Neural Face Editing with Intrinsic Image Disentangling

Zhixin Shu[1] Ersin Yumer[2] Sunil Hadap[2] Kalyan Sunkavalli[2] Eli Shechtman [2] Dimitris Samaras[1,3]

[1]Stony Brook University [2]Adobe Research [3] CentraleSupélec, Université Paris-Saclay
[1]{zhshu,samaras}@cs.stonybrook.edu [2]{yumer,hadap,sunkaval,elishe}@adobe.com

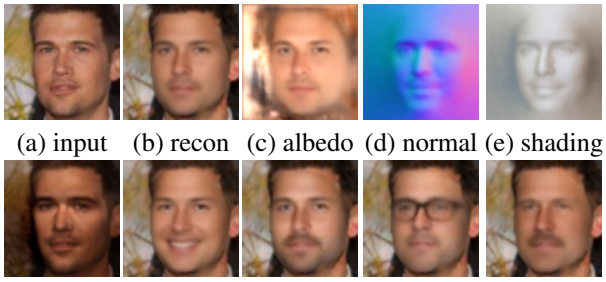
Abstract

Traditional face editing methods often require a number of sophisticated and task specific algorithms to be applied one after the other — a process that is tedious, fragile, and computationally intensive. In this paper, we propose an end-to-end generative adversarial network that infers a face-specific disentangled representation of intrinsic face properties, including shape (i.e. normals), albedo, and lighting, and an alpha matte. We show that this network can be trained on "in-the-wild" images by incorporating an in-network physically-based image formation module and appropriate loss functions. Our disentangling latent representation allows for semantically relevant edits, where one aspect of facial appearance can be manipulated while keeping orthogonal properties fixed, and we demonstrate its use for a number of facial editing applications.

1. Introduction

Understanding and manipulating face images in-the-wild is of great interest to the vision and graphics community, and as a result, has been extensively studied in previous work. This ranges from techniques to relight portraits [34], to edit or exaggerate expressions [36], and even drive facial performance [31]. Many of these methods start by explicitly reconstructing face attributes like geometry, texture, and illumination, and then edit these attributes to edit the image. However, reconstructing these attributes is a challenging and often ill-posed task; previous techniques deal with this by either assuming richer data (e.g., RGBD video streams) or a strong prior on the reconstruction that is adapted to the particular editing task that they seek to solve (e.g., low-dimensional geometry [6]). As a result, these techniques tend to be both costly and not generalize well to the large variations in facial identity and appearance that exist in images-in-the-wild.

In this work, our goal is to learn a compact, meaningful manifold of facial appearance, and enable face edits by walking along paths on this manifold. The remarkable success of morphable face models [6] – where face geometry and texture are represented using low-dimensional linear manifolds – indicates that this is possible for facial appearance. However, we would like to handle a much wider range of manipulations including changes in viewpoint, lighting, expression, and even higher-level attributes like facial hair and age – aspects that cannot be represented using previous models. In addition, we would like to learn this model without the need for expensive data capture [7].



(a) input (b) recon (c) albedo (d) normal (e) shading

(f) relit (g) smile (h) beard (i) eyewear (j) older

Figure 1. Given a face image (a), our network reconstructs the image (b) with in-network learned albedo (c), normal (d), and shading(e). Using this network, we can manipulate face through lighting (f), expression (g), appearance (h), eyewear (i), and time (j).

To this end, we build on the success of deep learning – especially unsupervised autoencoder networks – to learn "good" representations from large amounts of data [4]. Trivially applying such approaches to our problem leads to representations that are not meaningful, making the subsequent editing challenging. However, we have (approximate) models for facial appearance in terms of *intrinsic face properties* like geometry (surface normals), material properties (diffuse albedo), and illumination. We leverage this by designing the network to explicitly infer these properties and introducing an in-network forward rendering model that reconstructs the image from them. Merely introducing these factors into the network is not sufficient; because of the ill-posed nature of the inverse rendering problem, the learnt intrinsic properties can be arbitrary. We guide the network by imposing priors on each of these intrinsic properties; these include a morphable model-driven prior on the geometry,

a Retinex-based [20] prior on the albedo, and an assumption of low-frequency spherical harmonics-based lighting model [25, 3]. By combining these constraints with adversarial supervision on image reconstruction, and weak supervision on the inferred face intrinsic properties, our network is able to learn disentangled representations of facial appearance.

Since we work with natural images, faces appear in front of arbitrary backgrounds, where the physical constraints of the face do not apply. Therefore, we also introduce a matte layer to separate the foreground (i.e., the face) from the image background. This enables us to provide optimal reconstruction pathways in the network specifically designed for faces, without distorting the background reconstruction.

Our network naturally exposes low-dimensional manifold embeddings for each of the intrinsic properties, which in turn enables direct and data-driven semantic editing from *a single input image*. Specifically, we demonstrate direct illumination editing with explicit spherical harmonics lighting built into the network, as well as latent space manifold traversal for semantically meaningful expression edits such as smiling, and more structurally global edits such as aging. We show that by constraining physical properties that do not affect the target edits, we can achieve significantly more realistic results compared to other learning-based face editing approaches.

