

Unsupervised Training for 3D Morphable Model Regression

Kyle Genova^{1,2} Forrester Cole² Aaron Maschinot² Aaron Sarna² Daniel Vlasic² William T. Freeman^{2,3}

¹Princeton University ²Google Research ³MIT CSAIL

Abstract

We present a method for training a regression network from image pixels to 3D morphable model coordinates using only unlabeled photographs. The training loss is based on features from a facial recognition network, computed on-the-fly by rendering the predicted faces with a differentiable renderer. To make training from features feasible and avoid network fooling effects, we introduce three objectives: a batch distribution loss that encourages the output distribution to match the distribution of the morphable model, a loopback loss that ensures the network can correctly reinterpret its own output, and a multi-view identity loss that compares the features of the predicted 3D face and the input photograph from multiple viewing angles. We train a regression network using these objectives, a set of unlabeled photographs, and the morphable model itself, and demonstrate state-of-the-art results.

1. Introduction

A 3D morphable face model (3DMM) [3] provides a smooth, low-dimensional “face space” spanning the range of human appearance. Finding the coordinates of a person in this space from a single image of that person is a common task for applications such as 3D avatar creation, facial animation transfer, and video editing (e.g. [2, 7, 28]). The conventional approach is to search the space through inverse rendering, which generates a face that matches the photograph by optimizing shape, texture, pose, and lighting parameters [13]. This approach requires a complex, non-linear optimization that can be difficult to solve in practice.

Recent work has demonstrated fast, robust fitting by regressing from image pixels to morphable model coordinates using a neural network [20, 21, 29, 27]. The major issue with the regression approach is the lack of ground-truth 3D face data for training. Scans of face geometry and texture are difficult to acquire, both because of expense and privacy considerations. Previous approaches have explored synthesizing training pairs of image and morphable model coordinates in a preprocess [20, 21, 29], or training an image-

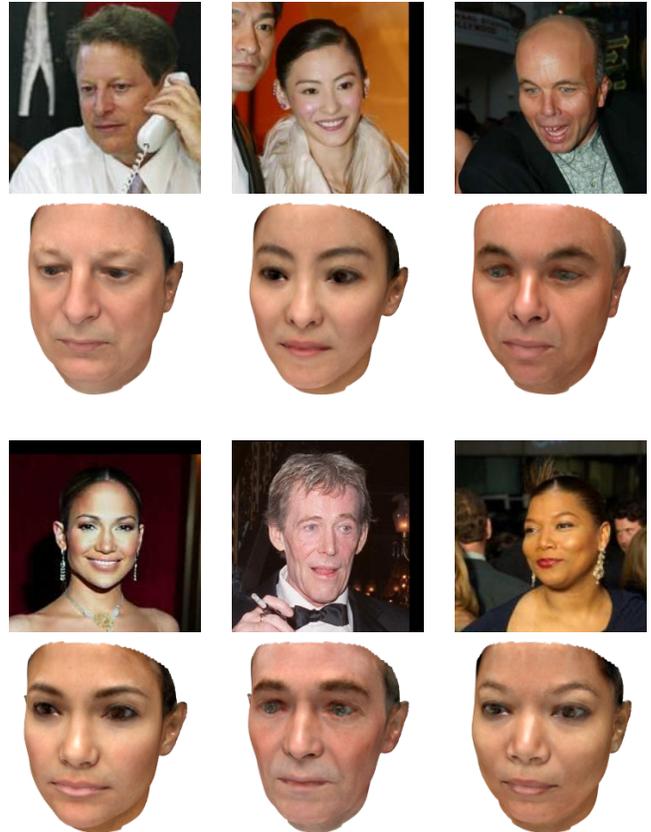


Figure 1. Neutral 3D faces computed from input photographs using our regression network. We map features from a facial recognition network [24] into identity parameters for the Basel 2017 Morphable Face Model [8].

to-image autoencoder with a fixed, morphable-model-based decoder and an image-based loss [27].

This paper presents a method for training a regression network that removes both the need for supervised training data and the reliance on inverse rendering to reproduce image pixels. Instead, the network learns to minimize a loss based on the facial identity features produced by a face recognition network such as VGG-Face [16] or Google’s FaceNet [24]. These features are robust to pose, expression, lighting, and even non-photorealistic inputs. We exploit this

invariance to apply a loss that matches the identity features between the input photograph and a synthetic rendering of the predicted face. The synthetic rendering need not have the same pose, expression, or lighting of the photograph, allowing our network to predict only shape and texture.

Simply optimizing for similarity between identity features, however, can teach the regression network to *fool* the recognition network by producing faces that match closely in feature space but look unnatural. We alleviate the fooling problem by applying three novel losses: a *batch distribution loss* to match the statistics of each training batch to the statistics of the morphable model, a *loopback loss* to ensure the regression network can correctly reinterpret its own output, and a *multi-view identity loss* that combines features from multiple, independent views of the predicted shape.

Using this scheme, we train a 3D shape and texture regression network using only a face recognition network, a morphable face model, and a dataset of unlabeled face images. We show that despite learning from unlabeled photographs, the 3D face results improve on the accuracy of previous work and are often recognizable as the original subjects.

