Supplementary Material for Synthesizing Normalized Faces from Facial Identity Features

Forrester Cole[1] David Belanger[1,2] Dilip Krishnan[1] Aaron Sarna[1] Inbar Mosseri[1] William T. Freeman[1,3]
[1]Google, Inc. [2]University of Massachusetts Amherst [3]MIT CSAIL
{fcole, dbelanger, dilipkay, sarna, inbarm, wfreeman}@google.com

1. Additional Results

Figures 2 and 3 contain additional results of face normalization on LFW and comparison to Hassner et al. [1].

Figure 4 show results from degraded photographs and illustrations, which push the method outside of its training domain but still produce credible results.

2. 3-D Model Fitting

To fit the shape of the face, we first manually establish a correspondence between the 65 predicted landmarks $\mathbf{l}_i$ and the best matching 65 vertices $\mathbf{v}_i$ of the 3-D mesh used to train the model of Blanz and Vetter [2]. This correspondence is based on the semantics of the landmarks and does not change for different faces. We then optimize for the shape parameters that best match $\mathbf{v}_i$ to $\mathbf{l}_i$ using gradient descent. The landmarks provide $65 \times 2 = 130$ constraints for the 199 parameters of the morphable model, so the optimization is additionally regularized towards the average face.

Once the face mesh is aligned with the predicted landmarks, we project the synthesized image onto the mesh as vertex colors. The projection works well for areas that are close to front-facing, but is noisy and imprecise at grazing angles. To clean the result, we project the colors further onto the model's texture basis to produce clean, but less accurate vertex colors. We then produce a final vertex color by blending the synthesized image color and the texture basis color based on the foreshortening angle.

2.1. Corresponding Landmarks and Vertices

As a pre-processing step, we determine which 65 vertices of the shape model's mesh best match the 65 landmark positions. Since the topology of the mesh doesn't change as the shape changes, the correspondence between landmark indices and vertex indices is fixed.

The correspondence could be determined completely manually, but we choose to find the it automatically by rendering the mean face and extracting landmarks from the rendered image (Fig 1).

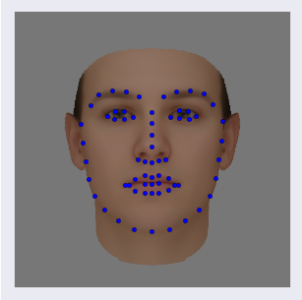

Figure 1. Landmarks extracted from the mean face of the Blanz and Vetter model.

The corresponding vertex for each landmark is found by measuring screen-space distance between the computed landmarks and the projected vertices. This projection is noisy around grazing angles and may pick back-facing vertices or other poor choices. To make the correspondence cleaner, we compute the correspondences separately for multiple, randomly jittered camera matrices, then use voting to determine the most stable matching vertex for each landmark. The final result is a set of 65 vertex indices.

2.2. Shape Fitting

Given a set of 65×2 matrix of landmark points L, our goal is to optimize for the best matching set of 199 shape coefficients $\mathbf{s}$. To find $\mathbf{s}$, we imagine that the landmarks L are the projection of their corresponding vertices V, where the 65×2 matrix V is defined by the shape parameters $\mathbf{s}$, a translation vector $\mathbf{t}$, a uniform scaling factor σ, and a fixed projection matrix P, as follows.

Let the 65×3 matrix of object-space vertex positions V_w be:

$$V_w = \begin{bmatrix} B^x\mathbf{s} & B^y\mathbf{s} & B^z\mathbf{s} \end{bmatrix} + \mu \qquad (1)$$

where $B^{x,y,z}$ are the 65×199 morphable model basis matrices and μ is the 65×3 matrix of mean vertex positions.

