

# High-Frequency Face Estimation from a Single Image

Duan Gao

We introduce a method to recover high-frequency albedo from a single image (Figure 1). We could recover camera pose, facial geometry, lighting and low-frequency facial albedo directly from the source image. Moreover, we extract high-frequency details from the source image and combine it with the low-frequency albedo to generate the final high-frequency albedo. Based on these recovered information, we have achieved different applications, including face relighting in different illumination condition and face aging by modifying the age attributes.

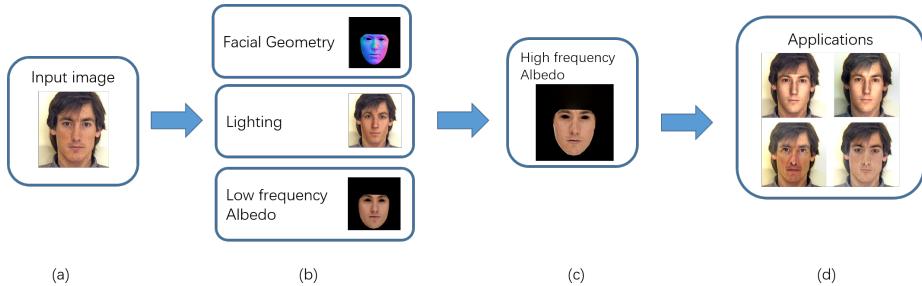


Figure 1: An overview of our method. For a given facial image (a), we can recover facial normal map, low-frequency albedo map and illumination (b, Section 1.2) based on traditional method. Then we can combine high-frequency albedo which are extracted from input image with low-frequency albedo to generate final high-frequency albedo map (c, Section 2). We can use the high-frequency albedo in many important applications like face relighting (the top line of (d), Section 3.2) and face aging (the bottom line of (d), Section 3.3).

## 1 3DMM based shape and texture recovery

In this section, we overview the traditional 3DMM based method

## 1.1 Fundamental

**Facial landmark points detection.** Facial landmark points detection is important in both pose estimation and geometry recovery. For an input facial image, we use OpenFace toolkit [2, 3], which is a state-of-art tool for facial landmark detection, to detect 68 landmark points. Meanwhile, we label these 68 landmark points manually in 3DMM according to their semantics. Then, we establish a correspondence between the input 2D image and the 3D models. Fig.2 shows the locations of 2D landmarks and corresponding landmarks in 3D.

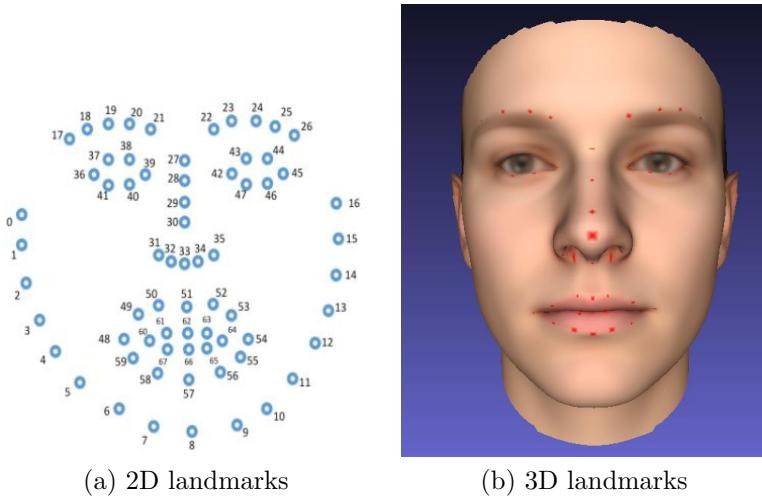


Figure 2: 2D landmarks and 3D landmarks.<sup>1</sup>

**3DMM.** 3D morphable model (3DMM) [5] is a powerful facial representation. It analyzed hundreds of human faces in the database using PCA. 3DMM represents a face using two coefficient vectors, i.e., a shape vector  $\alpha_i$  ( $0 \leq i < m$ ) and a texture vector  $\beta_i$  ( $0 \leq i < m$ ). The facial shape and the facial texture could be computed through a dot product between the coefficient vectors and eigenvectors, respectively:

$$s = \mu_s + \sum_{i=0}^{m-1} \alpha_i s_i, \quad (1)$$

$$\rho = \mu_\rho + \sum_{i=0}^{m-1} \beta_i \rho_i,$$

where  $s, \rho$  are facial shape and facial texture, respectively;  $\mu_s, \mu_\rho$  are the mean shape and mean texture, respectively;  $s_i, \rho_i$  ( $0 \leq i < m$ ) denote the shape