2. Related Work

2.1. Morphable 3D Face Models

Blanz and Vetter [3] introduced the 3D morphable face model as an extension of the 2D active appearance model [6]. They demonstrated face reconstruction from a single image by iteratively fitting a linear combination of registered scans and pose, camera, and lighting parameters. They decomposed the geometry and texture of the face scans using PCA to produce separate, reduced-dimension geometry and texture spaces. Later work [8] added more face scans and extended the model to include expressions as another separate space. We build directly off of this work by using the PCA weights as the output of our network.

Convergence of iterative fitting is sensitive to the initial conditions and the complexity of the scene (i.e., lighting, expression, and pose). Subsequent work ([4, 22, 28, 7, 13] and others) has applied a range of techniques to improve the accuracy and stability of the fitting, producing very accurate results under good conditions. However, iterative approaches are still unreliable under general, in-the-wild, conditions, leading to the interest in regression-based approaches.

2.2. Learning to Generate 3D Face Models

Deep neural networks provide the ability to learn a regression from image pixels to 3D model parameters. The chief difficulty becomes how to collect enough training data to feed the network.

One solution is to generate synthetic training data by

drawing random samples from the morphable model and rendering the resulting faces [20, 21]. However, a network trained on purely synthetic data may perform poorly when faced with occlusions, unusual lighting, or ethnicities that are not well-represented by the morphable model. We include randomly generated, synthetic faces in each training batch to provide ground truth 3D coordinates, but train the network on real photographs at the same time.

Tran et al. [29] address the lack of training data by using an iterative optimization to fit an expressionless model to a large number of photographs, and treat results where the optimization converged as ground truth. To generalize to faces with expression, identity labels and at least one neutral image are required, so the potential size of the training dataset is restricted. We also directly predict a neutral expression, but our unsupervised approach removes the need for an initial iterative fitting step.

An approach closely related to ours was recently proposed by Tewari, et al. [27], who train an autoencoder network on unlabeled photographs to predict shape, expression, texture, pose, and lighting simultaneously. The encoder is a regression network from images to morphable-model coordinates, and the decoder is a fixed, differentiable rendering layer that attempts to reproduce the input photograph. Like ours, this approach does not require supervised training pairs. However, since the training loss is based on individual image pixels, the network is vulnerable to confounding variation between related variables. For example, it cannot readily distinguish between dark skin tone and a dim lighting environment. Our approach exploits a pretrained face recognition network, which distinguishes such related variables by extracting and comparing features across the entire image.

Other recent deep learning approaches predict depth maps [25] or voxel grids [10], trading off a compact and interpretable output mesh for more faithful reproductions of the input image. As for [27], identity and expression are confounded in the output mesh. The result may be suitable for image processing tasks, such as relighting, at the expense of animation tasks such as rigging.

2.3. Facial Identity Features

Current face recognition networks achieve high accuracy over millions of identities [12]. The networks operate by embedding images in a high-dimensional space, where images of the same person map to nearby points [24, 16, 26]. Recent work [5, 27] has shown that this mapping is somewhat reversible, meaning the features can be used to produce a likeness of the original person. We build on this work and use FaceNet [24] to both produce input features for our regression network, and to verify that the output of the regression resembles the input photograph.

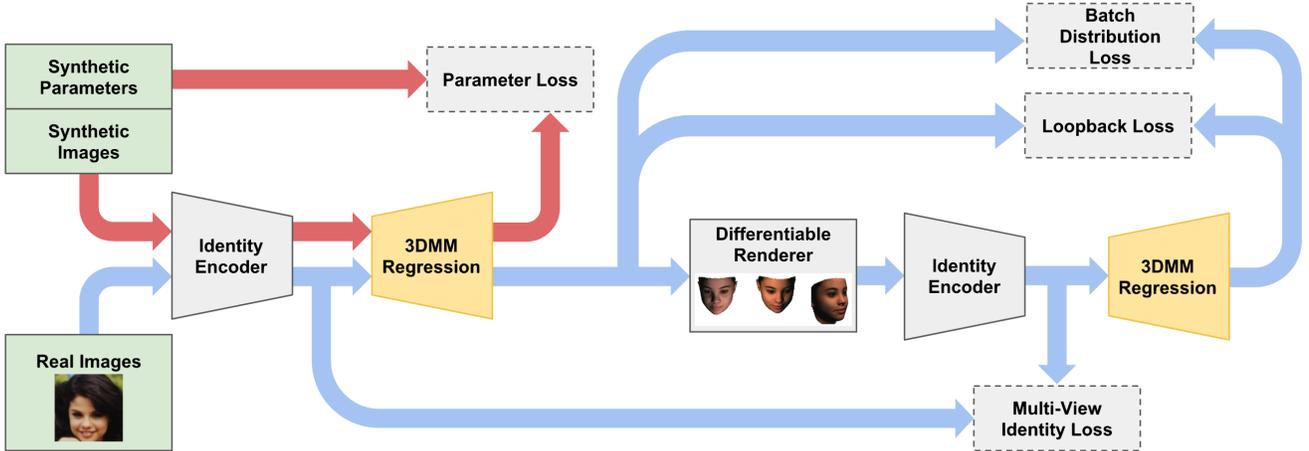


Figure 2. End-to-end computation graph for unsupervised training of the 3DMM regression network. Training batches consist of combinations of real (blue) and synthetic (red) face images. Identity, loopback and batch distribution losses are applied to real images, while the 3DMM parameter loss is applied to synthetic images. The regression network (yellow) is shown in two places, but both correspond to the same instance during training. The identity encoder network is fixed during training.

