax the sharp sketch lines with one-pixel width. One of the straightforward ways is to smooth the lines of sketches using image smoothing algorithm. Another is to dilate the sketch lines so that the widths of lines are of multiple pixels [?]. However, the capacity of either the two hand-craft ways above is limited. Because the smoothness and the dilate radius are the same for all positions of the whole image. We argue that the balance between the realism of the output face image and the alignment between the input sketch and the output face image differs from one position to another across the face image. Therefore, the smoothness or the dilation radius should be spatial-specific.

Based on the discussion above, we propose a new module, called spatial attention pooling (SAP), to dilate the input sketch in a spatial-specific way. Let $\mathrm{r} = \{r_i | i = 1, 2, ..., N_r\}$ be a set of dilation radius. Given an input sketch $s \in \mathbb{R}^{H \times W}$, we first pass it through N_r pooling branches with kernel sizes of r to get $\{P_{r_i}(s) | i = 1, 2, ..., N_r\}$. Then we compute the spatial attention map $W \in \mathbb{R}^{N_r \times H \times W}$ with $W = Softmax(f(s))$, where $f()$ is implemented with two convolutional layers. A softmax layer which is computed over channels is added at the end of the convolutional layers, ensuring that for each position, the sum of weights of all channels equals to 1. The output of SAP is computed as:

$$SAP(s) = \sum_{i=1}^{N_r} W_i * P_{r_i}(s), \tag{1}$$

where W_i is the ith channel of W.

3.1.6 Losses. generator feature matching effect and losses summary

Let $G_q()$ be the feature maps of the qth generator layer. Given an input sketch s and the corresponding deformed sketch s', we compute the generator feature matching loss as:

$$\mathcal{L}_{GFM}(G) = \mathbb{E}_{s \sim p_{data}(s)} \frac{1}{N_Q} \sum_{q \in Q} \frac{1}{N_Q} \|G_q(s) - G_q(s')\|_1, \tag{2}$$

where n_q denotes the number of elements of G_q, Q indicates a set of the selected generator layers for computing this loss and the size of Q is N_Q. We select the xxx layers of the generator in our experiments.

Besides the generator feature matching loss $\mathcal{L}_{GFM}(G)$, for generator G and multi-scale discriminator $D = D_k | k = 1, 2, ..., N_D$, the adversarial loss $\mathcal{L}_{GAN}(G, D)$ and the discriminator feature matching loss $\mathcal{L}_{DFM}(G, D)$ are computed as the same form as those in pix2pixHD [?]. The objective of the proposed model is:

$$\min_G \max_D \mathcal{L}_{GAN}(G, D) + \lambda \mathcal{L}_{DFM}(G, D) + \mu \mathcal{L}_{GFM}(G). \tag{3}$$

where λ and μ are the weights for balancing different losses. We set $\lambda = xxx$ and $\mu = xxx$ in our experiments.

.

4 EXPERIMENTS

4.1 Sketch Interface

We develop a sketch-based interface which allows common users to describe their desired face and part shapes by a few strokes.

Figure 4: Sketch interface. Full interface with editing tools.

Once the user finishes his drawing, the generated face image is shown on the right bottom of the sketch? This will be helpful for local editing., as Fig. 4 shows. The user is allowed to edit the sketch by erasing strokes or adding new stroke to change eye shapes, noses shapes, eyebrows, and so on. Each round of face image generation after users' modification takes about ... seconds on a XXX GPU with xxGB memory.

4.2 Comparison with Image Translation networks

Existing DNNs for image translation can be trained for sketch-photo translation using the paired dataset.

4.3 Comparison with Image Editing

4.4 Limitations and future work

Add color

Combine with attribute

REFERENCES

Tao Chen, Ming ming Cheng, Ping Tan, Ariel Shamir, and Shi min Hu. [n.d.]. Sketch2Photo: Internet image montage. *ACM Trans. Graph* ([n. d.]).

Xuejin Chen, Sing Bing Kang, Ying qing Xu, and Julie Dorsey. 2008. Sketching reality: Realistic interpretation of architectural designs. *ACM Trans. Graph.* 27, 2 (2008), 11:1–11:15.

Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. 1999. Teddy: A sketching interface for 3-D freeform design. In *Proc. SIGGRAPH*. 409–416.

Ivan E. Sutherland. 1964. Sketch pad a man-machine graphical communication system. In *DAC*.

Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. 1996. SKETCH: An Interface for Sketching 3D Scenes. In *Proc. SIGGRAPH*. 163–170.