The 4×4 projection matrix P is a perspective projection with a field of view of 10° to roughly match the perspective

of the training images. The 4×4 modelview matrix M is defined by the translation $\mathbf{t}$ and scaling σ as:

$$M = \begin{bmatrix} \sigma & 0 & 0 & t^x \\ 0 & \sigma & 0 & t^y \\ 0 & 0 & \sigma & t^z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Given P and M, the 65×4 matrix of post-projection vertices V_p is defined as:

$$V_p = \begin{bmatrix} V_w & \mathbf{1} \end{bmatrix} M^T P^T \tag{3}$$

and the final, 65×2 vertex position matrix V is found by perspective division:

$$V = \begin{bmatrix} \frac{\mathbf{x}_p}{\mathbf{w}_p} & \frac{\mathbf{y}_p}{\mathbf{w}_p} \end{bmatrix} \tag{4}$$

where $\mathbf{x}_p$, $\mathbf{y}_p$, and $\mathbf{w}_p$ are first, second, and fourth columns of V_p.

Finally, we optimize for $\mathbf{s}$ using gradient descent with the loss function:

$$f(\mathbf{s}) = \|L - V\|^2 + \lambda \|\mathbf{s}\|^2 \tag{5}$$

where length term for $\mathbf{s}$ regularizes the optimization towards the mean face (i.e., $\mathbf{s} = \mathbf{0}$), and $\lambda = 0.001$ in our experiments.

2.3. Fitting Texture

Once the shape parameters and pose of the model are found, we project the remaining ≈ 53K vertices of the mesh onto the synthesized face image. The projection produces a $53K \times 3$ matrix of vertex colors C_p.

Due to noise in the synthesized image and the inherent inaccuracy of projection at grazing angles, the colors C_p have ugly artifacts. To repair the artifacts, we compute a confidence value α_i at each vertex that downweights vertices outside the facial landmarks and vertices at grazing angles:

$$\alpha_i = m(x_i, y_i)(1.0 - n_i^z) \tag{6}$$

where m is a mask image that is 1 inside the convex hull of the landmark points and smoothly decays to 0 outside, and n_i^z is the z component of the i^{th} vertex normal.

Using the confidences, we project the vertex colors C_p onto the morphable model color basis. Let $\mathbf{c}_p$ be the 160K vector produced by flattening C_p, $\mathbf{a}$ be the 160K vector produced by repeating the confidences α_i for each color channel, and A be the $160K \times 199$ matrix of confidences produced by tiling $\mathbf{a}$. The 199 color parameters $\mathbf{z}$ are found by solving an over-constrained linear system in the least-squares sense:

$$\begin{bmatrix} (B \circ A) \\ \lambda I \end{bmatrix} \mathbf{z} = \begin{bmatrix} (\mathbf{c_p} - \mu) \circ \mathbf{a} \\ \mathbf{0} \end{bmatrix} \tag{7}$$

where $\circ$ represents the element-wise product, B is the $160K \times 199$ color basis matrix, I is the identity matrix, μ is the model's mean color vector, and λ is a regularization constant.

The flattened model color vector $\mathbf{c}_b$ is found by unprojecting $\mathbf{z}$:

$$\mathbf{c}_b = B^T \mathbf{z} + \mu \tag{8}$$

and the final flattened color vector $\mathbf{c}$ is defined by interpolating between the projected and model colors:

$$\mathbf{c} = \mathbf{c}_p \circ \mathbf{a} + \mathbf{c}_b \circ (\mathbf{1} - \mathbf{a}) \tag{9}$$

3. Automatic Photo Adjustment

Let $\mathbf{m}_P$ and $\mathbf{m}_N$ be the mean face colors for the input and normalized images, respectively. Our adjusted image is computed using a per-channel, piecewise-linear color shift function $r^c(\mathbf{p})$ over the pixels of P:

$$r^c(\mathbf{p}) = \left\{ \begin{array}{rll} \mathbf{p}^c \frac{\mathbf{m}_N^c}{\mathbf{m}_P^c} & \text{if} & \mathbf{p}^c <= \mathbf{m}_P^c \\ 1 - (1 - \mathbf{p}^c) \frac{1 - \mathbf{m}_N^c}{1 - \mathbf{m}_P^c} & \text{if} & \mathbf{p}^c > \mathbf{m}_P^c, \end{array} \right\} \tag{10}$$

where c are the color channels. We chose YCrCb as the color representation in our experiments.