3. Model

We employ an encoder-decoder architecture that permits end-to-end unsupervised learning of 3D geometry and texture morphable model parameters (Fig. 2). Our training framework utilizes a realistic, parameterized illumination model and differentiable renderer to form neutral-expression face images under varying pose and lighting conditions. We train our model on hybrid batches of real face images from VGG-Face [16] and synthetic faces constructed from the Basel Face 3DMM [8].

The main strength and novelty of our approach lies in isolating our loss function to identity. By training the model to preserve identity through conditions of varying expression, pose, and illumination, we are able to avoid network fooling and achieve robust state-of-the-art recognizability in our predictions.

3.1. Encoder

We use FaceNet [24] for the network encoder, since its features have been shown to be effective for generating face images [5]. Other facial recognition networks such as VGG-Face [16], or even networks not focused on recognition, may work equally well.

The output of the encoder is the penultimate, 1024-D avgpool layer of the “NN2” FaceNet architecture. We found the avgpool layer more effective than the final, 128-D normalizing layer as input to the decoder, but use the normalizing layer for our identity loss (Sec. 3.3.2).

3.2. Decoder

Given encoder outputs generated from a face image, our decoder generates parameters for the Basel Face Model 2017 3DMM [8]. The Basel 2017 model generates shape meshes $\mathbf{S} \equiv \{\mathbf{s}_i \in \mathbb{R}^3 | 1 \leq i \leq N\}$ and texture meshes $\mathbf{T} \equiv \{\mathbf{t}_i \in \mathbb{R}^3 | 1 \leq i \leq N\}$ with $N = 53, 149$ vertices.

$$\begin{aligned} \mathbf{S} &= \mathbf{S}(\mathbf{s}, \mathbf{e}) = \boldsymbol{\mu}_S + \mathbf{P}_{SS} \mathbf{W}_{SS} \mathbf{s} + \mathbf{P}_{SE} \mathbf{W}_{SE} \mathbf{e} \\ \mathbf{T} &= \mathbf{T}(\mathbf{t}) = \boldsymbol{\mu}_T + \mathbf{P}_T \mathbf{W}_T \mathbf{t} \end{aligned} \quad (1)$$

Here, $\mathbf{s}, \mathbf{t} \in \mathbb{R}^{199}$ and $\mathbf{e} \in \mathbb{R}^{100}$ are shape, texture, and expression parameterization vectors with standard normal distributions; $\boldsymbol{\mu}_S, \boldsymbol{\mu}_T \in \mathbb{R}^{3N}$ are the average face shape and texture; $\mathbf{P}_{SS}, \mathbf{P}_T \in \mathbb{R}^{3N \times 199}$ and $\mathbf{P}_{SE} \in \mathbb{R}^{3N \times 100}$ are linear PCA bases; and $\mathbf{W}_{SS}, \mathbf{W}_T \in \mathbb{R}^{199 \times 199}$ and $\mathbf{W}_{SE} \in \mathbb{R}^{100 \times 100}$ are diagonal matrices containing the square roots of the corresponding PCA eigenvalues.

The decoder is trained to predict the 398 parameters constituting the shape and texture vectors, \mathbf{s} and \mathbf{t} , for a face. The expression vector \mathbf{e} is not currently predicted and is set to zero. The decoder network consists of two 1024-unit fully connected + ReLU layers followed by a 398-unit regression layer. The weights were regularized towards zero. Deeper networks were considered, but they did not significantly improve performance and were prone to overfitting.

3.2.1 Differentiable Renderer

In contrast to previous approaches [21, 27] that backpropagate loss through an image, we employ a general-purpose, differentiable rasterizer based on a deferred shading model. The rasterizer produces screen-space buffers containing triangle IDs and barycentric coordinates at each pixel. After

rasterization, per-vertex attributes such as colors and normals are interpolated at the pixels using the barycentric coordinates and IDs. This approach allows rendering with full perspective and any lighting model that can be computed in screen-space, which prevents image quality from being a bottleneck to accurate training. The source code for the renderer is publicly available¹.

The rasterization derivatives are computed for the barycentric coordinates, but not the triangle IDs. We extend the definition of the derivative of barycentric coordinates with respect to vertex positions to include negative barycentric coordinates, which lie outside the border of a triangle. Including negative barycentric coordinates and omitting triangle IDs effectively treats the shape as locally planar, which is an acceptable approximation away from occlusion boundaries. Faces are largely smooth shapes with few occlusion boundaries, so this approximation is effective in our case, but it could pose problems if the primary source of loss is related to translation or occlusion.

3.2.2 Illumination Model

Because our differentiable renderer uses deferred shading, illumination is computed independently per-pixel with a set of interpolated vertex attribute buffers computed for each image. We use the Phong reflection model [19] for shading. Because human faces exhibit specular highlights, Phong reflection allows for improved realism over purely diffuse approximations, such as those used in MoFA [27]. It is both efficient to evaluate and differentiable.

To create appropriately even lighting, we randomly position two point light sources of varying intensity several meters from the face to be illuminated. We select a random color temperature for each training image from approximations of common indoor and outdoor light sources, and perturb the color to avoid overfitting. Finally, since the Basel Face model does not contain specular color information, we use a heuristic to define specular colors K_s from the diffuse colors K_d of the predicted model: $K_s := c - cK_d$ for some manually selected constant $c \in [0, 1]$.