<sup>1</sup>Please visit [https://github.com/TadasBaltrusaitis/OpenFace](https://github.com/TadasBaltrusaitis/OpenFace/wiki/Output-Format) to see more details about landmark points. The 3D landmarks are visualized by MeshLab [9]. Note that this manual labelling between 2D landmarks and 3D landmarks only need once.

and texture eigenvectors, respectively;  $m$  is the number of used eigenvectors. In our implementation, we use the open-source 3DMM implementation, Basel Face Model (BFM) [12].

**Spherical harmonic illumination.** Spherical harmonic (SH) function is widely used in computer graphics to represent environment illumination. We also use SH to represent environment lighting. Environment lighting  $L$  is reduced to a few SH coefficients  $l_j$  ( $0 \leq j < 9$ ). The environment lighting  $L$  could be recovered from the SH coefficients by:

$$L = \sum_{j=0}^{j<9} l_j \mathbf{h}_j \quad (2)$$

where  $\mathbf{h}_j$  denote the order- $j$  spherical harmonic basis function.

## 1.2 Shape and texture estimation

According to [4], we roughly estimate facial shape and textures based on 3DMM. This process consists of three steps: pose and shape estimation, Laplacian deformation and texture parameter optimization.

**Pose and shape estimation.** In order to recover shape and texture parameters of 3DMM from the given image, we also need to estimate the pose parameters for the image. This is done by registering the 2D landmarks in the image and the corresponding 3D landmarks on face mesh. Specifically, we recover pose and shape parameters by minimizing the L2 difference between them:

$$\arg \min_{\alpha, \gamma, P, M} \sum_k \|L_{2D}(k) - PML_{3D}(k)\|_2 \quad (3)$$

where  $\alpha, \gamma$  are 3DMM shape and expression coefficients, respectively;  $L_{2D}(k), L_{3D}(k)$  denote the  $k$ -th 2D landmark point in the image, and 3D landmark point on the face mesh, respectively. Pose parameters include the perspective project matrix  $P$  and the model matrix  $M$ .

**Laplacian deformation.** After the previous optimization, there may still exist some landmark points that cannot be matched very well. To reduce the matching error, we use classic Laplacian deformation [13, 10] to deform the meshes to improve matching. Note that this step is optional.

**Texture parameter optimization.** Recall that we use SH to represent environment illumination. To recover texture from the single image, we need to solve for both lighting (SH coefficients) and texture (3DMM texture coefficients). We assume facial material is Lambertian. We could obtain a rendered facial image by:

$$I(\mathbf{p}) = E(\mathbf{n}_p)\rho(\mathbf{p}) = \rho(\mathbf{p}) \sum_{j=0}^{j<9} \mathbf{h}_j(\mathbf{n}_p) \cdot l_j \quad (4)$$

where  $\mathbf{p}$  is a pixel,  $I(\mathbf{p})$  is the pixel color,  $\mathbf{n}_p$  is the normal,  $\rho(\mathbf{p})$  is the albedo color (Equation 1) and  $E(\mathbf{n}_p)$  is the irradiance.

We minimize the L2 difference between the source image  $I_{src}$  and the rendered image:

$$\arg \min_{\beta, l_j} \sum_{\mathbf{p}, j} \|I_{src}(\mathbf{p}) - I(\mathbf{p})\|_2 \quad (5)$$

where  $\beta$  is 3DMM texture coefficients (Equation 1).

