# ICCV 2017 Rebuttal – Face Sketch Synthesis

We thank the reviewers for detailed comments and suggestions and address their main concerns below.

**To R1**

Q1: Technical novelty

Ans: To the best of our knowledge, this paper is the first attempt to transfer styles from multiple images into one target image. It is very different from all previous face sketch synthesis methods which compose different spatial patch together. Besides, as noticed by R1, one of our contribution is successfully introducing style transfer to face sketch synthesis. It is not just a simple application, as Sec. 5.1 shows. We would like to emphasise that this purpose is not easy to achieve. Many of our early attempts failed, such as using weighted summation of gram matrices from a train set to approximate the target gram matrix or generating the target gram matrix by a linear transform from that of a photo etc. We also tried generating gram matrix for each level independently, however, the results are not good either. We believe this is because the correlation between the gram matrices crossing levels. The proposed pyramid column feature helps for our purpose.

Q2: Boundary discontinuities

Ans: In our implementation, we don’t decompose the photo into patches when we compute features. Instead, we keep a feature pool of the whole training sketch and select the pyramid column feature from the pool according to the patch location illustrated in Fig. 4. The receptive field of conv5\_1 of VGG is 132 x 132, which means the receptive field of pyramid column features are overlapped.

Since the boundary information is provided by the content image, we assume the discontinuities in Fig. 8 is caused by the limit size of training data and hence the insufficient training of the content network. As a result, it may experience some difficulties when generating edges for weak boundaries of real world photo and may produce some unnecessary shadings and lines.

Q3: Difference from [17, 18]

Ans: The key difference between [17, 18] and ours is the synthesis of texture and shading. [18] doesn’t have the texture part, and [17] used a feed forward network to generate texture. Different from them, we borrow the idea from style transfer, and use the proposed pyramid column feature to represent target textures. We combat the over-smooth artifact by texture synthesis network rather than L1 loss as adopted in content image. As shown in Figure 10, the results from content network are still suffering from this artifact, while this artifact is greatly alleviated when the style transfer is adopted as shown in Figure 10.We will add experiments to compare two losses in our final version.

Q4: Normalized Gram matrix as performance metric

Ans: Gram matrix of CNN feature maps helps to achieve good results on texture generation as suggested in [4]. As Fig. 10 shows, the textures are controlled by the style loss. While most methods can achieve good recognition results but looks quite different from human perspective, we draw the conclusion that textures are more important than content when evaluate the quality of sketch. And gram matrix is a good quantitative metric of texture.

Q5: Motivation of key component loss

Ans: People are more sensitive to key facial parts, such as mouth and nose, since these parts are more discriminative, which was varied and was adopted in facial components recognition tasks [i]. Therefore, we also expect our mode can be more focused on these parts and thus we impose a separate loss on these parts.

[i].  *Heisele, Bernd, et al. "Face recognition: component-based versus global approaches."*

**To R2**

Q1: How do we preserve sketch detail?

Our sketch details are preserved in a style transfer way. The style features are generated by the proposed pyramid column feature (Sec. 4.2), and the style representation is computed by Gram matrix (Sec. 3). The gram matrix has shown to be effective in representing textures in [4] and [5­­­­­­­­­­]. The style loss in Equ. (5) is the key part of sketch details. As Fig. 10 show, when we increase the style loss weight, more textures are added to the content image.

Q2: Limitations.

Ans: In real world photo, the background and boundary discontinuities are not well handled.

Due to the difficulty of collecting large training data, content network is insufficiently trained, which may cause the boundary discontinuities (see also Q2 for R1). And the whole process takes a bit long time. The patch searching and optimization process take most of the time (see also Q1 for R3) because they were run on CPU.

Q3: Figure size

We mistakenly put the wrong image in Fig. 1(e), we will fix it in the final version. We also noticed that the portrait sizes of different methods are not unified. That’s because some methods can’t generate the same size of sketches as input photos. For example, in FCNN [18], the author didn’t add padding before convolution to avoid border effects. Hence, the sketch result shrinks.

**To R3**

Q1: Complexity and memory

Our network is implemented by Keras with Theano backend. The program runs on a NVIDIA Titan X GPU with 12GB memory and 4.0GHZ Core i7 CPU.

Content network on GPU: We use SGD to train the network, and the learning rate decrease from 0.001 to 0.0001. It takes about 8 hours to converge. After the model is trained, we can generate a content image in 0.4s.

Patch feature searching on CPU: We first precomputed the feature maps of all training photos and store them on the disk. The photo is divided into 324 patches, and it takes around 30s to get the target sketch feature of all patches.

Texture optimization on CPU: We use LBFGS to update the content image. The optimization process takes about 66s.

The whole process takes about 100s, and all the programs above take less than 1GB GPU memory.

Q2: Real world quantitative evaluation

Ans: We agree that it would be better to have real world quantitative evaluation, however, due to the lack of public real world benchmark datasets, we can only use photos under laboratory conditions, which are also used by other methods. It is better for evaluation/comparison since we have ground truth. Collecting a real world datasets is a non-trival task, due to the copyright and portraiture right issues. We would consider the real world dataset collection in the future.

Q3: Do we need face alignment?

Ans: Yes, the face should be roughly aligned when detecting the corresponding patches. But we don’t need the alignment to be very accurate. Our method is robust to small scale and shift variation. For example, in Fig. 7 and Fig. 8, there are slight scale changes and offsets between photos, but the generated sketches are still good.