Deep Shapely Portraits - Supplementary Material

A WARPING OPTIMIZATION

In order for the deformed face to fit the warped background as good as possible, we optimize the control grid such that the background region warps outward/inward with the same proportion as the face region. More specifically, we construct a grid mesh M on the original image plane and optimally adjust M to warp the background region. The optimal solution $\boldsymbol{V}^* = \{\boldsymbol{v}_i^*\}$ is:

$$\boldsymbol{V}^* = \arg\min_{\boldsymbol{v}_i} E(\boldsymbol{V}), \tag{1}$$

where $\boldsymbol{v}_i$ is the i-th control point of mesh M, and E is the weighted sum of several energy terms. Similar to [4], we use line-bending term, regularization term, and border term, to make sure the background changes smoothly without much distortion. In addition, we add a new fitting term to optimize the consistency between the reshaped face and the background as follows.

Fitting Term. The boundary between the face region of the original image and the background is across a set of (boundary) control points with indices in a set C. These points should move according to the reshaped 3D face. After reshaping the face model, we calculate the new positions of the boundary pixels by projecting the reshaped face boundary back to 2D, which in turn determines the target positions of (boundary) control points. The fitting term E_f drives all control points in C towards the target positions, and is defined as:

$$E_f = \sum_{i \in C} ||\mathbf{u}_i - \boldsymbol{v}_i||^2, \tag{2}$$

where $\mathbf{u}_i$ is the target position of the control point $\boldsymbol{v}_i$.

Non-distortion terms. The movements of control points will cause obvious distortions and jitters of the background region. And when a face gets thinner or fatter, the border of the image is liable to shrink or expand accordingly. We encourage the output mesh to keep lines straight and smooth by adding the non-distortion terms below, similar as the work of [4]:

$$E_s = w_l E_l + w_r E_r + w_b E_b, \tag{3}$$

where w_l is the weight of the term E_l, which preserves the straight line, w_r is the weight of E_r for optimizing the mesh to be smooth, and w_b is the weight of E_b for keeping image border from shrinking or expanding. E_l,E_r are described as follows:

$$E_l = \sum_i \sum_{j \in \mathbf{N}(i)} ||(\boldsymbol{v}_i - \boldsymbol{v}_j) \times \mathbf{e}_{ij}||_2^2, \tag{4}$$

$$E_r = \sum_i \sum_{\mathbf{N}(i)} ||\boldsymbol{v}_i - \boldsymbol{v}_j||_2^2, \tag{5}$$

where $\mathbf{N}(i)$ is the set of 4-way adjacent points of control point $\boldsymbol{v}_i$, $\mathbf{e}_{ij}$ is the unit vector along the direction $\boldsymbol{v}_i - \boldsymbol{v}_j$, and the symbol $\times$ denotes the cross product. And we use the distance between grid points on border , which are initially on the image borders and the image borders for calculating E_b:

$$E_b = \sum_{i \in \mathbf{b}} ||d||^2. \tag{6}$$

where $\mathbf{b}$ is the set of grid border points, and d is the distance between the i-th point and the border. We combine all the equations to generate the final energy function:

$$E = w_c E_f + w_l E_l + w_r E_r + w_b E_b \tag{7}$$

where w_c, w_l, w_r, w_b are the weights for the corresponding energy terms, and they are set to 4, 2, and 0.5, 1, respectively.

We use the Levenberg-Marquardt algorithm implemented in the Ceres Solver [1] for warping optimization. It is easy to see that our reshaping process mainly warps the pixels of a face along the horizontal direction. Therefore, we set the grid to 90×75 for better control.

To accelerate the warping optimization, we initialize the position of a grid point $\boldsymbol{v}_i$ based on the distance between its initial position and the face boundary points. More specifically, we calculate an influence factor w_j of the j-th boundary point to $\boldsymbol{v}_i$ using a Gaussian kernel:

$$w_j = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d^2}{2\sigma^2}}, \tag{8}$$

where σ is the variance of the kernel (empirically set to 0.1 in our experiments), and d is the distance between $\boldsymbol{v}_i$ and the j-th boundary point. We normalize d using the length of the warping grid. $\boldsymbol{v}_i$ is initialized by the following equation in the optimization:

$$\boldsymbol{v}_i = \boldsymbol{v}_i^0 + \frac{\sum_j w_j \boldsymbol{s}_j}{\sum_j w_j}, \tag{9}$$

where $\boldsymbol{v}_i^0$ is the initial position of the grid point, and $\boldsymbol{s}_j$ is the offset of the j-th boundary point. To further accelerate the optimization, we remove the grid points covered by the rendered face, which consequently reduces the number of parameters to be optimized.