## 2 High-frequency facial albedo recovery

The albedo map generated by 3DMM based method misses many important details like pores, freckles and beard. In this section, we will introduce how to extract these details from the source image and then composite them into the low-frequency albedo.

### 2.1 Extracting high-frequency details

After shape and texture estimation (Section 1.2), we have obtained the rendered image  $I$  (Equation 5), which is rendered by optimized facial geometry, facial albedo and lighting parameters obtained in Equation 5. In the ideal situation, the rendered image  $I$  should be identical to the source image  $I_{src}$ . However, since the albedo is always low-frequency due to the nature of 3DMM representation, so the rendered image  $I$  is low-frequency too.

To extract high-frequency details, we divide the source image  $I_{src}$  by the rendered image  $I$  and obtain a ratio image  $I_{ratio}$ ,

$$I_{ratio} = \frac{I_{src}}{I}. \quad (6)$$

After that, the high frequency albedo map  $I_{hf}$  could be obtained by multiplying the low frequency albedo map  $I_{lf}$  with the ratio image:

$$I_{hf} = I_{lf} \times I_{ratio}. \quad (7)$$

Fig.3 shows an example of the process. Our key observation is that the similarity of  $I$  and  $I_{src}$  guarantees  $I_{ratio}$  only store high frequency details. However, the high frequency details not only include albedo map details, but may also include unresolved highlight. Therefore, besides the above process, which could add high-frequency details back into the albedo map, we also need to remove unexpected specular highlights.

### 2.2 Specular removal

Due to 3DMM texture and SH based environment lighting are both low frequency, the specular highlights are always missing in the rendered image  $I$ . Therefore, the ratio image  $I_{ratio}$  and the high frequency albedo map  $I_{hf}$  probably contain specular highlights. Meanwhile, the low frequency albedo map  $I_{lf}$  is rather smooth, and generally will not contain specular highlights and shadows.

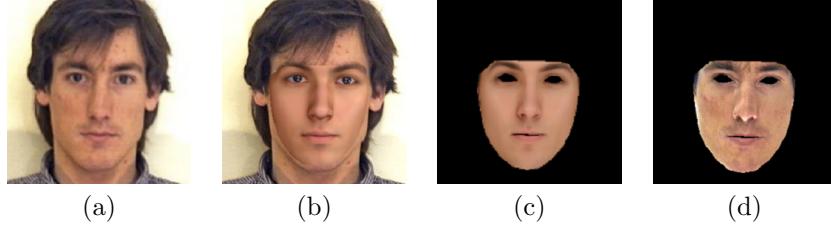


Figure 3: Extracting high-frequency details. (a) the source image  $I_{src}$ ; (b) the rendered image  $I$ ; (c) the low-frequency albedo  $I_{lf}$ ; (d) the high-frequency albedo  $I_{hf}$ .

The above observation give us a hint on how to remove the highlight. We can remove the highlight in  $I_{hf}$  under the guide of  $I_{lf}$ , in other words, we can propagate the high-frequency details from  $I_{hf}$  to  $I_{lf}$  without changing the major color of  $I_{lf}$ .

We present an optimization target function as following,

$$\arg \min_{\mathcal{I}} \|\nabla \mathcal{I} - \nabla I_{hf}\|_2 + w \|\mathcal{I} - I_{lf}\|_2 \quad (8)$$

where  $\mathcal{I}$  is the optimized image,  $\nabla$  is gradient operator,  $w$  is the weight factor. In our implementation, we set  $w$  to 0.5. After optimization, the result image  $\mathcal{I}$  will have two properties:

1. The pixel color is similar to the color of the reference image  $I_{lf}$ ;
2. The pixel's gradient is similar to the gradient of input image  $I_{hf}$ ;
3. notice that specular highlights are usually wider than high frequency facial details, so the specular areas will be eliminated with proper weights factor.

Fig.4 shows an example of specular removal.

### 2.3 [Optional] Editing propagation.

In actual applications, the environment region like the background and the forehead needs to vary together with the facial area. For example, if we change the lighting to directional lighting that points from left to right, we need to propagate the facial editing to background so that the left part should be lighter and the right part should be darker.