Our main contributions are: (1) We introduce an end-to-end generative network specifically designed for the understanding and editing of face images in the wild; (2) We encode the image formation and shading processes as in-network layers enabling the disentangling in the the latent space, of physically based rendering elements such as shape, illumination, and albedo; (3) We introduce statistical loss functions (such as *batchwise white shading* (BWS) corresponding to color consistency theory [20]) to improve disentangling latent representations.

2. Related Work

Face Image Manipulation. Face modeling and editing is an extensively studied topic in vision and graphics. Blanz and Vetter [6] showed that facial geometry and texture can be approximated by a low-dimensional morphable face model. This model and its variants have been used for a variety of tasks including relighting [34, 8], face attribute editing [7], expression editing [5, 22], authoring facial performances [32, 31], and aging [16]. Another class of techniques uses coarse geometry estimates to drive image-based editing tasks [36, 28, 14]. Each of these works develops techniques that are specifically designed for their application and often can not be generalized to other tasks. In contrast, our work aims to learn a general manifold for facial appearance that can support all these tasks.

Intrinsic decompositions. Barrow and Tanenbaum [2] proposed the concept of decomposing images into their physical intrinsic components such as surface normals, surface shading, etc. Barron and Mallik [1] extended this decomposition assuming a Lambertian rendering model with low-frequency illumination and made use of extensive priors on geometry, albedo, and illumination. This rendering model has also been used in face relighting [34] and shape-from-shading-based face reconstruction [15]. We use a similar rendering model in our work, but learn a face-specific appearance model by training a deep network with weak supervision.

Neural Inverse Rendering. Generative network architectures have shown to be effective for image manipulation. Kulkarni et al. [18] utilized a variational autoencoder (VAE) [17] for synthesizing novel variations of the input image where the objects pose and lighting conditions are altered. Yang et al. [37] demonstrated novel view synthesis for the object in a given image, where view specific properties were disentangled in latent space utilizing a recurrent network. In contrast, Tatarchenko et al. [30] used an autoencoder style network for the same task, where transformations were encoded through a secondary input stream and mixed with the input image in the latent space. Recently, Yan et al. [35] used a VAE variant and layered representations to generate images with specific semantic attributes. We adopt their background-foreground disentangling scheme through an in-network matte layer.

Face Representation Learning. Face representation learning is generally performed with a standard convolutional neural network trained for a recognition or labeling task [29, 24, 27]. Such approaches often need a significantly large amount of data since the network is treated as a black box. Synthetically boosting the dataset using normalizations and augmentations [29, 12, 13] has proven useful. Most recently, Masi et al. [23] used face fitting using morphable models similar to our approach, but used the resulting 3D faces to generate more data for traditional recognition network training. Even though such learned representations are powerful, especially in recognition, they are not straight forward to utilize for face editing.

Recently, Gardner et al. [10] demonstrated face editing through a standard recognition network. Since the network does not have a natural generation pathway, they use a two step optimization procedure (one in latent space, and one at low level feature space) to reconstruct the edited image. This, combined with the fact that they use a global latent space, leads to unintended changes and artifacts. On the other hand, our generative autoencoder style network allows for a physically meaningful latent space disentangling, thereby solving both problems: we constrain semantic edits to their corresponding latent representation, and our decoder generates the editing result in a single forward pass.

We formulate the face generation process as an end-to-

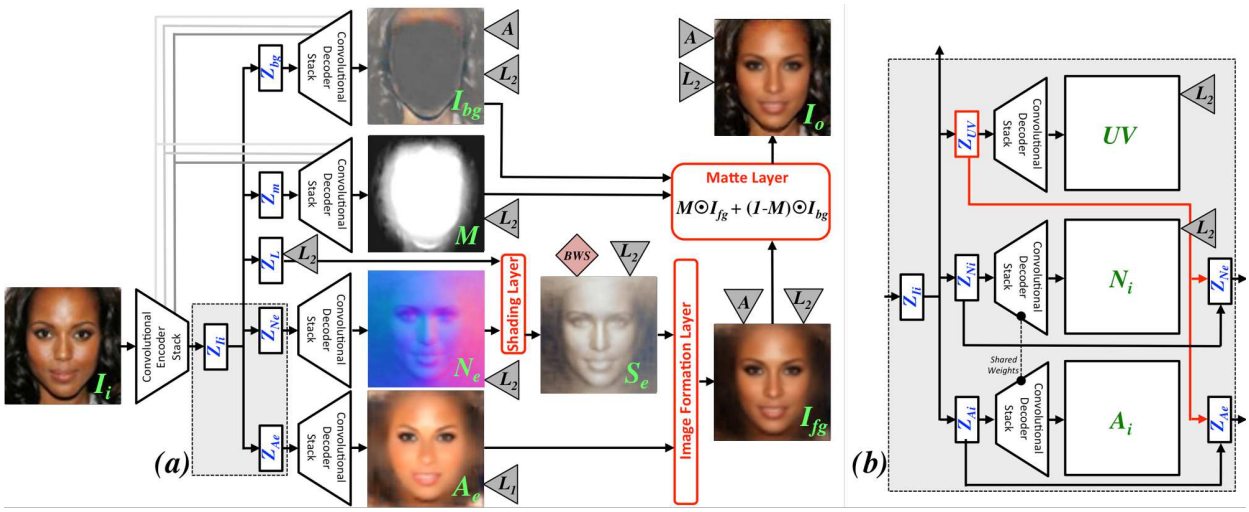


Figure 2. Network Architectures. The interchangeable modules (grey background-dashed boundary) highlight the difference between our two proposed architectures: (a) Direct modeling of explicit normal (N_e) and albedo (A_e) maps. (b) Implicit coordinate system (UV), albedo (A_i) and normal (N_i) modeling to aid further disentangling in the face foreground.

end network where the face is physically grounded by explicit in-network representations of its shape, albedo, and lighting. Fig. 2 shows the overall network structure. We first introduce the foreground *Shading Layer* and the *Image Formation Layer* (Sec. 2.1), followed by two alternative in-network face representations (Fig. 2(a)-(b) and Sec. 2.2) that are compatible with in-network image formation. Finally, we introduce in-network matting (Sec. 2.3) which further disentangles the learning process of the foreground and background for face images in the wild.

