Learning Formation of Physically-Based Face Attributes Supplementary Material

Ruilong Li[1,2*] Karl Bladin[1*] Yajie Zhao[1*] Chinmay Chinara[1] Owen Ingraham[1]
Pengda Xiang[1,2] Xinglei Ren[1] Pratusha Prasad[1] Bipin Kishore[1] Jun Xing[1] Hao Li [1,2,3]

[1]*USC Institute for Creative Technologies* [2]*University of Southern California* [3]*Pinscreen*

1. Experiment Details

1.1. Gender Control

Step1. Pre-computing mean gender latent code. First, we propose a classifier ψ, trained with ground truth data to classify our input pair (*albedo* and *geometry* maps) into two categories (*male* and *female*). Then we randomly sample $Z_{id} \sim N(\mu_{id}, \sigma_{id})$ to generate $10k$ sample pairs $G_{id}(Z_{id})$ using our *identity network*. The classifier separates all the samples into two groups. Finally, we extract the mean vector of each category as Z_{male} and Z_{female} using equation 1.

$$Z_{mean} = \frac{1}{\sum_{i=1}^{10k} \Omega(Z_{id}^{(i)})} \sum_{i=1}^{10k} Z_{id}^{(i)} \cdot \Omega(Z_{id}^{(i)}) \tag{1}$$

Where $\Omega(Z_{id})$ is the gender activation function which converts the outputs of gender classifier ψ into binary values defined as follows:

$$\Omega(Z_{id}) = \begin{cases} 1, \psi(G_{id}(Z_{id})) <= 0.5 \\ 0, \psi(G_{id}(Z_{id})) > 0.5 \end{cases} \tag{2}$$

Where $\Omega(Z_{id}) = 1$ is defined to be female, and $\Omega(Z_{id}) = 0$ means male. In equation 1, the mean vector in each category Z_{male} and Z_{female} is computed by simply averaging the samples where $\Omega(Z_{id}^{(i)})$ equals to 1 and 0 separately.

Step2. Conditioned Generation. Instead of directly using a randomly sampled $Z_{id} \sim N(\mu_{id}, \sigma_{id})$ as input, we combine it with the mean gender latent code Z_{male} and Z_{female}:

$$Z_{id}^{gender} = (1-\alpha-\beta)\times Z_{id}+\alpha\times Z_{male}+\beta\times Z_{female} \tag{3}$$

*Joint first authors

We can set $\alpha = 0.5, \beta = 0.0$ to ensure generated results are all male, or $\alpha = 0.0, \beta = 0.5$ to ensure generated results are all female. We can also gradually decrease α and increase β at the same time to interpolate a male generation into female. An example of this is shown in Fig.9 of the paper.

1.2. Age Control

The main idea of age control is similar to the gender control (Sec 1.1) with two main differences: (1) Instead of a classifer ψ for gender classification, we use a regressor ϕ to predict the true age (in years). (2) We compute an average vector for Z_{old} and Z_{young} separately using the method of sampling Z_{id} with $\phi(G_{id}(Z_{id})) > 50$ and $\phi(G_{id}(Z_{id})) < 30$. So the final age latent code is represented as:

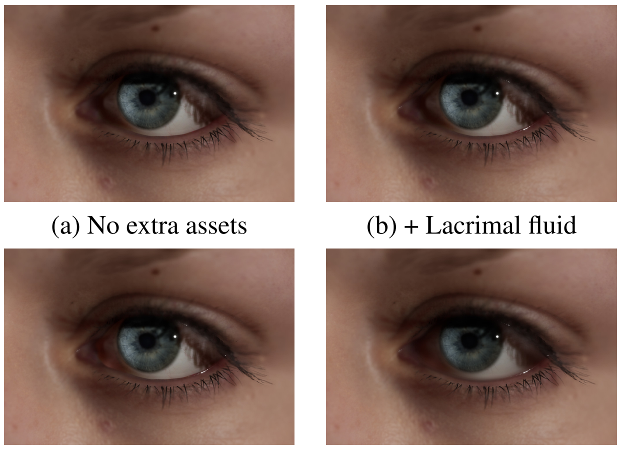
$$Z_{id}^{age} = (1-\alpha-\beta)\times Z_{id}+\alpha\times Z_{old}+\beta\times Z_{young} \tag{4}$$

Figure 9 in the main paper also shows a example of aging interpolation by gradually increasing α from 0.0 to 0.7, and decreasing β from 0.7 to 0.0.

1.3. 3D Model Fitting

Given a face scan, or face model, we firstly convert it into our albedo and geometry map format by fitting a linear face model followed by Laplacian warping and attribute transfer. The ground truth latent code of the input is denoted Z_{id}. Our goal of fitting is to find the latent code $Z_{id}^{'}$ that best approximates Z_{id} while retaining the embodyment of our model. To achieve this, one can find $Z_{id}^{'}$ that minimizes $MSE(G_{id}(Z_{id}^{'}), G_{id}(Z_{id}))$ through gradient descent.

In particular, we first use the *Adam* optimizer with a constant $learningrate = 1.0$ to update the input variable $Z_{id}^{'}$, then we update the variables in the *Noise Injection* Layers with $learningrate = 0.01$ to fit those details. Fig.10 in the paper shows the geometry of the fitting results.



(a) No extra assets (b) + Lacrimal fluid

(c) + Blend mesh (d) + Occlusion mesh

Figure 1: Closeup of real time rendered eye with our model's additional eye geometries successively added. The eyeball and eyelashes are considered as default eye geometry and therefore kept in all subfigures.

1.4. Low-quality Data Enhancement.

In order to enhance the quality of low-resolution data, so that it can be better utilized, the data point needs to be encoded as Z_{id} in our latent space. This is done using our fitting method 1.3. The rest of the high fidelity assets are generated using our generative pipeline. Unlike the fitting procedure, we don't want *true-to-groundtruth* fitting which would result in a recreation of a low resolution model. We instead introduce a discriminator loss to balance the *MSE* loss. This provides an additional constraint on reality and quality during gradient descent. Empirically we give a 0.001 weight to the discriminator loss to balance the *MSE* loss. We also use the *Adam* optimizer with a constant $learning - rate = 1.0$ for this experiment. The attained variable $Z_{id}^{'}$ is then fed in as the new input, and the process is iteratively repeated until convergence after about 4000 iterations.

2. Real Time Rendering Assets

To demonstrate the use of additional eye rendering assets (lacrimal fluid, blend mesh, and eye occlusion) available in our model, we show a real time rendering of a close up of an eye and its surrounding skin geometry and material from scan data in Figure 1. The rendering is performed using Unreal Engine 4. Materials and shaders are adopted from the *Digital Human* project [1].

References

[1] Unreal Engine - Digital Human. https://docs.unrealengine.com/en-US/Resources/Showcases/DigitalHumans/index.html. Online; Accessed: 2020-03-29. 2