For a given input image A and a relighting image B (image B is only changed in facial region  $\Omega$ ). Our goal is to propagate the editing to whole image.

In our implementation, we solve the following optimization target to propagate the editing,

$$\arg \min_C \|\Delta C - \Delta A\|_2 + w \|\Omega_C - \Omega_B\|_2 \quad (9)$$

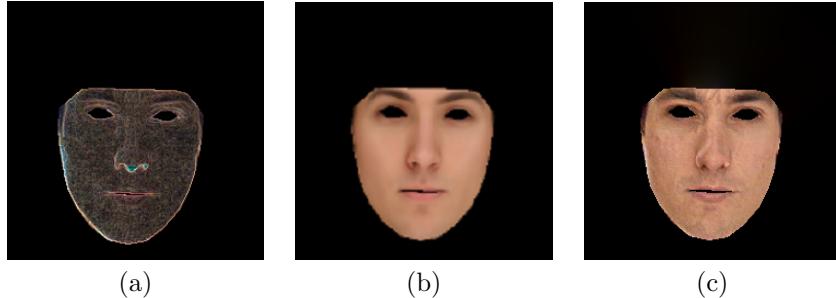


Figure 4: Illustration of specular removal method. Image (a) is the gradients of  $I_{hf}$ , image (b) is  $I_{lf}$  and the right image is the result image  $\mathcal{I}$ .

where  $C$  is the result image,  $\Delta$  is the laplacian operator,  $w$  is the weight factor and  $\Omega$  is facial region. In our implementation, we set  $w$  to 50.

### 3 Experiments

In our experiments, we use the PICS [1] and CACD [7, 8] as the facial image dataset. We implement our method on an Intel i7-6700K CPU, the total optimization takes roughly two seconds while the input image size is  $250 \times 250$ . Specifically, the time cost of pose and geometry optimization accounts for 15%, the cost of texture and lighting optimization accounts for 70%, the cost of specular removal optimization accounts for 10% and the other parts accounts for 5%.

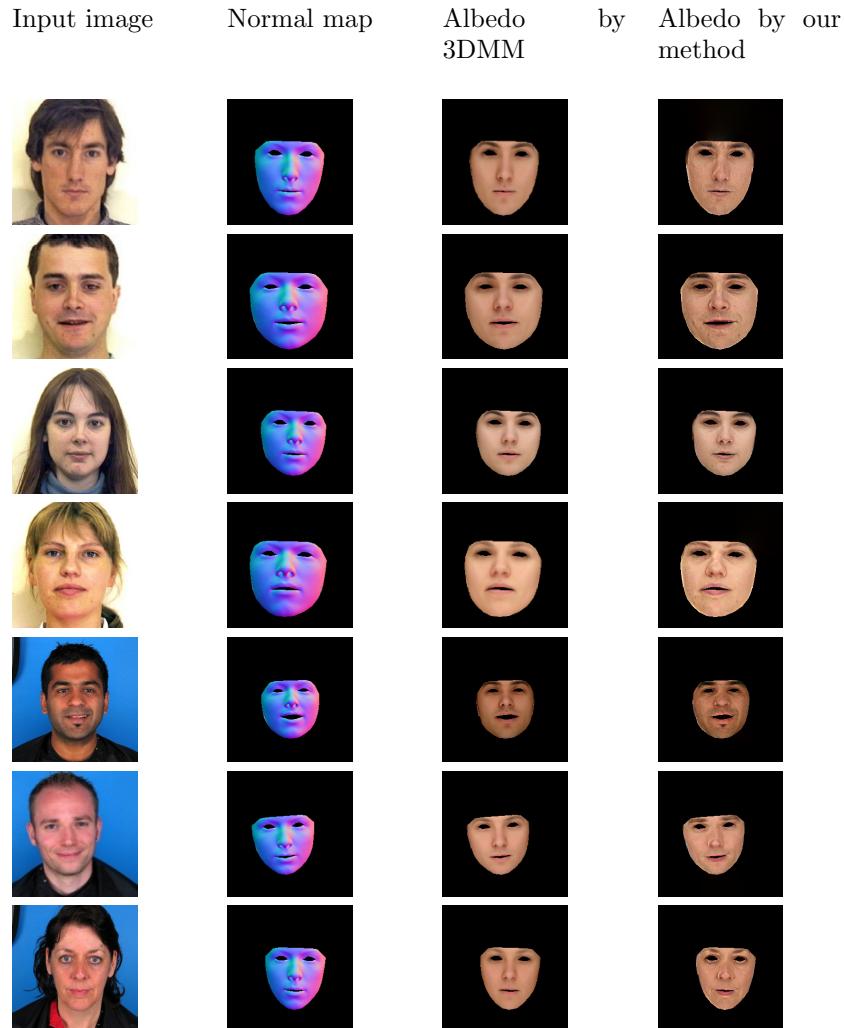
We preprocessing each input image as following,