3.3. Losses

We propose a novel loss function that focuses on facial identity, and ignores variations in facial expression, illumination, pose, occlusion, and resolution. This loss function is conceptually straightforward and enables unsupervised end-to-end training of our network. It combines four terms:

$$L = L_{param} + L_{id} + \omega_{batch}L_{batch} + \omega_{loop}L_{loop} \quad (2)$$

Here, L_{param} imposes 3D shape and texture similarity for the synthetic images, L_{id} imposes identity preservation on

the real images in a batch, $L_{batchdistr}$ regularizes the predicted parameter distributions within a batch to the distribution of the morphable model, and $L_{loopback}$ ensures the network can correctly interpret its own output. The effects of removing the batch distribution, loopback, and limiting the identity loss to a single view are shown in Figure 3. We use $\omega_{batch} = 10.0$ and $\omega_{loop} = 0.07$ for our results.

Training proceeds in two stages. First, the model is trained solely on batches of synthetic faces generated by randomly sampling for shape, texture, pose, and illumination parameters. This stage performs only a partial training of the model: since shape and texture parameters are sampled independently in this stage, the model is restricted from learning correlations between them. Second, the partially-trained model is trained to convergence on batches consisting of a combination of real face images from the VGG-Face dataset [16] and synthetic faces. Synthetic faces are subject to only the L_{param} loss, while real face images are subject to all losses except L_{param} .

3.3.1 Parameter Loss

For synthetic faces, the true shape and texture parameters are known, so we use independent Euclidean losses between the randomly generated true synthetic parameter vectors, $\bar{\mathbf{s}}_b$ and $\bar{\mathbf{t}}_b$, and the predicted ones, \mathbf{s}_b and \mathbf{t}_b , in a batch.

$$L_{param} = \omega_s \sum_b |\mathbf{s}_b - \bar{\mathbf{s}}_b|^2 + \omega_t \sum_b |\mathbf{t}_b - \bar{\mathbf{t}}_b|^2 \quad (3)$$

where ω_s and ω_t control the relative contribution of the shape and texture losses. Due to different units, we set $\omega_s = 0.4$ and $\omega_t = 0.002$.

3.3.2 Identity Loss

Robust prediction of recognizable meshes can be facilitated with a loss that derives from a facial recognition network. We used FaceNet [24], though the identity-preserving loss generalizes to other networks such as VGG-Face [16]. The final FaceNet normalizing layer is a 128-D unit vector such that, regardless of expression, pose, or illumination, same-identity inputs map closer to each other on the hypersphere than different-identity ones. For our identity loss L_{id} , we define similarity of two faces as the cosine score of their respective output unit vectors, γ_1 and γ_2 :

$$L_{id}(\gamma_1, \gamma_2) = \gamma_1 \cdot \gamma_2 \quad (4)$$

To use this loss in an unsupervised manner on real faces, we calculate the cosine score between a face image and the image resulting from passing the decoder outputs into the differentiable renderer with random pose and illumination.

Identity prediction can be further enhanced by using multiple poses for each face. Multiple poses decrease the

¹http://github.com/google/tf_mesh_renderer

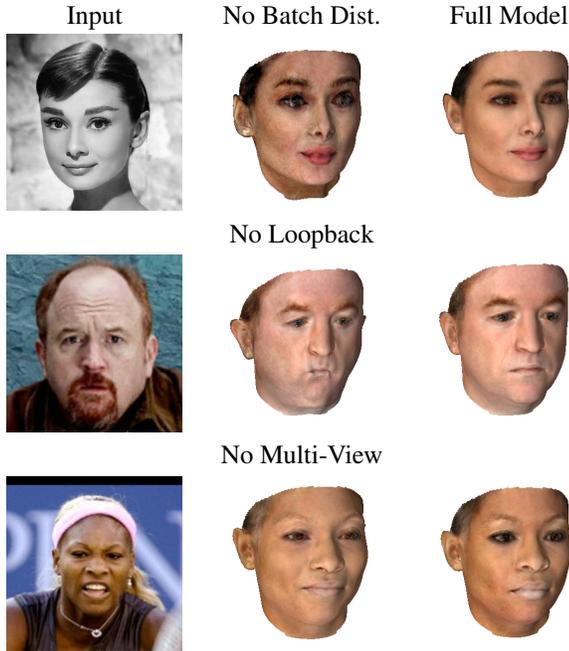


Figure 3. Ablation test showing failures caused by removing individual losses. Batch distribution (top) keeps the results in the space of human faces, while loopback (middle) helps avoid exaggerated features. Multi-view identity (bottom) increases robustness to expression and lighting variation. Ablated result is computed by rendering a single frontal view for identity loss.

presence of occluded regions of the mesh. Additionally, since each pose provides a backpropagation path to the mesh vertices, the model trains in a more robust manner than if only a single pose is used. We use three randomly determined poses for each real face.

3.3.3 Batch Distribution Loss

Applying the identity loss alone allows training to introduce biases into the decoder outputs that change their distribution from the zero-mean standard normal distribution assumption made by the Basel Face Model. These changes are likely due to domain transfer effects between the real images and those rendered from the decoder outputs. Initially, we attempted to compensate for these effects by adding a shallow network to transform the model-rendered encoder outputs prior to calculating the identity loss. While this approach did increase overall recognizability in the model, it also introduced unrealistic artifacts into the model outputs.