2.1. In-Network Physically-Based Face Rendering

From a graphics point of view, we assume a given face image I_{fg} is the result of a rendering process, $f_{\text{rendering}}$ where the inputs are an albedo map A_e, a normal map N_e, and illumination/lighting L:

$$I_{fg} = f_{\text{rendering}}(A_e, N_e, L) \tag{1}$$

We assume Lambertian reflectance and adopt Retinex Theory [20] to separate the albedo (i.e. reflectance) from the geometry and illumination:

$$I_{fg} = f_{\text{image-formation}}(A_e, S_e) = A_e \odot S_e \tag{2}$$

in which $\odot$ denotes the per-element product operation in the image space, and S_e represents a shading map rendered by N_e and L:

$$S_e = f_{\text{shading}}(N_e, L) \tag{3}$$

If Eqns. 2 and 3 are differentiable, they can be realized as in-network layers in an autoencoder network (Fig. 2(a)). This allows us to represent the image with disentangled latent variables for physically meaningful factors in the image formation process: the albedo latent variable Z_{A_e} , the normals variable Z_{N_e} and the lighting variable Z_L. We show that this is advantageous over the traditional approach of a single latent variable that encodes the combined effect of all image formation factors. Each of the latent variables allows us access to a specific manifold, where semantically relevant edits can be performed while keeping irrelevant latent variables fixed. For instance, one can trivially perform image relighting by only traversing the lighting manifold given by Z_L or changing only the albedo (e.g., to grow a beard) by traversing Z_{A_e}.

Computing shading from geometry (N_e) and illumination (L) is nontrivial under unconstrained conditions, and might result in $f_{\text{shading}}(\cdot,\cdot)$ being a discontinuous function in a significantly large region of the space it represents. Therefore, we further assume distant illumination, L, that is represented by *spherical harmonics* [25] s.t. the Lambertian shading function, $f_{\text{shading}}(\cdot,\cdot)$ has an analytical form and is differentiable.

Following previous work [25, 3, 34, 1], lighting L is represented by a 9-dimensional spherical harmonics coefficient vector. For a given pixel, i, with normal $\mathbf{n_i} = [n_x, n_y, n_z]^\top$, the shading is rendered as:

$$S_e^i = S_e(\mathbf{n}_i, L) = [\mathbf{n_i}; 1]^\top K [\mathbf{n_i}; 1] \tag{4}$$

where

$$K = \begin{bmatrix} c_1 L_9 & c_1 L_5 & c_1 L_8 & c_2 L_4 \\ c_1 L_5 & -c_1 L_9 & c_1 L_6 & c_2 L_2 \\ c_1 L_8 & c_1 L_6 & c_3 L_7 & c_2 L_3 \\ c_2 L_4 & c_2 L_2 & c_2 L_3 & c_4 L_1 - c_5 L_7 \end{bmatrix} \tag{5}$$

$$c1 = 0.429043 \quad c2 = 0.511664$$
$$c3 = 0.743125 \quad c4 = 0.886227 \quad c5 = 0.247708$$

We provide the formulas for the partial derivatives $\frac{\partial S_e^i}{\partial n_x}$, $\frac{\partial S_e^i}{\partial n_y}$, $\frac{\partial S_e^i}{\partial n_z}$ and $\frac{\partial S_e^i}{\partial L_j}$ in the supplementary material. Using these two differential rendering modules f_{shading} and $f_{\text{image-formation}}$, we can now implement the rendering modules within the network as shown in Figure 2.

2.2. In-Network Face Representation

Explicit Representation. The formulation introduced in the previous section requires the image formation and shading variables to be defined in the image coordinate system. This can be achieved with an *explicit* per-pixel representation of the face properties: N_e, A_e. Figure 2(a) depicts the module where the explicit normals and albedo are represented by their latent variables Z_{N_e}, Z_{A_e}. Note that the lighting, L, is independent of the face representation; we represent it using spherical harmonics coefficients, i.e., $Z_L = L$ is directly used by the shading layer whose forward process is given by Eqn. 4.

Implicit Representation. Even though the explicit representation helps disentangle certain properties and relates edits more intuitively to the latent variable manifolds (i.e. relighting), it might not be satisfactory in some cases. For instance, pose and expression edits might change both the explicit per-pixel normals, as well as the per-pixel albedo in the image space. We therefore introduce an *implicit* representation, where the parametrization is over the face coordinate system rather than the image coordinate system. This will allow us to further constrain pose and expression changes to the shape (i.e. normal) space only.

