Self-Supervised 3D Face Reconstruction via Conditional Estimation – Appendix –

A. Approach

A.1. Image Cropping

The viewpoint $\boldsymbol{v}$ comprises the scale factor $\boldsymbol{v}_1$, 3D spatial rotation parameters $[\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4]$, and 3D translation parameters $[\boldsymbol{v}_5, \boldsymbol{v}_6, \boldsymbol{v}_7]$. The original image $\boldsymbol{I}$ is cropped to its canonical view in 2D plane with viewpoint v. The cropping is given by $(\boldsymbol{I} \circ \boldsymbol{v})(x', y') = \boldsymbol{I}(x, y)$, where the transformation from (x', y') to (x, y) is formulated in the following.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \exp(\boldsymbol{v}_1) \cdot \cos \boldsymbol{v}_4 & \exp(\boldsymbol{v}_1) \cdot \sin \boldsymbol{v}_4 & \boldsymbol{v}_5 \\ -\exp(\boldsymbol{v}_1) \cdot \sin \boldsymbol{v}_4 & \exp(\boldsymbol{v}_1) \cdot \cos \boldsymbol{v}_4 & \boldsymbol{v}_6 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{10}$$

Bilinear interpolation is used if x or y is not an integer.

A.2. Weak Perspective Transformation

The 3D spatial rotation is represented by a rotation vector $\boldsymbol{w} = [\boldsymbol{v}_2; \boldsymbol{v}_3; \boldsymbol{v}_4] \in \mathbb{R}^{3\times1}$: the unit vector $\boldsymbol{u} = \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$ is the axis of rotation, and the magnitude $\phi = \|\boldsymbol{w}\|_2$ is the rotation angle. The weak perspective transformation is used to project the world-coordinate facial shape $\boldsymbol{S}$ to image-coordinate $\boldsymbol{Q}$, as formulated in

$$\begin{bmatrix} \boldsymbol{Q}(i,1) \\ \boldsymbol{Q}(i,2) \\ \boldsymbol{Q}(i,3) \end{bmatrix} = \exp(\boldsymbol{v}_1) \cdot \left(\boldsymbol{w}\boldsymbol{w}^\intercal \begin{bmatrix} \boldsymbol{S}(i,1) \\ \boldsymbol{S}(i,2) \\ \boldsymbol{S}(i,3) \end{bmatrix} + (\cos\phi) \cdot (1 - \boldsymbol{w}\boldsymbol{w}^\intercal) \begin{bmatrix} \boldsymbol{S}(i,1) \\ \boldsymbol{S}(i,2) \\ \boldsymbol{S}(i,3) \end{bmatrix} + (\sin\phi) \cdot \boldsymbol{w} \times \begin{bmatrix} \boldsymbol{S}(i,1) \\ \boldsymbol{S}(i,2) \\ \boldsymbol{S}(i,3) \end{bmatrix} \right) + \begin{bmatrix} \boldsymbol{v}_5 \\ \boldsymbol{v}_6 \\ \boldsymbol{v}_7 \end{bmatrix}. \tag{11}$$

A.3. Barycentric Coefficients

Given the vertices of a triangle $(\boldsymbol{Q}(i), \boldsymbol{Q}(j), \boldsymbol{Q}(k))$ and its enclosing grid point (x, y) on image. The barycentric coefficients can be computed by

$$\begin{gathered} \boldsymbol{d}_i = \begin{bmatrix} \boldsymbol{Q}(j,1) - \boldsymbol{Q}(i,1) \\ \boldsymbol{Q}(j,2) - \boldsymbol{Q}(i,2) \end{bmatrix}, \quad \boldsymbol{d}_j = \begin{bmatrix} \boldsymbol{Q}(k,1) - \boldsymbol{Q}(i,1) \\ \boldsymbol{Q}(k,2) - \boldsymbol{Q}(i,2) \end{bmatrix}, \quad \boldsymbol{d}_k = \begin{bmatrix} x - \boldsymbol{Q}(i,1) \\ y - \boldsymbol{Q}(i,2) \end{bmatrix}, \\ d_{ii} = \boldsymbol{d}_i^\intercal \boldsymbol{d}_i, \quad d_{jj} = \boldsymbol{d}_j^\intercal \boldsymbol{d}_j, \quad d_{ij} = \boldsymbol{d}_i^\intercal \boldsymbol{d}_j, \quad d_{ki} = \boldsymbol{d}_k^\intercal \boldsymbol{d}_i, \quad d_{kj} = \boldsymbol{d}_k^\intercal \boldsymbol{d}_i, \\ \kappa_2 = \frac{d_{jj} d_{ki} - d_{ij} d_{kj}}{d_{ii} d_{jj} - d_{ij} d_{ij}}, \quad \kappa_3 = \frac{d_{ii} d_{kj} - d_{ij} d_{ki}}{d_{ii} d_{jj} - d_{ij} d_{ij}}, \quad \kappa_1 = 1 - \kappa_2 - \kappa_3. \end{gathered} \tag{12}$$

