UV Completion with Self-referenced Discrimination

Jiwoo Kang[†], Seongmin Lee[†] and Sanghoon Lee[‡]

Department of Electrical & Electronic Engineering, Yonsei University, Seoul, Korea

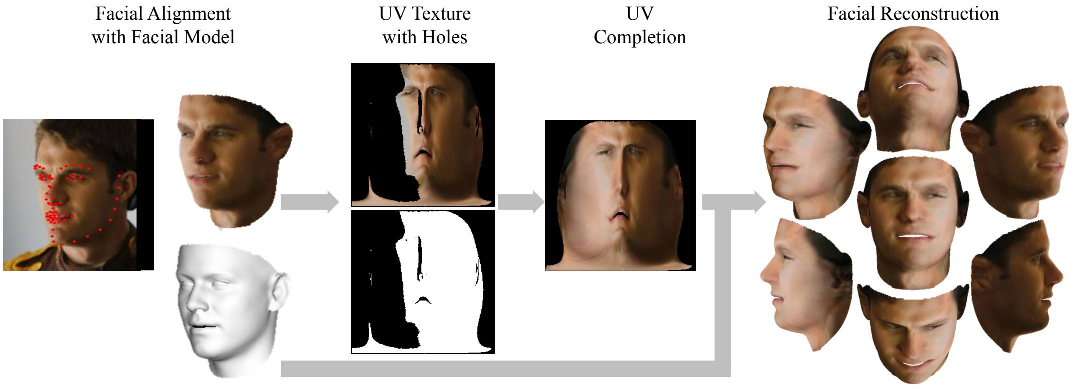


Figure 1: *A facial UV completion example for an occluded UV texture constructed by sampling an image using the fitted facial model. In this paper, a self-referenced discrimination method is presented to model the facial UV distribution without the complete UV ground-truth based on face symmetry, enabling the network to be trained to synthesize high-quality facial texture with a set of incomplete UVs.*

Abstract

A facial UV map is used in many applications such as facial reconstruction, synthesis, recognition, and editing. However, it is difficult to collect a number of the UVs needed for accuracy using 3D scan device, or a multi-view capturing system should be required to construct the UV. An occluded facial UV with holes could be obtained by sampling an image after fitting a 3D facial model by recent alignment methods. In this paper, we introduce a facial UV completion framework to train the deep neural network with a set of incomplete UV textures. By using the fact that the facial texture distributions of the left and right half-sides are almost equal, we devise an adversarial network to model the complete UV distribution of the facial texture. Also, we propose the self-referenced discrimination scheme that uses the facial UV completed from the generator for training real distribution. It is demonstrated that the network can be trained to complete the facial texture with incomplete UVs comparably to when utilizing the ground-truth UVs.

CCS Concepts

• ***Computing methodologies*** → ***Image processing;*** *Neural networks;*

1. Introduction

Significant progress of recent years in 3D face alignment enables us to obtain accurate and dense correspondence between a 3D face model and a 2D facial image [ZLL*17]. A facial UV map generated by sampling textures over the fitted image using the correspondence has been widely used in many applications such as facial reconstruction, face recognition, and face editing [DCX*18, TL19]. The facial UV has many missing pixels due to the self-occlusion of the face, i.e., the UV map is an image with hole regions. Fortunately, image inpainting methods recently proposed have demonstrated impressive completion capability of the hole regions on image [DCX*18, LRS*18, YLY*19]. In particular, Deng et al. [DCX*18] proposed a framework for Deep Convolutional Neural Network (DCNN) to complete the facial UV map with the self-occluded region. However, the corresponding ground-truth images without the holes for training the hole completion networks are necessarily required for the previous methods. Whereas lots of complete images

[†] J. Kang and S. Lee contributed equally to this work.

[‡] Corresponding author.

DOI: 10.2312/egs.20201018

are collectible, gathering a sufficient number of complete UVs to train the deep network would have come with restrictions in reality. Textured meshes from a 3D scan device or a multi-view capturing environment should be required to construct the complete UVs.

In this paper, we propose a facial UV completion framework for training the DCNN with the self-occluded UVs obtained from the alignment to a face image as described in Fig. 1. For training the network to encode and decode the incomplete region from the valid region without the use of the complete UV, we devise an adversarial network to model the complete UV distribution of the facial texture. The facial textures of left and right half-sides are not same exactly. However, their distributions for a sufficient number of faces become almost equal. At least, one of the horizontal half-sides on the facial UV has the valid pixels mostly, even for a profile face.