1. detect facial area in each image and remove the images which have more than one faces or have no face;
2. crop every image to  $250 \times 250$  and make sure the face is in the center of image. It should be noted this step is just for convenient because our method can handle arbitrary size.
3. detect landmark points inside facial area and then calculate the convex hull of landmark points to generate matte image.

#### 3.1 Albedo recovery

We compare the albedo map which generated by our method with the results generate by traditional 3DMM-based method. In our implementation, we set the dimension of 3DMM shape and texture vectors to 99, and dimension of expression vectors to 29. In specular removal of albedo map, we set the weight factor in range [0.1, 1.0]. We use the Levenberg Marquardt algorithm [11] to solve all optimizations in our method.

**Basic albedo recovery comparison.** Figure 5 shows the results of albedo recovery by our method and traditional 3DMM based method [4]. We can see that the results of our method contain more high frequency details in facial area than the results of traditional 3DMM based methods. The artifacts in marginal area are caused by inaccurate facial contour detection.



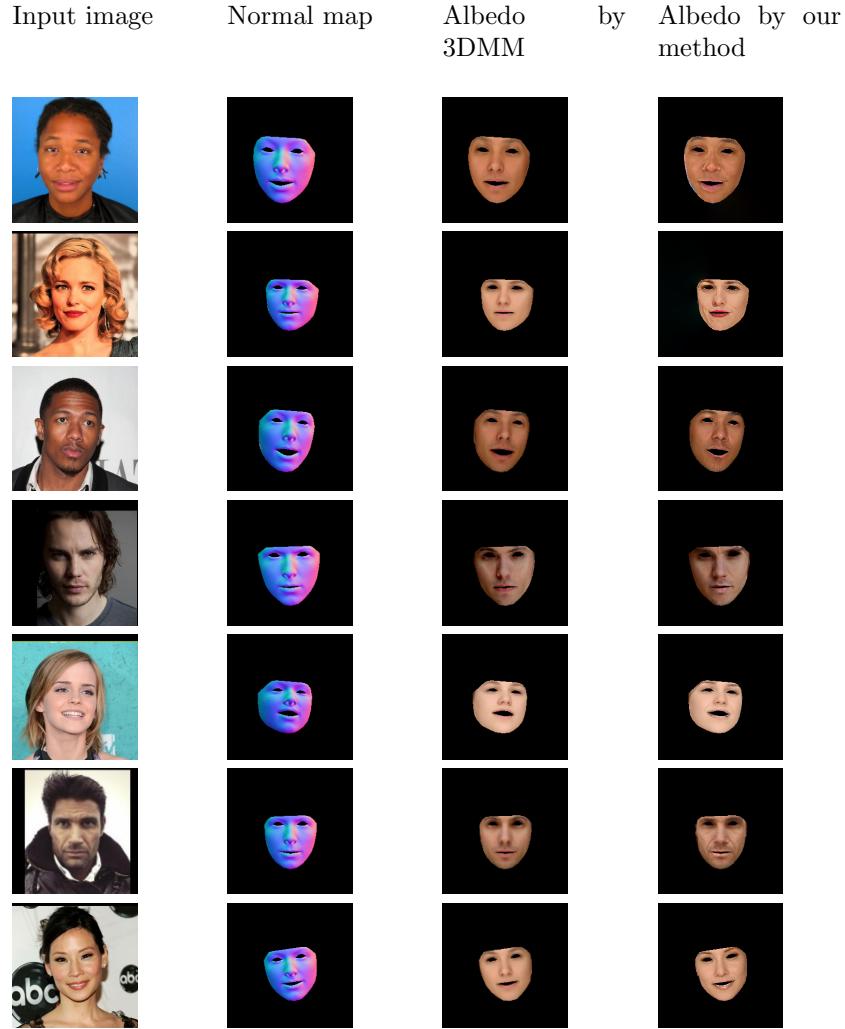


Figure 5: Face albedo recover experiments. The first column is input image, the second column is recovered normal map, the third column is recovered albedo map by traditional 3DMM-based method, the last column is recovered albedo map by our method.