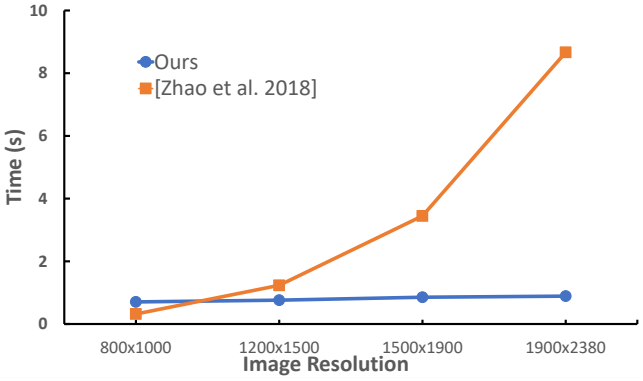


Figure S1: Comparison on computational performance with different image resolutions. The cost for our warping optimization (about 0.8 seconds) is almost the same with increasing image sizes, while for [5], it is quite high for images with 4 megapixels (about 8.6 seconds), and even fails over 4 megapixels because of memory overflow.

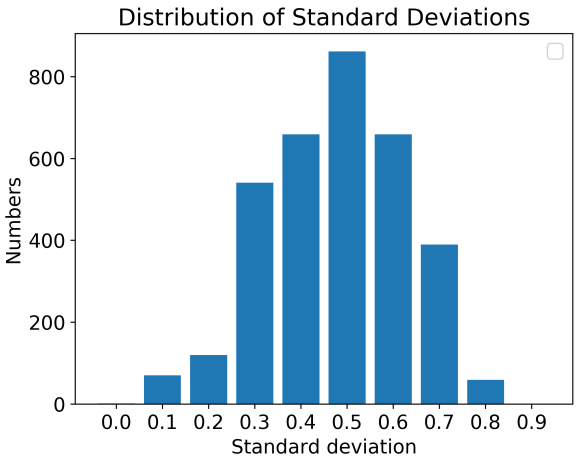


Figure S2: Distribution of standard deviations of all collected shapely level distributions of SHAPEFACENET.

This reduces the optimization time by 20%. With all the above accelerations, the optimization can run eight times faster.

B SHAPEFACENET

B.1 Portrait Images Collection

The SHAPEFACENET Dataset contains 27,200 portrait images, including 3,400 individuals (8 images with different shapely degrees for each individual) consisting of 1,620 and 1,780 examples selected from FFHQ dataset [2] and CelebA dataset [3], respectively. It contains 1,500 females and 1,900 males. We randomly select 2,800, 300, 300 individuals as a training set, validation set, and testing set.

B.2 Shapely Degrees

All the individuals are reshaped with face roundness ranging from {-2.0, -1.6, -1.2, -0.8, -0.4, 0.0, 0.4, 0.8}. Then we developed a web-based system that allows the raters to choose the most shapely one from a set of reshaped faces of each individual. The voting system was deployed on the internet, and the voting tasks are distributed to each rater in crowdsourcing manners. 3,400 Asians aging from 18 to 35 are asked to rate SHAPEFACENET. Each rater rates 20 individuals from SHAPEFACENET. Finally, we collect 20 labels rated by 20 different raters to form a shapely level distribution for each image. Figure S2 present the distribution of standard deviations of all collected shapely level distributions of SHAPEFACENET. For each individual, we chose the most voted one as the most shapely face. Then we can obtain the *shapely degree* of a face by computing the δBMI difference of the current face to the most shapely face.