The barycenteric coefficients κ_1, κ_2, and κ_3 are in the range of $[0, 1]$ if the grid point (x, y) is in the triangle.

A.4. Wrapping Function

The wrapping function $\Psi : \boldsymbol{A} \in \mathbb{R}^{256\times256\times3} \rightarrow \boldsymbol{R} \in \mathbb{R}^{K\times3}$ is defined as $\boldsymbol{R}(i) = \boldsymbol{A}(\boldsymbol{U}(i,1), \boldsymbol{U}(i,2))$, where i is the index for the vertices of a 3D face. $\boldsymbol{R}(i)$ and $\boldsymbol{A}(\boldsymbol{U}(i,1), \boldsymbol{U}(i,2))$ are 3-dimensional vectors. $\boldsymbol{U} \in \mathbb{R}^{K\times2}$ is the coordinates of shape in UV space from 3DMM [4]. Again, bilinear interpolation is used if $\boldsymbol{U}(i,1)$ or $\boldsymbol{U}(i,2)$ is not an integer.

B. Experiments

B.1. Network Architecture

We use standard encoder networks for viewpoint, shape and illumination predictions, and a network similar to U-Net [30] for reflectance prediction. The detailed configurations are given in Table 1. Parameter d is 7 for viewpoint network f_v and 9 for illumination network f_ℓ. Conv $3_{/2,1}$ denotes convoluitonal layer with kernel size of 3, where the stride and padding are 2 and 1, respectively. Each convolutional layer is followed by a Batch Normalization (BN) [15] layer and Rectified Linear Units (ReLU). Bilinear interpolation is adopted for the upsampling operation. Specifically, in Table 1, the layers in brackets are residual blocks. In Table 2, we use shortcut to connect the feature maps of encoder and decoder, but different from U-Net, we use addition rather than concatenation to integrate information in the feature maps. For those encoder output shapes in brackets (*e.g.*, "$[128 \times 128 \times 64]$"), the feature map will be added as a shortcut to the decoder feature map (also with the same brackets).

Viewpoint & Illumination Network			Shape Network		
Layer	Act.	Output shape	Layer	Act.	Output shape
Input	-	$256 \times 256 \times 3$	Input	-	$256 \times 256 \times 3$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$128 \times 128 \times 32$	Conv $4 \times 4_{/2,1}$	BN + ReLU	$128 \times 128 \times 64$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$64 \times 64 \times 32$	Conv $4 \times 4_{/2,1}$	BN + ReLU	$64 \times 64 \times 64$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$32 \times 32 \times 64$	Conv $4 \times 4_{/2,1}$	BN + ReLU	$32 \times 32 \times 128$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$16 \times 16 \times 64$	Conv $4 \times 4_{/2,1}$	BN + ReLU	$16 \times 16 \times 128$
$\begin{bmatrix}\text{Conv3} \times 3_{/1,1} \\ \text{Conv3} \times 3_{/1,1}\end{bmatrix}$	BN + ReLU BN + ReLU	$16 \times 16 \times 64$ $16 \times 16 \times 64$	$\begin{bmatrix}\text{Conv3} \times 3_{/1,1} \\ \text{Conv3} \times 3_{/1,1}\end{bmatrix}$	BN + ReLU BN + ReLU	$16 \times 16 \times 128$ $16 \times 16 \times 128$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$8 \times 8 \times 128$	Conv $4 \times 4_{/2,1}$	BN + ReLU	$8 \times 8 \times 256$
$\begin{bmatrix}\text{Conv3} \times 3_{/1,1} \\ \text{Conv3} \times 3_{/1,1}\end{bmatrix}$	BN + ReLU BN + ReLU	$8 \times 8 \times 128$ $8 \times 8 \times 128$	$\begin{bmatrix}\text{Conv3} \times 3_{/1,1} \\ \text{Conv3} \times 3_{/1,1}\end{bmatrix}$	BN + ReLU BN + ReLU	$8 \times 8 \times 256$ $8 \times 8 \times 256$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$4 \times 4 \times 128$	Conv $4 \times 4_{/2,1}$	BN + ReLU	$4 \times 4 \times 256$
Conv $4 \times 4_{/2,1}$	-	$1 \times 1 \times d$	Conv $4 \times 4_{/2,1}$	-	$1 \times 1 \times 228$