To address this, we introduce an alternative network architecture where the explicit representation depicted in the module in Fig. 2(a) is replaced with Fig. 2(b). Here, UV represents the per-pixel face space uv-coordinates, N_i and A_i represent the normal and albedo maps in the face uv-coordinate system, and Z_{UV}, Z_{N_i}, and Z_{A_i} represent the corresponding latent variables respectively. This is akin to the standard UV-mapping process in computer graphics. Facial features are aligned in this space (eyes correspond to eyes, mouths to mouths, etc.), and as a result the network has to learn a smaller space of variation, leading to sharper, more accurate reconstructions. Note that even though the network only uses the explicit latent variables at test time, we have auxiliary decoder stacks for all implicit variables to encourage disentangling of these variables during training. The implementation and training details will be explained in Sec. 3.2.

2.3. In-Network Background Matting

To further encourage the physically based representations of albedo, normals and lighting to concentrate on the face region, we disentangle the background from the foreground with a matte layer similar to the work by Yan et al. [35]. The matte layer computes the composite of the foreground face onto the background:

$$I_o = M \odot I_{fg} + (1 - M) \odot I_{bg} \tag{6}$$

The matting layer also enables us to utilize efficient skip layers where unpooling layers in the decoder stack can use pooling switches from the corresponding encoder stack of the input image (grey links from the input encoder to background and mask decoders in Figure 2). The skip connection between the encoder and the decoder, allow for the details of the background to be preserved to a greater extent. Such skip connections bypass the bottleneck Z and therefore allow only partial information flow through Z during training.

For the foreground face region we chose to "filter" all the information through the bottleneck Z without any skip connections in order to gain full control over the latent manifolds for editing, at the expense of some detail loss.

3. Implementation

3.1. Network Architecture

The convolutional encoder stack (Fig. 2) is composed of three convolutions with $32{*}3{\times}3$, $64{*}3{\times}3$ and $64{*}3{\times}3$ filter sets. Each convolution is followed by max-pooling and a ReLU nonlinearity. We pad the filter responses after each pooling layer so that the final output of the convolutional stack is a set of filter responses with size $64 * 8 \times 8$ for an input image $3 * 64 \times 64$.

Z_{I_i} is a latent variable vector of 128×1 which is fully connected to the last encoder stack downstream as well as the individual latent variables for background Z_{bg}, mask Z_m, light Z_L, and the foreground representations. For the explicit foreground representation, it is directly connected to Z_{N_e} and Z_{A_e} (Fig. 2(a)), whereas for the implicit representation it is connected to Z_{UV}, Z_{N_i}, and Z_{A_i} (Fig. 2(b)). All individual latent representations are 128×1 vectors except for Z_L which represents the light L directly and is thus a 27×1 vector (three 9×1 concatenated vectors representing the spherical harmonics of the RGB components).

All decoder stacks for upsampling per-pixel (explicit or implicit) values are strictly symmetric to the encoder stack. As described in Sec. 2.3, the decoder stacks for the mask and background have skip connections to the input encoder stack at corresponding layers. The implicit normals N_i and implicit albedo A_i share weights in the decoder, since we have supervision of the implicit normals only.

3.2. Training

We use "in-the-wild" face images for training. Hence, we only have access to the image itself (denoted by I^*), and do not have ground-truth data for either illumination,

normal map, or the albedo. The main loss function is therefore on the reconstruction of the image I_i at the output I_o:

$$E_o = E_{\text{recon}} + \lambda_{\text{adv}} E_{\text{adv}} \tag{7}$$

where $E_{\text{recon}} = ||I_i - I_o||^2$. E_{adv} is given by the adversarial loss, where a discriminative network is trained at the same time to distinguish between the generated and real images [11]. Specifically, we use an energy-based method [38] to incorporate the adversarial loss. In this approach an autoencoder is used as the discriminative network, $\mathcal{D}$. The adversarial loss for the generative network is defined as $E_{\text{adv}} = D(I')$, where I' is the reconstruction of the discriminator input I_o, hence $D(.)$ is the L_2 reconstruction loss of the discriminator $\mathcal{D}$. We train $\mathcal{D}$ to minimize the margin-based reconstruction error proposed by [38],

Fully unsupervised training using only the reconstruction and adversarial loss on the output image will often result in semantically meaningless latent representations. The network architecture itself cannot prevent degenerate solutions, e.g. when A_e captures both albedo and shading information while S_e remains constant. Since each of the rendering elements has a specific physical meaning, and they are explicitly encoded as intermediate layers in the network, we introduce additional constraints through intermediate loss functions to guide the training.

First, we introduce $\hat{N}$, a "pseudo ground-truth" of the normal map N_e, to keep the normal map close to plausible face normals during the training process. We estimate $\hat{N}$ by fitting coarse face geometry to every image in the training set using a 3D Morphable Model [6]. We then introduce the following objective to N_e:

$$E_{\text{recon-N}} = ||N_e - \hat{N}||^2 \tag{8}$$

Similar to $\hat{N}$, we provide a $L2$ reconstruction loss w.r.t $\hat{L}$, on the lighting parameters Z_L:

$$E_{\text{recon-L}} = ||Z_L - \hat{L}||^2 \tag{9}$$

where $\hat{L}$ is computed from $\hat{N}$ and the input image using least square optimization and a constant albedo assumption [33, 34].