Instead, we opted to regularize each batch [23, 11] to directly constrain the lowest two moments of the shape and texture parameter distributions to match those of a zero-mean standard normal distribution. This loss, which is applied at a batch level, combines four terms:

$$L_{batchdistr} = L_{\mu_S} + L_{\sigma_S} + L_{\mu_T} + L_{\sigma_T} \quad (5)$$

Here, L_{μ_S} and L_{μ_T} regularize the batch shape and texture parameters to have zero mean, and L_{σ_S} and L_{σ_T} regularize them to have unit variance.

$$\begin{aligned} L_{\mu_S} &= |Mean_b[\mathbf{s}_b] - \mathbf{0}_{199}|^2 \\ L_{\sigma_S} &= |Var_b[\mathbf{s}_b] - \mathbf{1}_{199}|^2 \\ L_{\mu_T} &= |Mean_b[\mathbf{t}_b] - \mathbf{0}_{199}|^2 \\ L_{\sigma_T} &= |Var_b[\mathbf{t}_b] - \mathbf{1}_{199}|^2 \end{aligned} \quad (6)$$

3.3.4 Loopback Loss

A limitation of using real face images for unsupervised training is that the true shape and texture parameters for the faces are unknown. If they were known, then the more direct lower-level parameter loss in Sec. 3.3.1 could be directly imposed instead of the identity loss in Sec. 3.3.2.

A close approximation to this lower-level loss for real images can be achieved using a “loopback” loss (Fig. 2). The nature of this loss lies in generalizing the model near the regions for which real face image data exists. Similar techniques have proven to be successful in generalizing model learning for image applications[15, 14].

To compute the loopback loss at any training step, the current-state decoder outputs for a batch of real face images are extracted and used to generate synthetic faces rendered in random poses and illuminations. The synthetic faces are then passed back through the encoder and decoder again, and the parameter loss in Sec. 3.3.1 is imposed between the resulting parameters and those first output by the decoder.

As shown in Fig. 2, two loopback loss backpropagation paths to the decoder exist. The effects of each are complementary: the synthetic face parameter path generalizes the decoder in the region near that of real face parameters, and the real image channel regularizes the decoder away from generating unrealistic faces. Additionally, the two paths encourage the regression network to match its responses for real and synthetic versions of the same face.

4. Experiments

We first show and discuss the qualitative improvements of our method compared with other morphable model regression approaches (Sec. 4.1). We then evaluate our method quantitatively by comparing reconstruction error against scanned 3D face geometry (Sec. 4.2) and features produced by VGG-Face, which was not used for training (Sec. 4.3 and 4.4). We also show qualitative results on corrupted and non-photorealistic inputs (Sec. 4.5).

4.1. Qualitative Comparison

Figure 4 compares our results with the methods of of Tran, et al. [29], Tewari, et al. [27] (MoFA), and Sela, et al. [25] on 7 images from an 84-image test set developed