**Complex situations.** There may exist some other objects (glasses, hair and so on) in the facial area. In our method, we simply regard these objects as a part of facial texture. Thus, these objects will show in the recovered albedo map. In most cases, our method can recover proper albedo in other facial areas. Figure 6 shows that our method can handle occlusion like glasses and hair.

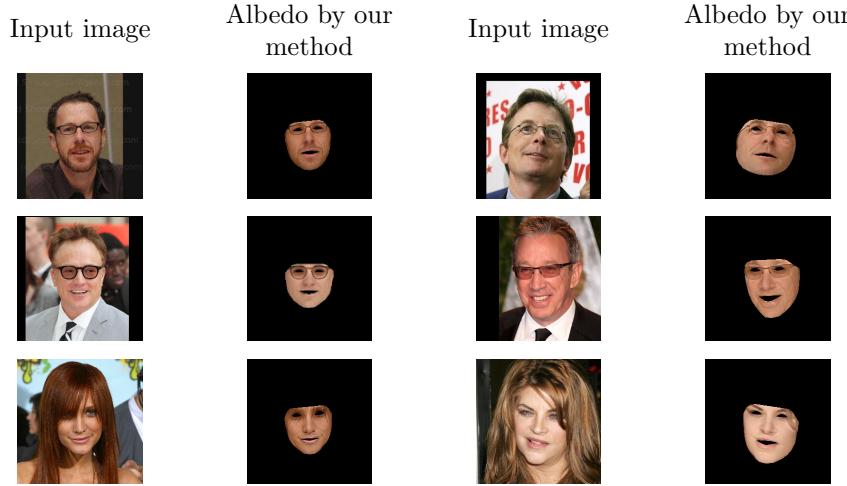


Figure 6: Face albedo recovery in occlusion situation. Column 1 and 3 are input images and column 2 and 4 are corresponding albedo map. Row 1 and 2 show glass occlusion situation, row 3 shows hair occlusion situation. In both cases, our method can recover qualified results.

### 3.2 Face relighting

The recovered high-frequency albedo map could be used to achieve face relighting effects. We could directly change the SH coefficients of the environment lighting, and obtain the relighted results through rendering. We show two relighting examples by a direction light in Figure 7, and show two lighting transfer examples in Figure 8.

In both cases, we use recovered albedo map, normal map and lighting parameters to re-render the facial area. Editing propagation (introduced in Section ??) is applied to propagate the changes to other area in the image like forehead, clothes and background.

### 3.3 Face aging

Basel face model has many facial attributes including ages, weights and gender. By adjusting the age attribute, the albedo map recovered by 3DMM will look younger or elder. We can composite this modified albedo with the albedo recovered by our method to generate facial aging results. Figure 9 shows the face aging results using our method.