Motivated by these arguments, we introduce an adversarial network that utilizes the facial symmetry to model the complete UV distribution from an incomplete UV set, rather than using the symmetry as a hard regularization constraint used in many works related to facial UVs [TL19]. Also, we introduce a self-referenced discrimination scheme where the UV texture completed by the generator is used for training the real distribution instead of using one in the training set as described in Fig. 2. It is shown that the network can be trained with the self-referenced discrimination to complete the facial UV comparably to one trained with the ground-truth. By helping the network to be trained without the use of the ground-truth UVs, the method allows to simplify the completion and analysis of the facial UV constructed from numerous images in-the-wild.

2. UV Texture Completion

2.1. UV Construction

Assume that a facial mesh $\mathbf{v}_I$ is approximatively aligned to face on image, and the mesh has the top pointing up the y axis and the front pointing at the z axis. We use an unwrapped 2D texture of the face onto the UV space by using spherical unwrap. For the facial mesh $\mathbf{v}_I = (x,y,z)$, the projected point $\mathbf{v}_U = (u,v)$ onto the UV coordinates is computed as:

$$u = \alpha_u \cdot \arctan2(x,z) + \beta_u, \;\; v = \alpha_v \cdot \arccos\left(\frac{y}{r}\right) + \beta_v, \qquad (1)$$

where $r = \sqrt{x^2+y^2+z^2}$, and α_u, β_u, α_v and β_v are scale and translation constants to locate the unwrapped face in image boundaries.

2.2. Generation Network

For an input UV texture with holes $\mathbf{I}_{ref}$ and the corresponding mask $\mathbf{M}$ that indicates the valid regions on the texture, the generator network G is trained to reconstruct the facial texture $\mathbf{I}_{out}$ on UV space as an auto-encoder. As our network assumes that the ground-truth texture, or the correspondingly completed texture, is unavailable, the reconstruction loss (L_G) of the generator network is defined by measuring the pixel-wise l_1 norm on the valid texture regions as:

$$L_{G_{rec}} = \frac{1}{N_{\mathbf{I}_{ref}}} \left\| (\mathbf{I}_{out} - \mathbf{I}_{ref}) \odot \mathbf{M} \right\|_1 \qquad (2)$$

where $\odot$ denotes element-wise multiplication and $N_{\mathbf{I}_{ref}}$ is the number of elements in $\mathbf{I}_{ref}$ ($N_{\mathbf{I}_{ref}} = H \cdot W \cdot C$, and H, W and C are the height, width and channel size of $\mathbf{I}_{ref}$, respectively).

For our generator network, we use a U-Net structure similar to the ones used in [LRS*18, DCX*18], replacing all convolutional layers by the gated convolution layers proposed in [YLY*19] as described in Fig. 2. In particular, *ReLU* layers are used for activation functions for the encoder of the generator and *LeakyReLU* layers are employed for the decoder except the last layer, where a *tanh* activation layer is used for clipping the output value. The 3x3 filters are used over the convolutional layers except the first 3 convolutional layers of the encoder, where the 5x5 filters are employed. The convolution layers of the encoder operated with striding of 2 and the features are nearest neighbor up-sampled after each convolution layer in the decoder. The gated convolution layer helps the generator to cope with various shapes of the holes by generating a mask dynamically according to the input features, and the input holes on the UV texture do not have to be filled with the mean value of the valid texture or random noise as in [DCX*18]. We concatenate the incomplete UV with the corresponding mask and its left-to-right flipped image as an input of the generator network. The U-Net structure enables to obtain a high-quality image by preserving the image information on the original scale through the encoder-to-decoder skip connections. However, the reconstruction loss $L_{G_{rec}}$ in (2) forces the network to skip and pass the input features (valid textures) to the decoder rather than to encode textures for the incomplete regions from the features. The reconstruction loss is to preserve the high-detail texture on valid regions, while the adversarial loss defined in the following subsection helps the generator to complete the occluded regions on the UV, i.e., to encode the features for the incomplete regions.

2.3. Discrimination Network

Our key idea comes from the basis that the facial texture distributions of the left and right half-sides are almost equal when a sufficient number of facial images are given. Also, the facial UV has at least one of the half-sides that is composed of the valid pixels on most part, even for a profile face on the image. Thus, we model the texture distribution on the more valid side and manipulated them to complete the hole of the other side using a Generative Adversarial Network (GAN) loss. Nevertheless, the holes exist even on the more valid side of largely posed faces and they can force to generate the flipped hole to the other side.