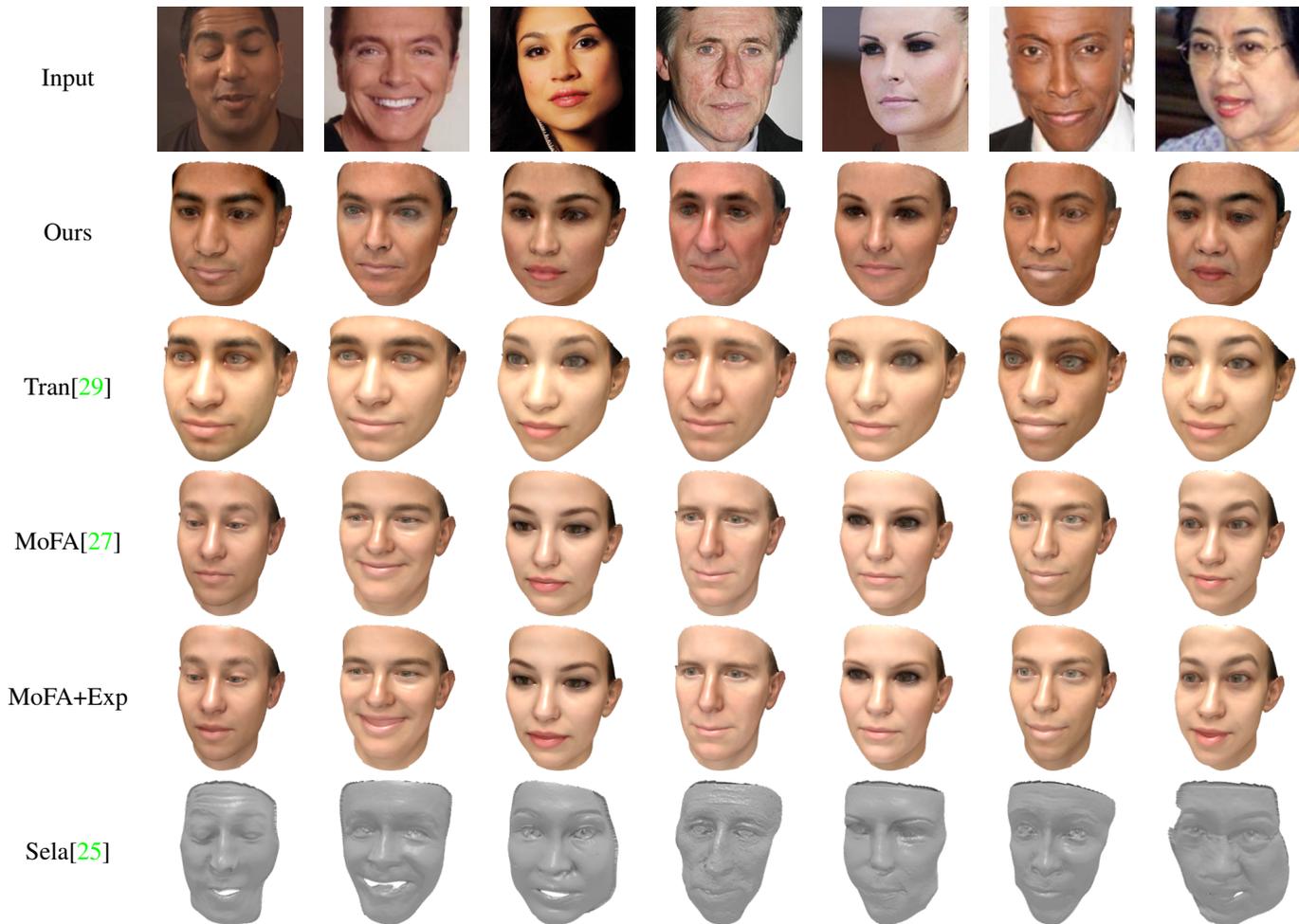


Figure 4. Results on the MoFA-Test dataset. Our method shows improved likeness and color fidelity over competing methods, especially in the shape of the eyes, eyebrows, and nose. Note that MoFA [27] solves for pose, expression, lighting, and identity, so is shown both with (row 5) and without (row 4) expression. The unstructured method of Sela, et al. [25] produces only geometry, so is shown without color.

as part of MoFA. An extended evaluation is available in the supplemental material. Our method improves on the likenesses of previous approaches, especially in features relevant to facial recognition such as the eyebrow texture and nose shape.

Our method also predicts coloration and skin tone more faithfully. This improvement is likely a consequence of our batch distribution loss, which allows individual faces to vary from the mean of the Basel model (light skin tone), so long as the faces match the mean *in aggregate*. Previous methods, by contrast, regularize *each face* towards the mean of the model’s distribution, tending to produce light skin tone overall.

The MoFA approach also sometimes confounds identity and expression (Fig. 4, second column), and skin tone and lighting (Fig. 4, first and sixth columns). Our method and Tran et al. [29] are more resistant to confounding variables. The unstructured method of Sela et al. [25] does not sepa-

rate identity and expression, predicts only shape, and is less robust than the model-based methods.

4.2. Neutral Pose Reconstruction on MICC

We quantitatively evaluate the ground-truth accuracy of our models on the MICC Florence 3D Faces dataset [1] (MICC) in Table 1. This dataset contains the ground truth scans of 53 Caucasian subjects in a neutral expression. Accompanying the scans are three observation videos for each subject, in conditions of increasing difficulty: ‘cooperative’, ‘indoor’, and ‘outdoor.’ We run the methods on each frame of the videos, and average the results over each video to get a single reconstruction. The results of Tran et al. [29] are averaged over the mesh, as in [29]. We instead average our encoder embeddings before making a single reconstruction.

To evaluate our predictions, we crop the ground truth scan to 95mm around the tip of the nose as in [29], and run ICP with isotropic scale to find an alignment. We solve

Method	Cooperative		Indoor		Outdoor	
	Mean	Std.	Mean	Std.	Mean	Std.
Tran et al.[29]	1.93	0.27	2.02	0.25	1.86	0.23
ours	1.50	0.13	1.50	0.11	1.48	0.11

Table 1. Mean Error on MICC Dataset using point-to-plane distance after ICP alignment of video-averaged outputs with isotropic scale estimation. Our errors lower on average and in variance, both within individual subjects and as conditions change.

for isotropic scale because we do not assume the methods predict absolute scale, and a small misalignment in scale can have a large effect on error. Table 1 shows the symmetric point-to-plane distance in millimeters within the ICP-determined region of intersection, rather than point-to-point distances, as the methods and ground truth have different vertex densities. Our results indicate that we have improved absolute error to the ground truth by 20-25%, and our results are more consistent from person to person, with less than half the standard deviation when compared to [29]. We are also more stable across changing environments, with similar results for all three test sets.

4.3. Face Recognition Results

In order to quantitatively evaluate the likeness of our reconstructions, we use the VGG-Face [16] recognition network’s activations as a measure of similarity. VGG-Face was chosen because FaceNet appears in our training loss, making it unsuitable as an evaluation metric. For each face in our evaluation datasets, we compute the cosine similarity of the $\phi(\ell_t)$ layers of VGG-Face between the input image and a rendering of our output geometry, as described in [16].

The similarity distributions for Labeled Faces in the Wild [9] (LFW), MICC, and MoFA-Test are shown in Figure 5. The similarity between all pairs of photographs in the LFW dataset, separated into same-person and different-person distributions, is shown for comparison in Fig. 5, top. Our method achieves an average similarity between rendering and photo of 0.403 on MoFA test (the dataset for which results for all methods are available). By comparison, 22.7% of pairs of photos of the same person in LFW have a score below 0.403, and only 0.04% of pairs of photos of different people have a score above 0.403.