Furthermore, following Retinex theory [20] which assumes albedo to be piecewise constant and shading to be smooth, we introduce an $L1$ smoothness loss on the gradients of the albedo, A:

$$E_{\text{smooth-A}} = ||\nabla A_e|| \tag{10}$$

in which ∇ is the spatial image gradient operation. In addition, since the shading is assumed to vary smoothly, we introduce an $L2$ smoothness loss on the gradients of the shading, S_e:

$$E_{\text{smoothS}} = ||\nabla S_e||^2 \tag{11}$$

For the implicit coordinate system (UV) variant (Fig. 2-(b)), we provide $L2$ supervisions to both UV and N_i:

$$E_{\text{UV}} = ||UV - \hat{UV}||^2 \tag{12}$$

$$E_{N_i} = ||N_i - \hat{N}_i||^2 \tag{13}$$

$\hat{UV}$ and $\hat{N}_i$ are obtained from the previously mentioned Morphable Model, in which vertex-wise correspondence on the 3D fit exists. We utilize the average shape of the Morphable Model $\bar{S}$ to construct a canonical coordinate map (UV) and surface normal map (N_i), and propagate it to each shape estimation via this correspondence. More details of this computation are presented in our supplemental document.

Due to ambiguity in the magnitude of lighting, and therefore the intensity of shading (Eq. 2), it is necessary to incorporate constraints on the shading magnitude to prevent the network from generating arbitrary bright/dark shading. Moreover, since the illumination is separated in individual colors $\mathbf{L}_r$, $\mathbf{L}_g$ and $\mathbf{L}_b$, we incorporate a constraint to prevent the shading from being too strong in one color channel vs. the others. To handle these ambiguities, we introduce a *Batch-wise White Shading (BWS)* constraint on S_e:

$$\frac{1}{m}\sum_{i,j} s_r^i(j) = \frac{1}{m}\sum_{i,j} s_g^i(j) = \frac{1}{m}\sum_{i,j} s_b^i(j) = c \tag{14}$$

where $s_r^i(j)$ denotes the j-th pixel of the i-th example in the first (red) channel of S_e. s_g and s_b denote the second and the third channel of shading respectively. m is the number of pixels in a training batch. In all experiments $c = 0.75$.

Since $\hat{N}$ obtained by the Morphable Model comes with a region of interest only on the face surface, we use it as the mask under which we compute all foreground losses. In addition, this region of interest is also used as the mask pseudo ground truth at training time for learning the matte mask:

$$E_M = ||M - \hat{M}||^2 \tag{15}$$

in which $\hat{M}$ represents the Morphable Model mask.

4. Experiments

We use the CelebA [21] dataset to train the network. For each image in the dataset, we detect landmarks [26], and fit a 3D Morphable Model [6, 36] to the face region to have a rough estimation of the rendering elements ($\hat{N}, \hat{L}$). These estimates are used to set-up the various losses detailed in the previous section. This data is subsequently used only for the training of the network as previously described.

4.1. Baseline Comparisons

For comparison, we train an autoencoder $\mathcal{B}$ as a baseline. The encoder and decoder of $\mathcal{B}$ is identical to the encoder and decoder for albedo in our architecture. To make