To handle the problem, we present a discrimination scheme where the discriminator the reconstructed UV texture from the generator instead of using one of the training set. In the proposed method, the more valid side of the reconstructed UV texture is used as the data (real) distribution and the other side as the generator (fake) distribution. As the generator, $G : \left(\mathbf{I}_{ref}, \mathbf{M}\right) \rightarrow \mathbf{I}_{out}$, has the U-Net architecture, training the network with the loss $L_{G_{rec}}$ in (2) makes the generated UV texture from the network nearly equivalent to the input facial UV in a few training iterations, i.e., $\mathbf{I}_{ref} \cong \mathbf{I}_{out}$. In this case, the filters of the encoder in G cannot be trained since skipping image features to the decoder without encoding features is the best way to decrease the loss. When the adversarial loss is jointly used, the filters of the encoder are trained to fill the hole, resulting in the more completed UV texture while the facial texture on the input UV is preserved in original detail. Thus, $\mathbf{I}_{out}$ becomes more completed in both the half-sides than $\mathbf{I}_{ref}$ over the training

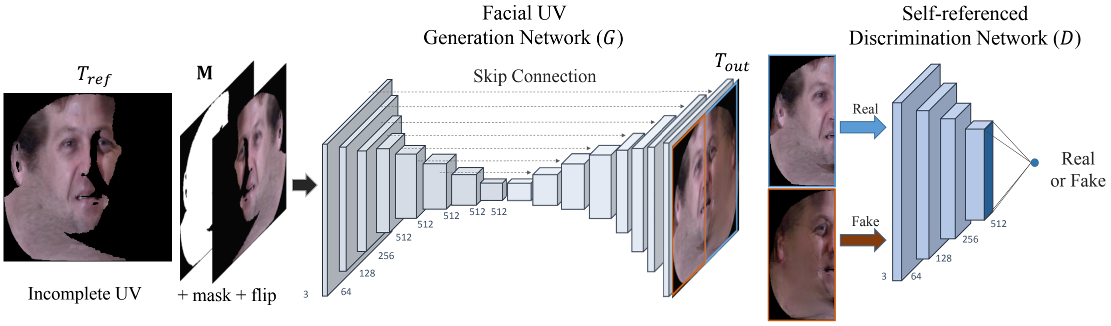


Figure 2: *The overview of our architecture for learning UV completion. It consists of a generation network G for the UV completion and a discrimination network for giving the context of occluded regions to the generation network during the training procedure. Between the left or right sides of the facial UV, the discrimination network learns to distinguish the more valid side as real and vice versa, while the generation network is trained to fool the discriminator by encoding features of incomplete regions.*

Algorithm 1 Split procedure of the output UV texture from the generator for the UV discrimination network.

```
function GET_SPLITS($\mathbf{I}_{out}$, $\mathbf{M}$)
    ($\mathbf{I}_{left}, \mathbf{I}_{right}$) $\leftarrow$ $split\_horizontal(\mathbf{I}_{out})$
    ($\mathbf{M}_{left}, \mathbf{M}_{right}$) $\leftarrow$ $split\_horizontal(\mathbf{M})$
    $\mathbf{I}_{right} \leftarrow flip\_horizontal(\mathbf{I}_{right})$
    $\mathbf{M}_{right} \leftarrow flip\_horizontal(\mathbf{M}_{right})$
    if $sum(\mathbf{M}_{left}) > sum(\mathbf{M}_{right})$ then
        $\mathbf{x}_{uv} \leftarrow \mathbf{I}_{left}$, $\mathbf{x}_z \leftarrow \mathbf{I}_{right}$
    else
        $\mathbf{x}_{uv} \leftarrow \mathbf{I}_{right}$, $\mathbf{x}_z \leftarrow \mathbf{I}_{left}$
    end if
    return ($\mathbf{x}_{uv}$, $\mathbf{x}_z$)
end function
```

iterations. Based on the property, we utilize $\mathbf{I}_{out}$ instead of $\mathbf{I}_{ref}$ as the data distribution of the adversarial loss in the discrimination network. In addition, we denoted the adversarial scheme as a self-referenced discrimination. The self-referenced discrimination prevents the flipped holes on the UV texture and enables the generator to complete the facial texture in detail. For $\mathbf{I}_{out}$, let two horizontal half-sides be $\mathbf{x}_{uv}$ and $\mathbf{x}_z$. The procedure to get $\mathbf{x}_{uv}$ and $\mathbf{x}_z$ from $\mathbf{I}_{out}$ is summarized in **Algorithm** 1. The discriminator is trained to distinguish those sides between real and fake, and the generator is trained to fool the discriminator. Thus, the adversarial losses for the discrimination (L_D) and generation networks ($L_{G_{adv}}$) are defined as:

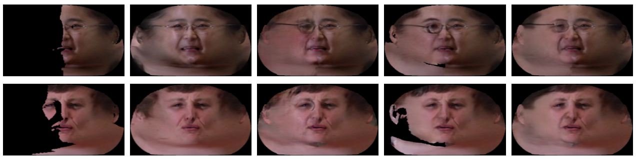
$$\begin{aligned} L_D &= \mathbb{E}_{\mathbf{x}_{uv}}\left[\log\left(D\left(\mathbf{x}_{uv}\right)\right)\right] + \mathbb{E}_{\mathbf{x}_z}\left[\log\left(1 - D\left(\mathbf{x}_z\right)\right)\right], \\ L_{G_{adv}} &= \mathbb{E}_{\mathbf{x}_z}\left[\log D\left(\mathbf{x}_z\right)\right]. \end{aligned} \tag{3}$$

The total loss of the generation network is defined as the summation of the reconstruction and the adversarial losses: $L_G = L_{G_{rec}} + \lambda_G L_{G_{adv}}$, where λ_G is a constant for balancing between two losses and we use $\lambda_G = 0.001$ for our experiments. The loss $L_{G_{adv}}$ enables the encoder filters of the generation network to be trained, yielding the more completed UV texture not only on the side $\mathbf{x}_z$ but the side $\mathbf{x}_{uv}$. The loss $L_{G_{rec}}$ helps the facial texture on the UV plane to be preserved in an original resolution. The discriminator is then trained by the updated regions, i.e., more completed regions. We use the spectral normalization (SN) [MKKY18], which is recently proposed weight normalization method, to more stabilize the training of the discriminator. The architecture of the discriminator, as well as the generator, is described in Fig. 2. The discriminator use the 5x5 filter and *LeakyReLU* activation for all convolution layers in our implementation.

3. Experiments

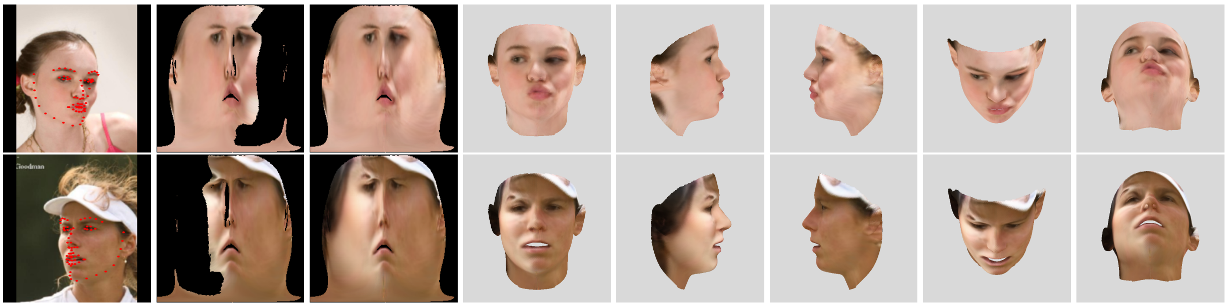
Training Data and Procedure For quantitative measurements of the completion performance, we employed the UV dataset constructed from Multi-PIE [GMC*10] with 337 identities in the work of [DCX*18]. The UV dataset is composed of 2,514 different facial UV maps for various illumination environment and 50,280 UV images in total. For qualitative analysis, we used the CelebA dataset [LLWT15], which composed of 202,599 facial images with 10,177 identities. To obtain the alignments of the facial model on images of the CelebA dataset, we employed one of the state-of-the-art methods [ZLL*17]. The UV maps corresponding to the alignment were constructed using the spherical unwrap in (1). The image size of 256x256 of the UV texture was used to feed the network for the training. Our model was trained with Tensorflow r1.14, CUDNNv7.3, and CUDA10.0. We used Adam for the optimization and trained using a single NVIDIA 2080 Ti (11GB) with a batch size of 16 and a learning rate of 0.0002.