For additional validation, Table 2 shows the Earth Mover’s distance [17] between the all-pairs LFW distributions and the results of each method. Our method’s results are closer to the same-person distribution than the different-person distribution in all cases, while the other methods’ results are closer to the different-person distribution. We conclude that ours is the first method that generates neutral-pose, 3D faces with recognizability approaching a photo.

The scores of the ground-truth 3D face scans from MICC

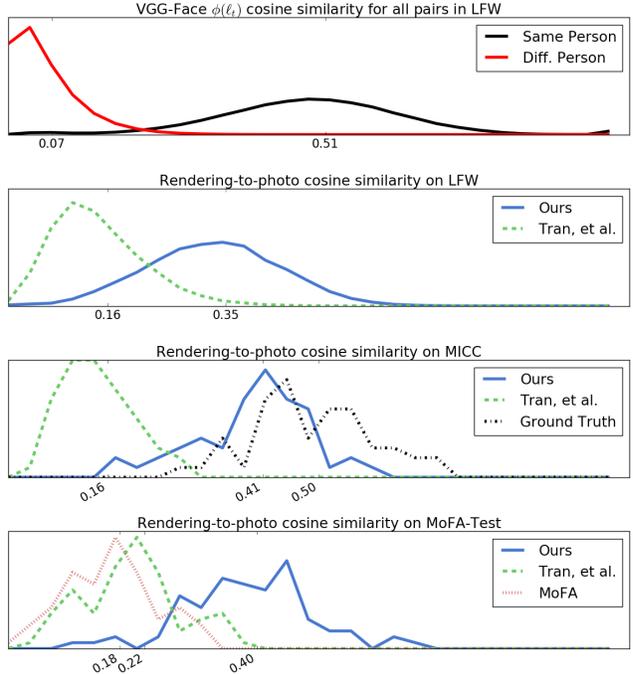


Figure 5. Distributions of cosine similarity between VGG-Face $\phi(\ell_t)$ layers for LFW, MICC, and MoFA-Test. Top: the similarity scores for all pairs of photos in LFW, divided into same and different person distributions. Below: similarity scores of our method, Tran, et al. [29], and MoFA [27] for photos and their corresponding 3D renderings on LFW, MICC, MoFA-Test. Mean values for each distribution are shown below. Camera and lighting parameters were fixed for all renderings.

Method	LFW		MICC		MoFA-T	
	Same	Diff.	Same	Diff.	Same	Diff.
MoFA [27]	—	—	—	—	0.30	0.11
Tran et al.[29]	0.32	0.09	0.32	0.09	0.27	0.14
ours	0.14	0.26	0.09	0.32	0.09	0.32
GT	—	—	0.03	0.41	—	—

Table 2. Earth mover’s distance between distributions of VGG-Face $\phi(\ell_t)$ similarity and distributions of same and different identities on LFW. A low distance for “Same” means the similarity scores between a photo and its associated 3D rendering are close to the scores of same identity photos in LFW, while a low distance for “Diff.” means the scores are close to the scores of different identity photos.

and their input photographs provide a ceiling for similarity scores. Notably, the distance between the GT distribution and the same-person LFW distribution is very low, with almost the same mean (0.51 vs 0.50), indicating the VGG-Face network has little trouble bridging the domain gap between photograph and rendering, and that our method does not yet reach the ground-truth baseline.

Method	MoFA-Test		LFW	
	Top-1	Top-5	Top-1	Top-5
random	0.01	0.06	0.0002	0.001
MoFA [27]	0.19	0.54	—	—
Tran et al. [29]	0.25	0.62	0.001	0.002
ours	0.87	0.96	0.16	0.51

Table 3. Identity Clustering Recall using VGG-Face distances on MoFA-Test and LFW. Given a rendered mesh, the task is to recover the unknown source identity by looking up the nearest neighbor photographs according to VGG-Face $\phi(\ell_t)$ cosine similarity. Top-1 and Top-5 show the fractions for which a photograph of the correct identity was recalled as the nearest neighbor, or in the nearest 5, respectively. Performance is higher for MoFA-Test because it contains 84 images and 78 identities, while the LFW set contains 12,993 images and 5,749 identities.

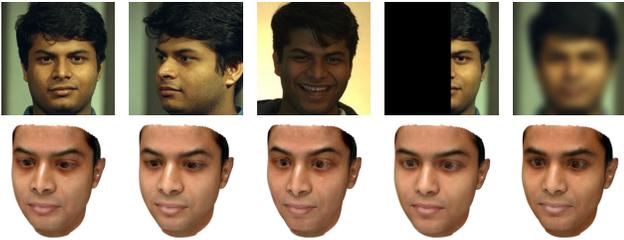


Figure 6. FERET dataset [18] stress test. The regression network is robust to changes in pose, lighting, expression, occlusion, and blur. See supplemental material for additional results.

4.4. Face Clustering

To establish that our reconstructions are recognizable, we perform a clustering task to recover the identities of our generated meshes. For each of LFW and MoFA-Test, we run our method on all faces in the dataset, and render the output geometry as shown in the figures in this paper. For each rendering, we find the nearest neighbors according to the VGG-Face $\phi(\ell_t)$ distance. Table 3 shows the fraction of meshes that recall a photo of the source identity as the nearest neighbor, and within the top 5 nearest neighbors.

On MoFA-Test, which has 84 images and 78 identities, we achieve a Top-1 recall of 87%, compared to 25% for Tran et al. and 19% for MoFA. On the larger LFW dataset, which contains over 5,000 identities in 13,000 photographs, we still achieve a Top-5 recall of 51%. We conclude our approach generates recognizable 3D morphable models, even in test sets with thousands of candidate identities.