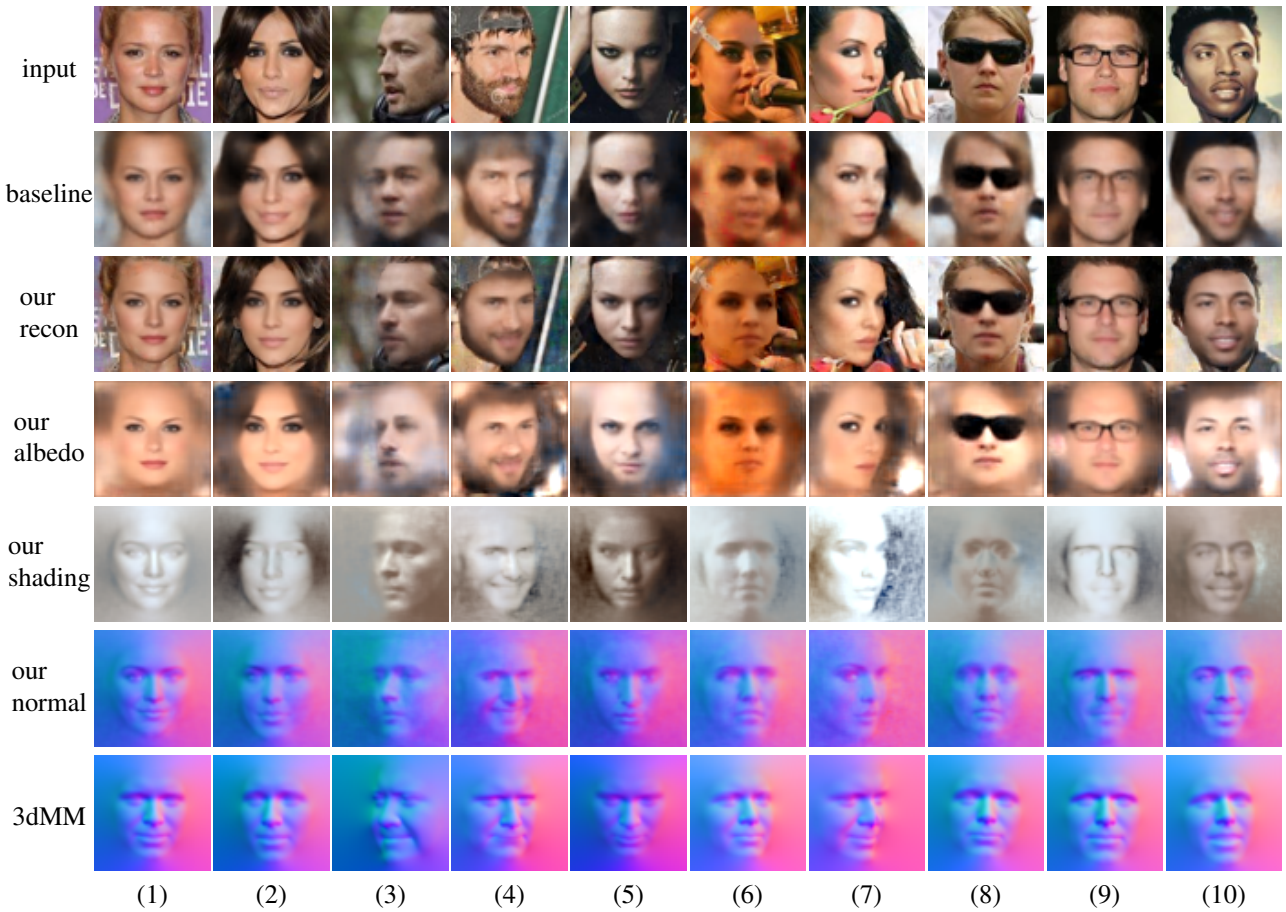


Figure 3. Feed-forward reconstruction and normals, shading, albedo estimation. Compared to the baseline autoencoder (row 2), our reconstruction (row 3) not only preserves the details of the background (1,2,4), but is also more robust to complex pose (3,4), illumination (5), and identity (9,10), thanks to the layered representation and in-network rendering procedure. Moreover, our network contains components that explicitly encode normals (row 4), shading/lighting (row 5), and albedo (row 6) for the foreground (face), which is helpful for the understanding and manipulation of face images. In the last row we show the normal estimation from a 3D Morphable Model. We can easily see that using our network, the generated shape retains more identity information from the original image, and does not fall in the sub-space of the PCA-based morphable model that is used as weak supervision for training. All results are produced by the network designed for explicit representation.

the comparison fair, the bottleneck layer of $\mathcal{B}$ is set to 265 ($= 128 + 128 + 9$) dimensions, which is more than twice as large as the bottleneck layer in our architecture (size 128), yielding slightly more capacity for the baseline. Even though our architecture has a narrower bottleneck, the disentangling of the latent factors and the presence of physically based rendering layers, lead to reconstructions that are more robust to complex background, pose, illumination, occlusion, etc., (Fig. 3).

More importantly, given an input face images, our network provides explicit access to an estimation of the albedo, shading and normal map (Fig. 3) for the face. Notably, in the last row of Fig. 3, we compare the inferred normals from our network with the normals estimated from the input image using the 3D morphable model that we deployed to guide the training process. The data to construct the morphable model contains only 16 identities; this small subspace of identity variation leads to normals that are often inaccurate approximations of the true face shape (row 7 in Fig. 3). By using these estimates as weak supervision in combination with an appearance-based rendering loss, our network is able to generate normal maps (row 6 in Fig. 3) that extend beyond the morphable model subspace, better fit the shape of the input face, and exhibit more identity information. Please refer to our supplementary material for more comparisons.

4.2. Face Editing by Manifold Traversal

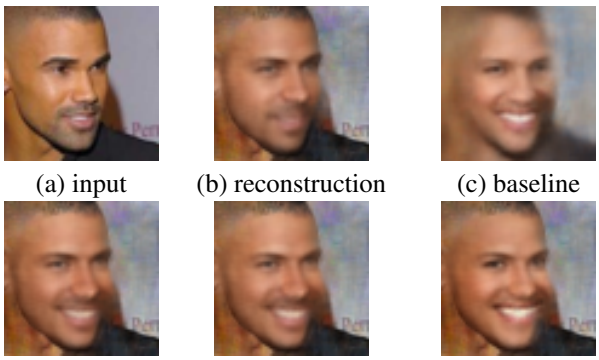
Our network enables manipulation of semantic face attributes, (e.g. expression, facial hair, age, makeup, and eyewear) by traversing the manifold(s) of the disentangled latent spaces that are most appropriate for that edit.

(a) (b) (c) (d) (e)

Figure 4. Smile editing via traversal on our representation (explicit albedo and normal) vs. a baseline autoencoder representation. Our network provides better reconstructions (d) of the input images (a) and captures the geometry and appearance changes associated with smiling (e). The baseline network leads to poorer reconstructions (b) and edits (c).

For a given attribute, e.g., the *smiling expression*, we feed both positive data $\{\mathbf{x}_p\}$ (smiling faces) and negative data $\{\mathbf{x}_n\}$ (faces with other expressions) into our network to generate two sets of Z-codes $\{\mathbf{z}_p\}$ and $\{\mathbf{z}_n\}$. These sets represent corresponding empirical distributions of the data on the low dimensional Z-space(s). Given an input face image I_{source} that is *not smiling*, we seek to *make it smile* by moving its Z-code(s) Z_{source} towards the distribution $\{\mathbf{z}\}_p$ to get a transformed code Z_{trans}. After that, we reconstruct the image corresponding to Z_{trans} with the decoders in our model.