References

[1] T. Hassner, S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4295–4304. 1, 3, 4

[2] 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*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194. 1

[3] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007. 3

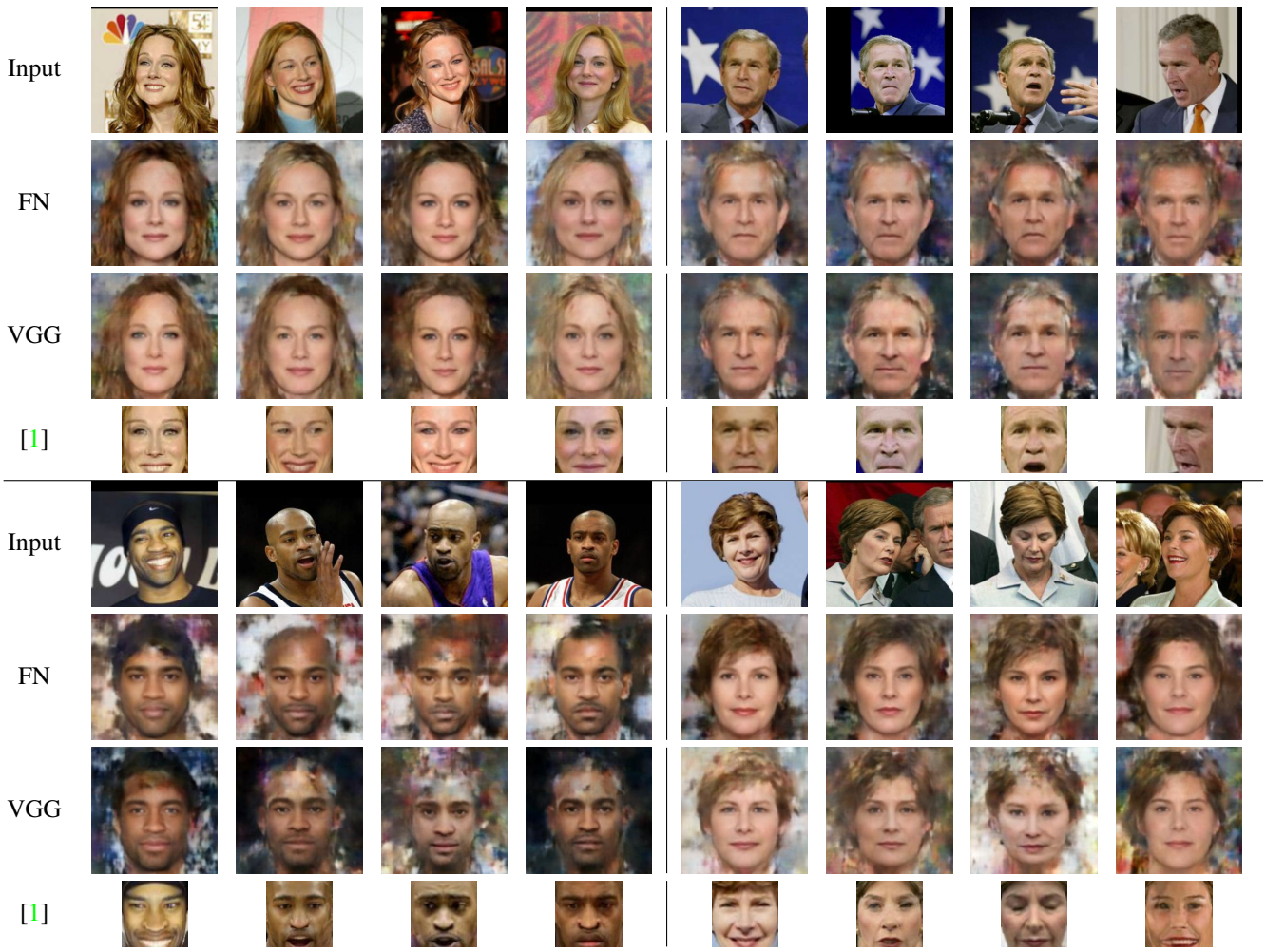

Figure 2. Additional face normalization results for the LFW dataset [3]. Top: input photographs. Middle: result of our method for FaceNet "avgpool-0" and VGG-Face "fc7" features. Bottom: result of Hassner et al. [1].