Table 1: The detailed CNNs architectures of viewpoint, illumination, and shape networks.

Reflectance Network					
U-Net Encoder (↓)			U-Net Decoder (↑)		
Encoder Layer	Act.	Output shape	Decoder Layer	Act.	Output shape
Input	-	$256 \times 256 \times 3$	Output	-	$256 \times 256 \times 3$
-	-	-	Conv $3 \times 3_{/1,1}$	Tanh	$256 \times 256 \times 3$
-	-	-	Conv $3 \times 3_{/1,1}$	BN + ReLU	$256 \times 256 \times 3$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$128 \times 128 \times 64$	Upsample ($2\times$)	-	$256 \times 256 \times 64$
Conv $3 \times 3_{/1,1}$	BN + ReLU	$[128 \times 128 \times 64]$	Conv $3 \times 3_{/1,1}$	BN + ReLU	$[128 \times 128 \times 64]$
-	-	-	Conv $3 \times 3_{/1,1}$	BN + ReLU	$128 \times 128 \times 64$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$64 \times 64 \times 64$	Upsample ($2\times$)	-	$128 \times 128 \times 64$
Conv $3 \times 3_{/1,1}$	BN + ReLU	$[64 \times 64 \times 64]$	Conv $3 \times 3_{/1,1}$	BN + ReLU	$[64 \times 64 \times 64]$
-	-	-	Conv $3 \times 3_{/1,1}$	BN + ReLU	$64 \times 64 \times 64$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$32 \times 32 \times 128$	Upsample ($2\times$)	-	$64 \times 64 \times 128$
Conv $3 \times 3_{/1,1}$	BN + ReLU	$[32 \times 32 \times 128]$	Conv $3 \times 3_{/1,1}$	BN + ReLU	$[32 \times 32 \times 128]$
-	-	-	Conv $3 \times 3_{/1,1}$	BN + ReLU	$32 \times 32 \times 128$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$16 \times 16 \times 128$	Upsample ($2\times$)	-	$32 \times 32 \times 128$
Conv $3 \times 3_{/1,1}$	BN + ReLU	$[16 \times 16 \times 128]$	Conv $3 \times 3_{/1,1}$	BN + ReLU	$[16 \times 16 \times 128]$
-	-	-	Conv $3 \times 3_{/1,1}$	BN + ReLU	$16 \times 16 \times 128$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$8 \times 8 \times 256$	Upsample ($2\times$)	-	$16 \times 16 \times 256$
Conv $3 \times 3_{/1,1}$	BN + ReLU	$[8 \times 8 \times 256]$	Conv $3 \times 3_{/1,1}$	BN + ReLU	$[8 \times 8 \times 256]$
Conv $4 \times 4_{/2,1}$	BN + ReLU	$4 \times 4 \times 256$	Conv $3 \times 3_{/1,1}$	BN + ReLU	$8 \times 8 \times 256$
Conv $3 \times 3_{/1,1}$	BN + ReLU	$4 \times 4 \times 256$	Upsample ($2\times$)	-	$8 \times 8 \times 256$

Table 2: The detailed CNNs architectures of reflectance networks. Note that, the layers in the decoder (from input to output) are listed from bottom to top.