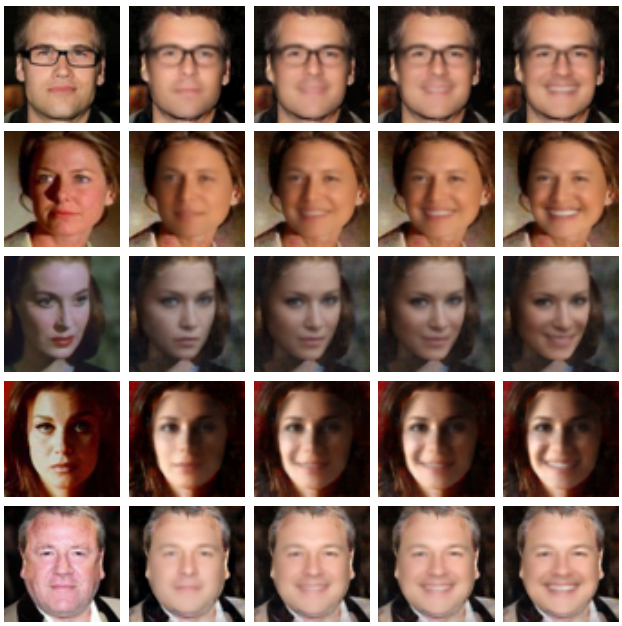
In order to compute the distributions for each attribute, we sample a subset of 2000 images from the CelebA [21] with the appropriate attribute label (e.g., smiling vs other expressions). We use the manifold traversal method proposed by Gardner et al. [10] independently on each appropriate variable. The extent of the traversal is parameterized by a regularization parameter, λ (see [10] for details).



(a) input (b) reconstruction (c) baseline

(d)Z_{UV} (e) Z_{UV},Z_{Ni} (f) Z_{UV},Z_{Ni},Z_{Ai}

Figure 5. Smile editing via implicit factor traversal. Our implicit representation directly captures smiling via a traversal of the UV manifold (d) and both UV and implicit normal (e). Traversing on the implicit albedo on the other hand, leads to noticeable appearance artifacts (f). For this experiment, we use the same regularization (λ= 0.03) on all manifolds.



(a) input (b) recon (c) (d) (e)

Figure 6. *Smile* editing via progressive traversal on the bottleneck manifolds (Z_{UV} and Z_{N_i}). From (c) to (e), λ is 0.07, 0.05, 0.03 respectively. As the latent representation moves closer to the $smiling$ mode, stronger features of smiling, such as rising cheeks and white teeth, appear. Note that we are also able to capture subtle changes in the eyes that are often correlated with smiling.

In Fig. 4, we compare the results using our network against the baseline autoencoder. We traverse the albedo and normal variables to produce edits which make the faces smile and are able to capture changes in expression and the appearance of teeth, while preserving the other aspects of the image. In contrast, the results from traversing the baseline latent space are much poorer – in addition to not being able to reconstruct the pose and identity of the input properly, the traversal is not able to capture the smiling transformation as well as we do.

In Fig. 5 we demonstrate the utility of our implicit representation. While lips/mouth and teeth might map to the same region of the image space, they are in fact separated in the face UV-space. This allows the implicit variables to learn more targeted and accurate representations, hence traversing just the Z_{UV}, already results in a smiling face. Combining this with traversal along Z_{N_i} exaggerates the smile. In contrast, we do not expect smiling to be correlated with the implicit albedo space, and traversing along the Z_{A_i} leads to poorer results with an incorrect frontal pose.

In Fig. 6 we demonstrate more results for *smiling* and demonstrate that relaxing the traversal regularization parameter, λ, gradually leads to stronger smiling expressions.

We also address the editing task of *aging* via manifold traversal. For this experiment, we construct the latent space distributions using images and labels from the PubFig [19]

(a) input (b) recon (c) (d) (e)

Figure 7. Aging via traversal on the albedo and normal manifolds. From (c) to (e), λ is 0.07, 0.05, 0.03 respectively. As the latent representation moving towards to the *senior* mode, stronger features of *aging*, such as changes in face shape and texture, appear while retaining other aspects of the appearance like pose, lighting, and eyewear.

dataset corresponding to the most and least *senior* images. We expect aging to be correlated with both shape and texture, and show in Fig. 7 that traversing these manifolds leads to convincing age progression.

Note that all of these edits have been performed on the exact same network, indicating that our network architecture is general enough to represent the manifold of face appearance, and is able to disentangle the latent factors to support specific editing tasks. Refer to our supplementary material for more results, and comparisons.

Limitations. Our current face masks do not include hair. This results in less control over some edits, e.g. aging, that are inherently affecting the hair as well. However, this can trivially be addressed, if a mask that also includes the hair can be generated [9].

4.3. Relighting

A direct application of the albedo-normal-light decomposition in our network is that it allows us to manipulate the illumination of an input face via Z_L while keeping the other latent variable fixed. We can directly "relight" the face by replacing its Z_L^{target} with some other Z_L^{source} (e.g. using the lighting variable of another face).