4.5. Reconstruction from Challenging Images

Our regression network uses a facial identity feature vector as input, yielding results robust to changes in pose, expression, lighting, occlusion, and resolution, while remaining sensitive to changes in identity. Figure 6 qualitatively demonstrates this robustness by varying conditions for a single subject and displaying consistent output.



Figure 7. Art from the BAM dataset [30]. Because the inputs to our regression network are high-level identity features, the results are robust to stylized details at the pixel level.

Additionally, Figure 7 shows that our network can reconstruct plausible likenesses from non-photorealistic artwork, in cases where a fitting approach based on inverse rendering would have difficulty. This result is possible because of the invariance of the identity features to unrealistic pixel-level information, and because our unsupervised loss focuses on aspects of reconstruction that are important for recognition.

5. Discussion and Future Work

We have shown it is possible to train a regression network from images to neutral, expressionless 3D morphable model coordinates using only unlabeled photographs and improve on the accuracy of supervised methods. Our results approach the face recognition similarity scores of real photographs and exceed the scores of other regression approaches by a large margin. Because of the accuracy of the approach, the predicted face can be directly used for face-tracking based on landmarks.

This paper focuses on learning an expressionless face, which is suitable for creating VR avatars or landmark-based tracking. In future work, we hope to extend the approach to predict pose, expression, and lighting, similar to Tewari, et al. [27]. Predicting these factors while avoiding their confounding effects should be possible by adding an inverse rendering stage to our decoder while maintaining the neutral-pose losses we currently apply.

The method produces generally superior results for young adults and Caucasian ethnicities. The differences could be due to limited representation in the scans used to produce the morphable model, bias in the features extracted from the face recognition network, or limited representation in the VGG-Face dataset we use for training. In future work, we hope to improve the performance of the method on a diverse range of ages and ethnicities.

References

- [1] A. D. Bagdanov, A. Del Bimbo, and I. Masi. The florence 2d/3d hybrid face dataset. In *Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding*, J-HGBU '11, page 79–80, New York, NY, USA, 2011. ACM. **6**
- [2] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. *Computer Graphics Forum*, 22(3):641–650, 2003. **1**
- [3] 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. **1, 2**
- [4] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1063–1074, Sept. 2003. **2**
- [5] F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, and W. T. Freeman. Synthesizing normalized faces from facial identity features. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3386–3395, 2017. **2, 3**
- [6] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 484–498. Springer, 1998. **2**
- [7] P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt. Reconstruction of personalized 3d face rigs from monocular video. *ACM Trans. Graph.*, 35(3):28:1–28:15, May 2016. **1, 2**
- [8] T. Gerig, A. Forster, C. Blumer, B. Egger, M. Lüthi, S. Schönborn, and T. Vetter. Morphable face models - an open framework. *CoRR*, abs/1709.08398, 2017. **1, 2, 3**
- [9] 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. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. **7**
- [10] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 1031–1039. IEEE, 2017. **2**
- [11] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. **5**
- [12] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016. **2**
- [13] M. D. Levine and Y. (Chris) Yu. State-of-the-art of 3d facial reconstruction methods for face recognition based on a single 2d training image per person. *Pattern Recogn. Lett.*, 30(10):908–913, July 2009. **1, 2**
- [14] X. Li, Y. Dong, P. Peers, and X. Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Trans. Graph.*, 36(4):45:1–45:11, July 2017. **5**
- [15] V. Nair, J. Susskind, and G. E. Hinton. Analysis-by-synthesis by learning to invert generative black boxes. In *Proceedings of the 18th International Conference on Artificial Neural Networks, Part I, ICANN '08*, pages 971–981, Berlin, Heidelberg, 2008. Springer-Verlag. **5**
- [16] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. **1, 2, 3, 4, 7**
- [17] O. Pele and M. Werman. Fast and robust earth mover's distances. In *2009 IEEE 12th International Conference on Computer Vision*, pages 460–467. IEEE, September 2009. **7**
- [18] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing J*, 16(5):295–306, 1998. **8**
- [19] B. T. Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, June 1975. **4**
- [20] E. Richardson, M. Sela, and R. Kimmel. 3d face reconstruction by learning from synthetic data. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 460–469, Oct 2016. **1, 2**
- [21] E. Richardson, M. Sela, R. Or-El, and R. Kimmel. Learning detailed face reconstruction from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. **1, 2, 3**
- [22] S. Romdhani and T. Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 986–993, Washington, DC, USA, 2005. IEEE Computer Society. **2**
- [23] T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems 29*, pages 901–909. Curran Associates, Inc., 2016. **5**
- [24] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, June 2015. **1, 2, 3, 4**
- [25] M. Sela, E. Richardson, and R. Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1585–1594. IEEE, 2017. **2, 5, 6**
- [26] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, June 2014. **2**
- [27] A. Tewari, M. Zollöfer, H. Kim, P. Garrido, F. Bernard, P. Perez, and T. Christian. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. **1, 2, 3, 4, 5, 6, 7, 8**
- [28] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reen-

- actment of rgb videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016. 1, 2
- [29] A. T. Tran, T. Hassner, I. Masi, and G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 5, 6, 7, 8
- [30] M. J. Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. Belongie. Bam! the behance artistic media dataset for recognition beyond photography. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 8