B.2. More Ablation Studies

We perform more ablations for different settings of CEST. We explore the averaged representations, an approach adopted in [37], for reflectance consistency, where the averaged reflectance of a video clip is used to reconstruct the 3D face in each video frame. Here, we fix the size of minibatch, *i.e.* 128, but vary the number of images from each video clip to 2, 4, and 8.

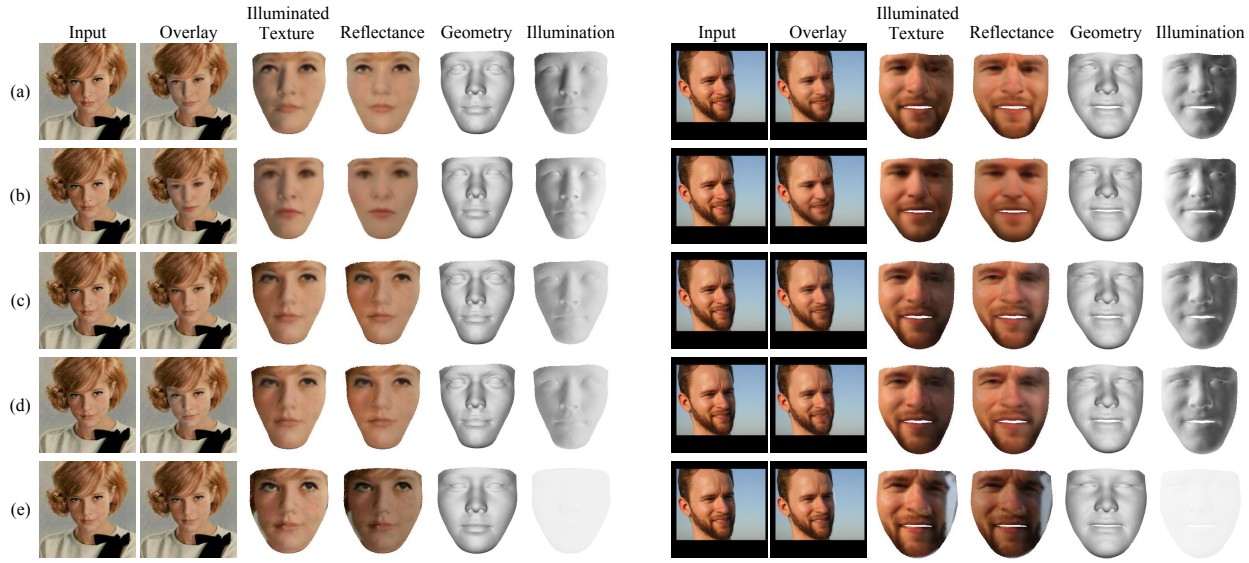

Figure 10: Ablations. (a) CEST with default settings. (b), (c) and (d) Averaged reflectance is used in training and the number of images from each video clips are 2, 4, and 8, respectively.

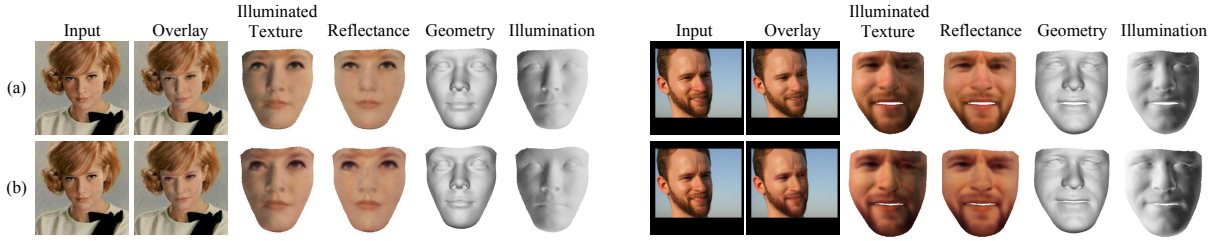

Figure 11: Ablations. (a) CEST with default settings. (b) Reflectance consistency is applied to videos, not video clips.