REFERENCES

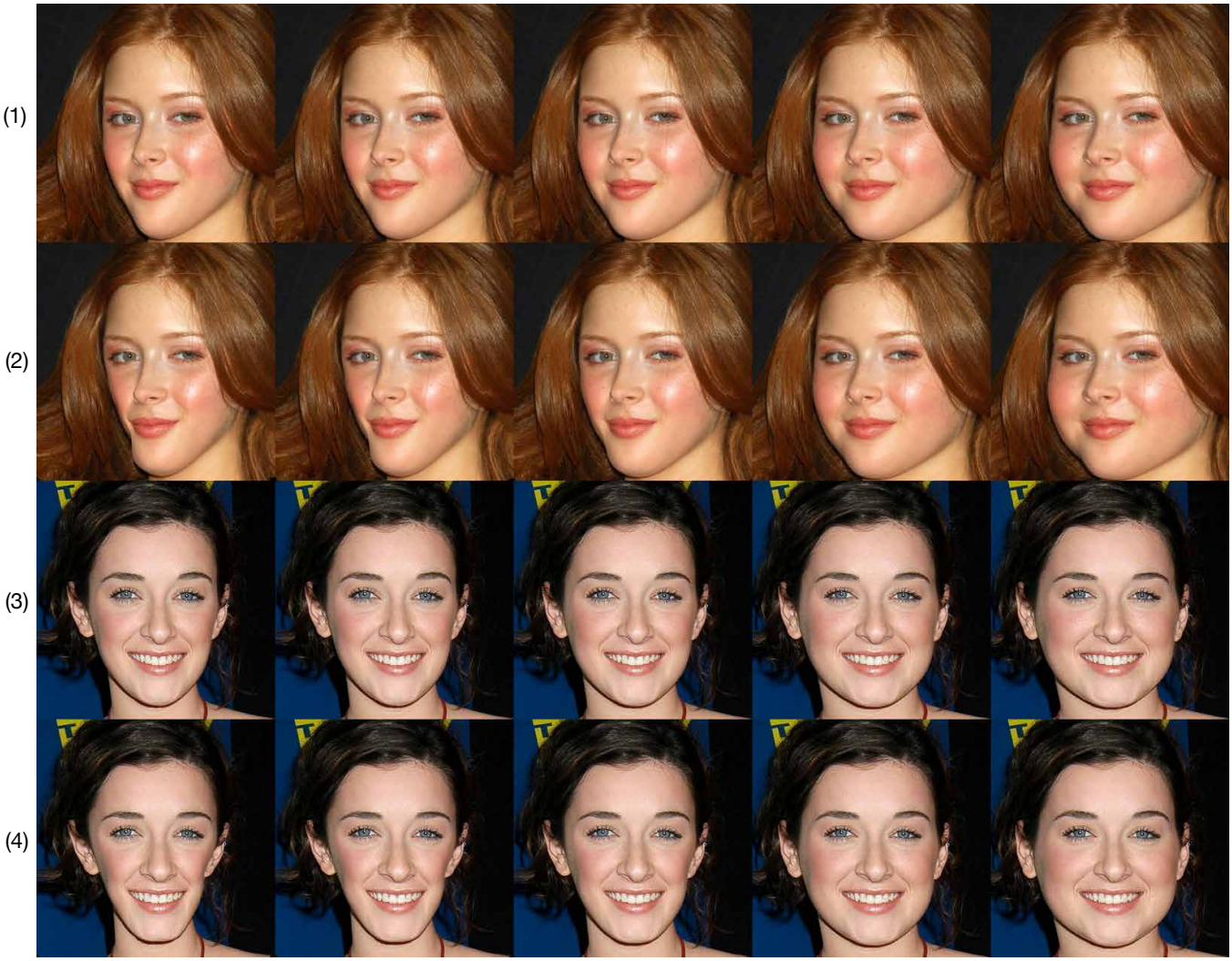
[1] Sameer Agarwal, Keir Mierle, and Others. 2012. Ceres Solver. http://ceres-solver.org.

[2] Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4401–4410.

[3] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[4] YiChang Shih, Wei-Sheng Lai, and Chia-Kai Liang. 2019. Distortion-Free Wide-Angle Portraits on Camera Phones. *ACM Transactions on Graphics* 38, 4, Article 61 (2019), 12 pages. https://doi.org/10.1145/3306346.3322948

[5] Haiming Zhao, Xiaogang Jin, Xiaojian Huang, Menglei Chai, and Kun Zhou. 2018. Parametric Reshaping of Portrait Images for Weight-Change. *IEEE Computer Graphics and Applications* 38, 1 (2018), 77–90. https://doi.org/10.1109/MCG.2018.011461529



(a) $\lambda = -2.4$ (b) $\lambda = -1.8$ (c) $\lambda = -0.8$ (d) $\lambda = 0.0$ (e) $\lambda = 0.8$

Figure S3: Comparison results between our reshaping model (1, 3) and [5] (2, 4) using the same reshaping parameter. Our results are natural and plausible compared to the prior work with noticeable artifacts.