While our network is trained to reconstruct the input, due to its limited capacity (especially due to the bottleneck layer dimensionality), the reconstruction does not reproduce the input with all the details. For illumination editing, however, we can directly manipulate the shading, that is also available in our network. We pass the source I^{source} and target images I^{target} through our network to estimate their individual factors. We use the target shading S^{target} with Eq. 2 to compute a "detailed" albedo A^{target}. Given the source light L^{source}, we render the shading of the target under this light with the target normals N^{target} (Eq. 3) to obtain the transferred shading S^{transfer}. In the end, the lighting transferred image is rendered with A^{target} and S^{transfer} using Eq. 2. This is demonstrated in Fig. 8 where we are able to successfully transfer the lighting from two sources with disparate identities, genders, and poses to a target while retaining all its details. We present more relighting results, as well as quantitative tests on illumination (i.e. spherical harmonics coefficients) prediction in the supplementary material.

(a) target (b) source (c) S^{source} (d) transfer (e) S^{transfer}

Figure 8. Lighting transfer using our model. We transfer the illumination of two source images (b) to a given target (a)(top: image; bottom: estimated normal), by generating the shading (e) of the target using the lighting of the source, and applying to the original target image.

5. Conclusions

We proposed a physically grounded rendering-based disentangling network specifically designed for faces. Such disentangling enables realistic face editing since it allows trivial constraints at manipulation time. We are the first to attempt in-network rendering for faces in the wild with real, arbitrary backgrounds. Comparisons with traditional autoencoder approaches show significant improvements on final edits, and our intermediate outputs such as face normals show superior identity preservation compared to traditional approaches.

6. Acknowledgements

This work started when Zhixin Shu was an intern at Adobe Research. This work was supported by a gift from Adobe, NSF IIS-1161876, the Stony Brook SensorCAT and the Partner University Fund 4DVision project.

References

[1] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2015.

[2] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. Technical Report 157, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Apr 1978.

[3] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE transactions on pattern analysis and machine intelligence*, 25(2):218–233, 2003.

[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[5] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating Faces in Images and Video. *Computer Graphics Forum*, 2003.

[6] 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.

[7] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, Mar. 2014.

[8] M. Chai, L. Luo, K. Sunkavalli, N. Carr, S. Hadap, and K. Zhou. High-quality hair modeling from a single portrait photo. *ACM Trans. Graph.*, 34(6):204:1–204:10, Oct. 2015.

[9] M. Chai, T. Shao, H. Wu, Y. Weng, and K. Zhou. Autohair: Fully automatic hair modeling from a single image. *ACM Transactions on Graphics (TOG)*, 35(4):116, 2016.

[10] J. R. Gardner, M. J. Kusner, Y. Li, P. Upchurch, K. Q. Weinberger, and J. E. Hopcroft. Deep manifold traversal: Changing labels with convolutional features. *arXiv preprint arXiv:1511.06421*, 2015.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[12] T. Hassner. Viewing real-world faces in 3d. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3607–3614, 2013.

[13] 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*, pages 4295–4304, 2015.

[14] I. Kemelmacher-Shlizerman. Transfiguring portraits. *ACM Transactions on Graphics (TOG)*, 35(4):94, 2016.

[15] I. Kemelmacher-Shlizerman and S. M. Seitz. Face reconstruction in the wild. In *2011 International Conference on Computer Vision*, pages 1746–1753. IEEE, 2011.

[16] I. Kemelmacher-Shlizerman, S. Suwajanakorn, and S. M. Seitz. Illumination-aware age progression. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3334–3341. IEEE, 2014.

[17] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2539–2547. Curran Associates, Inc., 2015.

[19] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 365–372. IEEE, 2009.

[20] E. H. Land and J. J. McCann. Lightness and retinex theory. *JOSA*, 61(1):1–11, 1971.

[21] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[22] Z. Liu, Y. Shan, and Z. Zhang. Expressive expression mapping with ratio images. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 271–276. ACM, 2001.

[23] I. Masi, A. T. an Trãn, T. Hassner, J. T. Leksut, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? In *European Conference on Computer Vision (ECCV)*, October 2016.

[24] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, volume 1, page 6, 2015.

[25] R. Ramamoorthi and P. Hanrahan. On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *JOSA A*, 18(10):2448–2459, 2001.

[26] J. Saragih. Principal regression analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2881–2888. IEEE, 2011.

[27] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[28] Z. Shu, E. Shechtman, D. Samaras, and S. Hadap. Eyeopener: Editing eyes in the wild. *ACM Trans. Graph.*, 36(1):1:1–1:13, Sept. 2016.

[29] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[30] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337. Springer, 2016.

[31] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

[32] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Trans. Graph.*, 24(3):426–433, July 2005.

[33] Y. Wang, Z. Liu, G. Hua, Z. Wen, Z. Zhang, and D. Samaras. Face re-lighting from a single image under harsh lighting conditions. pages 1–8, June 2007.

[34] Y. Wang, L. Zhang, Z. Liu, G. Hua, Z. Wen, Z. Zhang, and D. Samaras. Face relighting from a single image under arbitrary unknown lighting conditions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1968–1984, 2009.

[35] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. *CoRR*, abs/1512.00570, 2015.

[36] F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. Expression flow for 3d-aware face component transfer. In *ACM Transactions on Graphics (TOG)*, volume 30, page 60. ACM, 2011.

[37] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in Neural Information Processing Systems*, pages 1099–1107, 2015.

[38] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *CoRR*, abs/1609.03126, 2016.