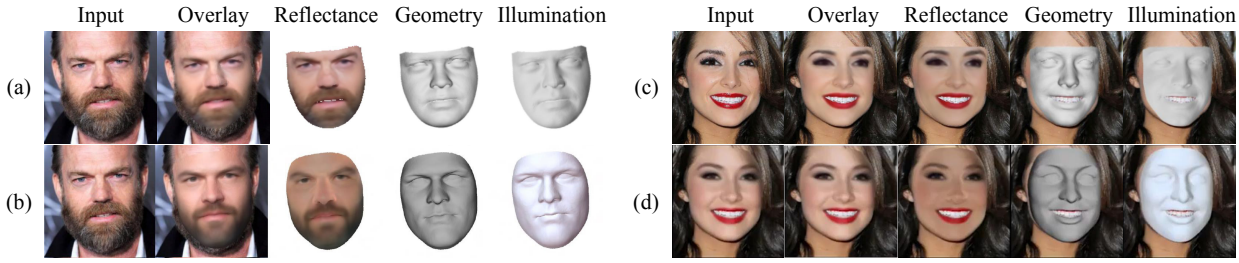

Figure 12: Comparisons to [38]. (a) and (c) Results from CEST. (b) and (d) Results from [38].

Results are shown in Fig. 10 (b), (c), and (d), respectively. As we can see, there are still some illumination in the reflectance, indicating that the averaged representation may not be a good strategy for learning disentangled facial parameters.

Fig. 11 shows the results from CEST trained with reflectance consistency across video. The performance is comparable to those from CEST trained with default setting (reflectance consistency across video clip). It shows that consistency constraint can be generalized to longer videos if the recording environments are not changed dramatically.

B.3. More Qualitative Comparisons

In this section, we show more comparisons to the state-of-art methods [5, 32, 29, 40]. Since there is no publicly available implementations for these methods, we compare to the results presented in their papers.

Overall, CEST produces more stable and reasonable geometries, detailed reflectances, and realistic reconstructions of the

3D faces. As shown in Fig. 12 (a) (b), Fig. 15, Fig. 16, and Fig. 17, the facial shapes predicted by CEST are more accurate in facial expressions and lip closure. In addition, the predicted reflectances show more personal characteristics, but less remaining illumination, as illustrated in Fig. 13 and Fig. 16. Lastly, CEST yields faithful 3D reconstructions, capturing more details than the other methods (see Fig 14 and Fig 15).

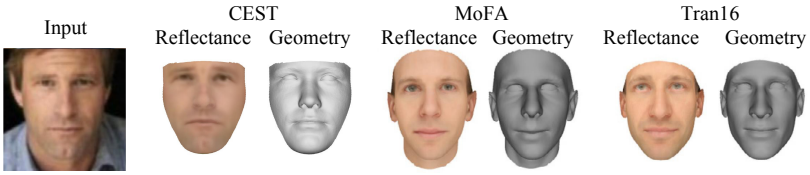

Figure 13: Comparisons to MoFA [39] and [43].

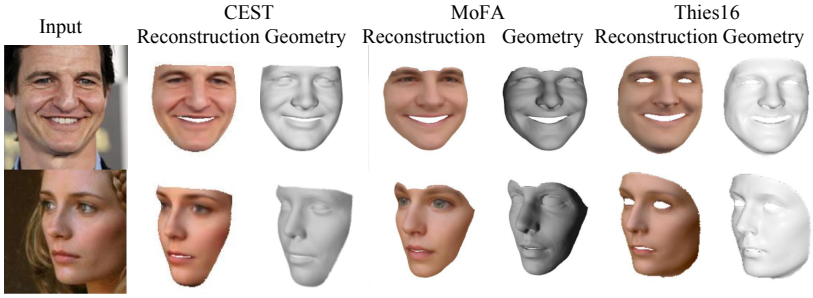

Figure 14: Comparisons to MoFA [39] and [40].

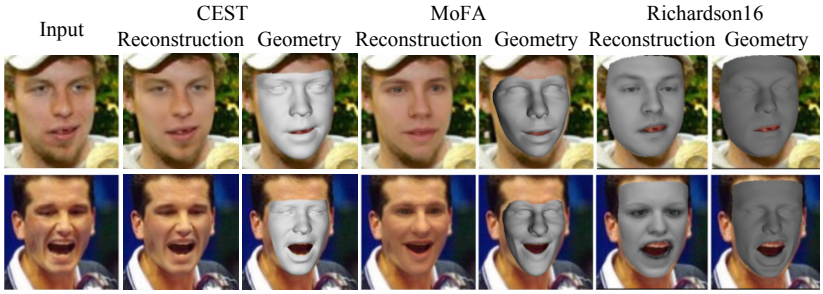

Figure 15: Comparisons to MoFA [39] and [29]. Our estimated shapes show more accurate expressions.