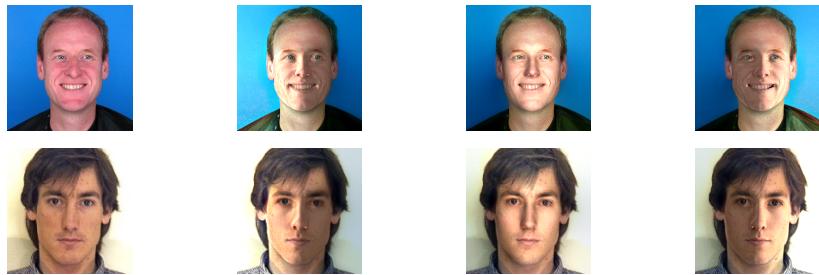


Figure 7: Face relighting results under distant lighting. We use Bling-Phong shading model [6] in this experiments. First column are source images, column 2-4 are relighting results under distant light. The direction of distant light are rotating from left to right.



Figure 8: Face relighting results under SH illumination. First column are source images, second column are target images and the last column are relighting results.

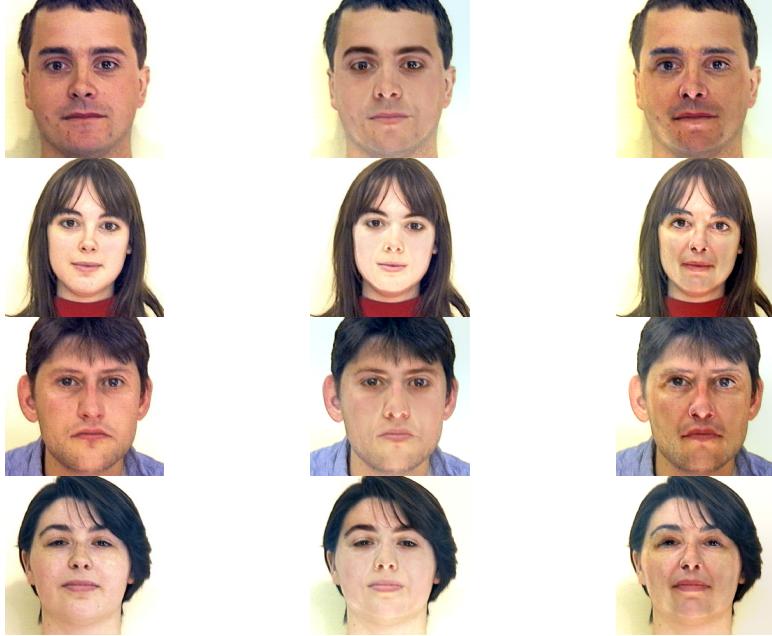


Figure 9: Face ageing results. The first column are input images, second column are younger faces and the last column are elder faces. The forehead area may looks inconsistent due to the facial landmark limitation.

We could generate new facial images with either younger or elder ages from a single image, for both man and woman faces.

## References

- [1] [pics.stir.ac.uk](http://pics.stir.ac.uk).
- [2] T. Baltrušaitis, P. Robinson, and L.-P. Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 354–361. IEEE, 2013.
- [3] T. Baltrušaitis, P. Robinson, and L.-P. Morency. Openface: an open source facial behavior analysis toolkit. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–10. IEEE, 2016.
- [4] V. Blanz, S. Romdhani, and T. Vetter. Face identification across different poses and illuminations with a 3d morphable model. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 202–207. IEEE, 2002.
- [5] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.
- [6] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH Computer Graphics*, volume 11, pages 192–198. ACM, 1977.

- [7] B.-C. Chen, C.-S. Chen, and W. H. Hsu. Cross-age reference coding for age-invariant face recognition and retrieval. In *European Conference on Computer Vision*, pages 768–783. Springer, 2014.
- [8] B.-C. Chen, C.-S. Chen, and W. H. Hsu. Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. *IEEE Transactions on Multimedia*, 17(6):804–815, 2015.
- [9] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. Meshlab: an open-source mesh processing tool. In *Eurographics Italian Chapter Conference*, volume 2008, pages 129–136, 2008.
- [10] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rossi, and H.-P. Seidel. Differential coordinates for interactive mesh editing. In *Shape Modeling Applications, 2004. Proceedings*, pages 181–190. IEEE, 2004.
- [11] J. J. Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.
- [12] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *Advanced video and signal based surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 296–301. Ieee, 2009.
- [13] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM, 2004.