Methods We evaluated the qualitative performance of our network with two variants. The first one used the flip-symmetric reconstruction loss used in [TL19] instead of using the adversarial loss in (3), for training the generator. The other one used the more valid side of the UV texture on the training set instead of using one obtained from the generator, for training the discriminator. Here, we denote the first and second ones as 'flip-loss' reconstruction and



(a) *Input UV* **(b)** *GT* **(c)** *Flip-loss* **(d)** *Data-ref.* **(e)** *Ours*

Figure 3: *Qualitative comparisons of UV completion results.*



(a) *Image* **(b)** *Incomplete* **(c)** *Completed* **(d)** *Frontal View* **(e)** *Left View* **(f)** *Right View* **(g)** *Top View* **(h)** *Bottom View*

Figure 4: *Qualitative results on the CelebA dataset. Best viewed in zoom-in.*

Table 1: *Results of measurements on the Multi-PIE UV dataset.*

	l_1 (%)	l_2 (%)	TV (%)	PSNR	SSIM
UV-GAN	-	-	-	26.5	0.898
Flip-loss	8.575	1.508	2.489	24.6	0.871
Data-referenced	12.168	3.028	2.732	21.8	0.846
Ours	7.257	1.083	2.212	26.7	0.907

'data-referenced' discrimination methods, respectively. We used the l_1 error, l_2 error, total variation (TV) loss, peak signal-tonoise ratio (PSNR), and structural similarity index (SSIM) to measure the completion performance of the network following the previous image completion works [YLY*19, DCX*18]. The qualitative measurements are summarized in Table. 1. The PSNR and SSIM measurements of the UV-GAN, which was trained with the complete UV ground-truth, are additionally represented in the table as reported in [DCX*18]. Our method outperforms the results of the flip-loss reconstruction and data-referenced discrimination. Also, it can be shown that through self-referenced discrimination, PSNR and SSIM values comparable to the UV-GAN can be obtained without the use of the ground-truth UV textures. Figure 3 shows some visualizations of the completion results on the Multi-PIE dataset. Although data-referenced discrimination fills the holes in fine resolution, the capacity of the filling is limited to the validity of the more valid sides of the training set. The flip-loss reconstruction method shows the lack of details in the completed regions. To verify our method on faces in-the-wild, the completion was performed on the UV maps constructed from the alignments to the CelebA dataset. Some results of those are visualized in Figure 4, where synthesized faces in different facial views are constructed from the completed UV. Our method completes the holes in a resolution of the valid region while preserving the original texture of the facial UV. Nevertheless, for some misaligned images where background textures are regarded as valid pixels in the networks, it might not correct the background region of the valid mask and complete the hole by using the incorrect clues. Thus, our method is somewhat dependent on the alignment accuracy used to construct the UV plane.

4. Conclusions

We have introduced a facial UV completion framework to train the deep neural network with a set of incomplete UV textures. When a sufficient number of facial images are given, the facial texture distributions of the left and right half-sides becomes almost equal. Based on the fact, we modeled the complete UV of the facial texture using the GAN. For training the GAN, we proposed the self-referenced discrimination scheme that uses the UV texture more completed from the generator as the real distribution rather than using one of the training set. Through the experiments, it is verified that the network can be trained to complete the facial hole on the UV plane without the use of the ground-truth UVs. Also, the quantitative results were comparable to those of the network trained with the ground-truth. Future works could be correcting non-facial texture, and expanding our method to a general inpainting scenario.

Acknowledgement

This work was supported by Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2016-0-00406, SIAT CCTV Cloud Platform).

References

[DCX*18] Deng J., Cheng S., Xue N., Zhou Y., Zafeiriou S.: UV-GAN: Adversarial facial uv map completion for pose-invariant face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (2018).

[GMC*10] Gross R., Matthews I., Cohn J., Kanade T., Baker S.: Multi-pie. *Image and Vision Computing 28*, 5 (2010), 807–813.

[LLWT15] Liu Z., Luo P., Wang X., Tang X.: Deep learning face attributes in the wild. In *International Conference on Computer Vision* (2015).

[LRS*18] Liu G., Reda F. A., Shih K. J., Wang T.-C., Tao A., Catanzaro B.: Image inpainting for irregular holes using partial convolutions. In *European Conference on Computer Vision* (2018).

[MKKY18] Miyato T., Kataoka T., Koyama M., Yoshida Y.: Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957* (2018).

[TL19] Tran L., Liu X.: On learning 3d face morphable model from in-the-wild images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[YLY*19] Yu J., Lin Z., Yang J., Shen X., Lu X., Huang T. S.: Free-form image inpainting with gated convolution. In *IEEE International Conference on Computer Vision* (2019).

[ZLL*17] Zhu X., Lei Z., Li S. Z., et al.: Face alignment in full pose range: A 3d total solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).