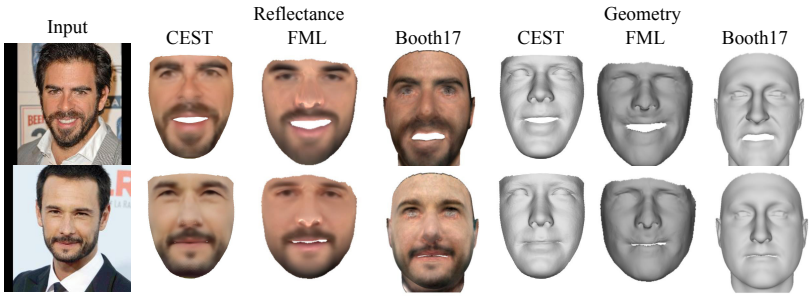

Figure 16: We compare CEST to FML [37] and [5].

B.4. Challenging Cases

We present some examples with dark skin in Fig. 18. Although most people in the training set (VoxCeleb) are Caucasian, CEST still produces reasonable illumination and albedo for these examples. One limitation is that the reconstruction of the non-lambertian surface is inaccurate, e.g. eyes with unusual gaze directions.

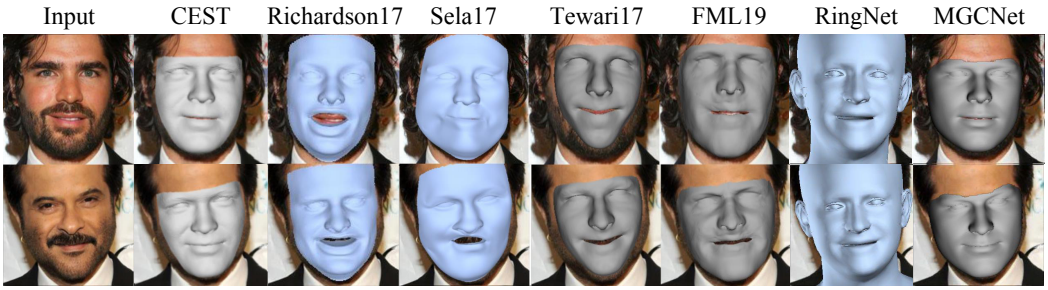

Figure 17: We compare the estimated shapes from CEST to those from [29], [32], [39], [37], [31], and [34] (from left to right). Our estimated shapes are more stable and accurate.

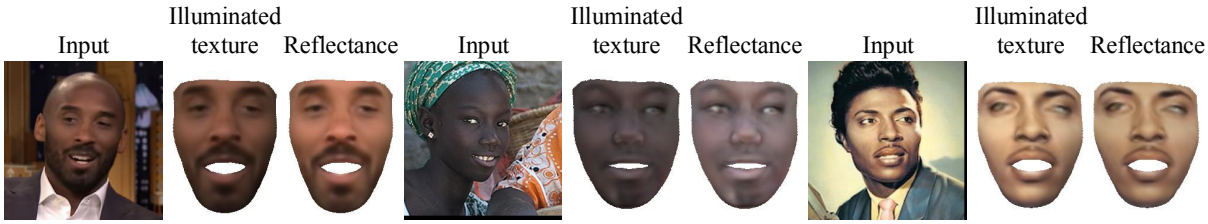

Figure 18: Some challenging examples.

B.5. Photometric Error

We compare CEST, IEST, FML [37] and Garrido [12] on overlay face reconstruction. To measure the quality of the overlay images, we compute the average photometric error (R,G,B pixel values are from 0 to 255) between the input face image and the overlay face image. We experiment on 1,000 images in CelebA dataset [25]. Table 3 shows that the conditional estimation is beneficial for reconstructing the 3D face, and the proposed CEST outperforms existing methods by a large margin.

Method	**CEST**	IEST	FML [37]	Garrido16 [12]
Photometric Error	**10.74**	13.76	20.65	21.95

Table 3: Photometric errors obtained by different methods.