Figure 3. Additional face normalization results similar to Fig. 2

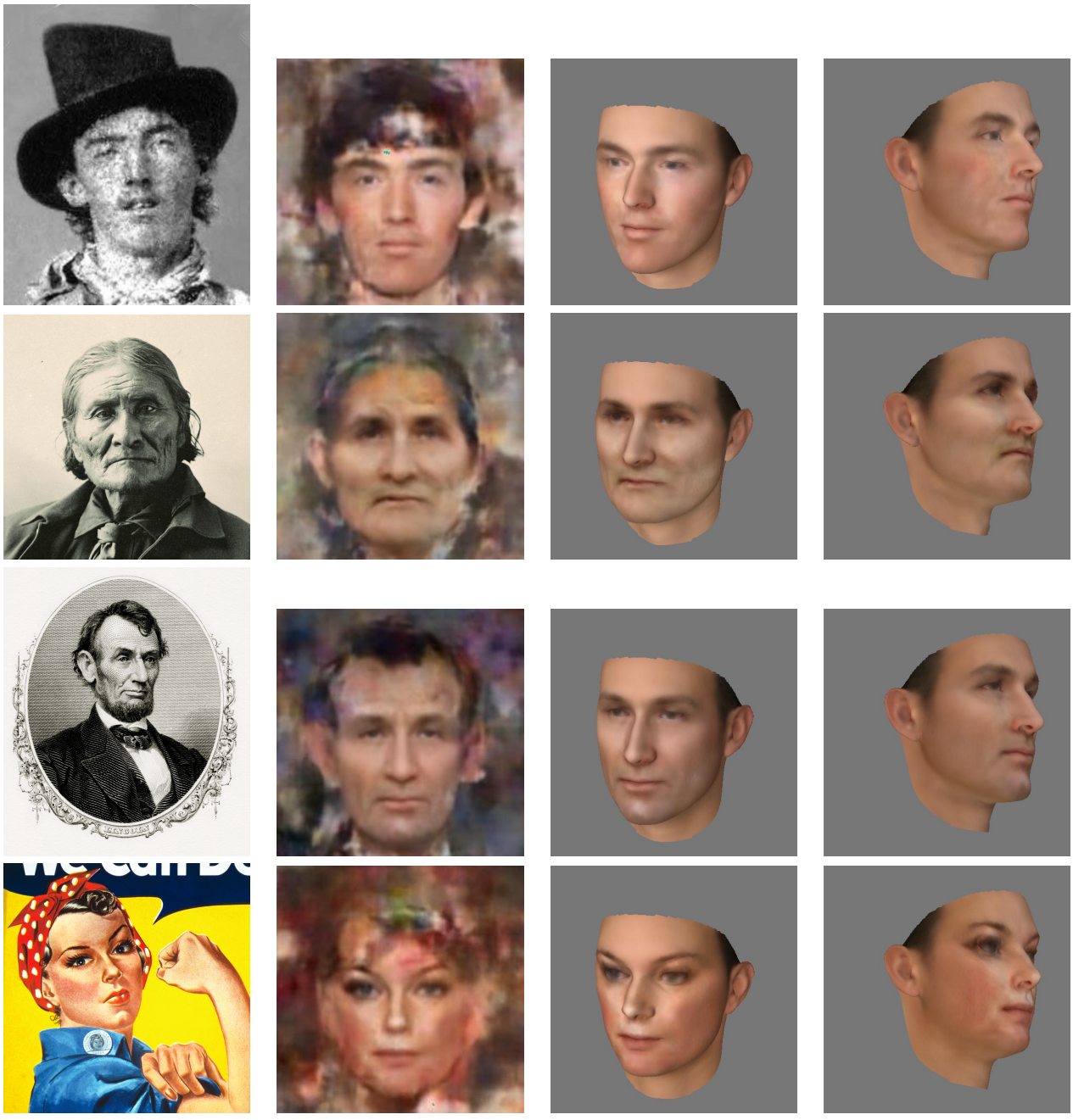

Figure 4. Though the model was only trained on natural images, it is robust enough to be applied to degraded photographs and illustrations. Column 1: input image. Column 2: generated 2-D image. Columns 3 and 4: images of 3-D reconstruction taken from 2 different angles.