SCIENCE CHINA
Information Sciences

November 2021, Vol. 64 210101:1–210101:17
https://doi.org/10.1007/s11432-020-3236-2

• RESEARCH PAPER •
Special Focus on Visual Computing with Machine Learning

Single-view facial reflectance inference with a differentiable renderer

Jiahao GENG[1], Yanlin WENG[1,2*], Lvdi WANG[2] & Kun ZHOU[1,2]

[1]*State Key Lab of CAD&CG, Zhejiang University, Hangzhou* 310058, *China;*
[2]*ZJU-FaceUnity Joint Lab of Intelligent Graphics, Hangzhou* 310015, *China*

Received 28 May 2020/Revised 30 December 2020/Accepted 23 March 2021/Published online 25 October 2021

Abstract We introduce a deep-learning based algorithm to infer high-fidelity facial reflectance from a single image. The algorithm uses convolutional neural networks to encode the input image into a latent representation, from which a decoder and a detail enhancing network reconstruct decoupled facial reflectance (albedo, specular, and normal) as well as the environmental lighting. These decoupled components, together with a 3D facial mesh estimated from the image, are then fed into a differentiable renderer to produce a rendered facial image. This allows us to iteratively optimize the latent representation of the facial image by minimizing the image-space reconstruction loss. Experimental results show that optimizing the latent representation through the differentiable renderer can effectively reduce the discrepancy between the original image and the rendered one, leading to a more accurate reconstruction of characteristic facial features such as skin tone, lip color, and facial hair.

Keywords facial modeling, reflectance inference, differentiable renderer

Citation Geng J H, Weng Y L, Wang L D, et al. Single-view facial reflectance inference with a differentiable renderer. Sci China Inf Sci, 2021, 64(11): 210101, https://doi.org/10.1007/s11432-020-3236-2

1 Introduction

Photorealistic modeling and rendering of human faces have always been of great interest in computer graphics and vision. The use of digital avatars in professional film production, for example, has advanced to the point that people can hardly realize that the face they see on the screen is probably not flesh but polygons. In video games and the emerging VR/AR (virtual reality/augmented reality) industries, the recent development of GPUs and rendering techniques makes it possible to perform real-time, physically based rendering of vivid characters in full HD (high definition). But ironically, the cost to create such high-quality graphics assets (be they mesh models, texture maps, or animations) that can take full advantage of the modern rendering engines and hardware has also become drastically higher.

A high quality appearance model of human faces is usually composed of a set of texture maps that encode different surface properties, such as albedo, specular intensity, and geometric variations. Traditionally, these texture maps are either created manually by experienced artists or captured by professional systems (e.g., [1, 2]). Both processes are costly and time-consuming.

On the other hand, the popularization of personalized digital avatars [3, 4] over the past few years has triggered an unprecedented demand for more accessible facial capture technologies. Most approaches [5–10] estimate a coarse facial geometry and color texture from only a single input photograph, which can be easily found on the Internet or taken with a smartphone. But the result is typically too simplified for photorealistic rendering.

This paper tackles the problem of inferring high-quality facial reflectance from a single image. Specifically, given a frontal portrait taken under low-frequency lighting, we wish to recover the underlying albedo, specular, and normal which captures the detailed facial appearance in the input image.

* Corresponding author (email: weng@cad.zju.edu.cn)

Figure 1 (Color online) Given a single frontal view portrait (top left), our algorithm robustly infers a set of high-fidelity reflectance components (middle insets), including albedo, specular, and normal, as well as the environmental lighting. Together these components allow faithful reconstruction (bottom left) or relighting (the large inset) of the subject's face. The original photo on the right courtesy of Andreas Serna.

Assuming a coarse facial geometry is estimated using a state-of-the-art method, these decoupled reflectance components would be key to the photorealistic rendering of the face under new lighting environments or viewpoints, as shown in Figure 1.

To this end, we propose a deep-learning based pipeline where the input image is first encoded into a low-dimensional latent representation that captures both the skin reflectance and the lighting coefficients. From this representation, a decoder and a detail enhancing network reconstruct decoupled facial reflectance components in the form of albedo, specular, and normal. A physically based differentiable renderer then uses these components, the lighting coefficients, and the estimated 3D mesh to render a facial image. The differentiability of the whole pipeline allows us to iteratively optimize the latent representation by minimizing the image-space discrepancy between the original image and the rendered one.

Our key observation is: While using only a deep convolutional neural network to infer reflectance is possible (e.g., [11]), satisfactory results rely heavily on a complicated loss function and sufficiently high-quality and diverse training data. More importantly, at runtime, when the output of the network fails to capture certain characteristic facial features, little can be done to improve the results further. This situation is even worsened by the fact that high-quality facial reflectance datasets are rare to find—most likely due to the high cost of creating them as mentioned above. Therefore both Ref. [11] and we generate the training data by rendering from the ground truth textures of a limited number of available subjects.

Introducing a differentiable renderer in an iterative optimization grants us two immediate benefits. First, the discrepancy between the original image and the reconstruction is directly minimized both numerically and perceptually. Starting from a good initialization output by the neural network, the optimization also converges very fast. Second, the iterative optimization makes our neural network less prone to the difference between the real input and training data, significantly reducing the requirements for training data. Experiments show that, when compared with state-of-the-art techniques, our method generates a more faithful reconstruction of the skin tone and personalized features such as facial hair and lip color. We also provide the supplementary video[1)] to show our results.

This paper makes the following contributions.

- An iterative optimization pipeline for facial reflectance inference from a single image, which features a differentiable face renderer and directly minimizes the discrepancy between the input and rendered images.
- By separating the environmental lighting and training a dedicated encoder-decoder network for each reflectance component, the system can reconstruct the skin albedo more accurately.
- The overall architecture makes it feasible to train the encoder-decoder networks with synthetic data. The scarcity of high-quality facial reflectance data is less of a concern.

1) https://jiahaogeng.github.io/SVFRI.github.io/.

2 Related work

2.1 Professional facial capture

Photorealistic facial geometry and reflectance capture can be done in controlled environments with specially designed devices and algorithms. High-quality data captured by Light Stages [1,2,12,13] have been powering the creation of numerous digital characters in feature films. Using a multi-camera setup, Beeler et al. [14,15] reconstructed pore-level facial geometry using shape-from-shading techniques. Graham et al. [16] used optical and elastomeric sensors to measure facial microstructures. Such techniques can be used to create digital avatars with an extremely high level of details [17]. Probably more importantly, data captured by these lengthy and costly processes are often considered as the de facto ground truth and training examples for data-driven approaches.

2.2 Image-based facial capture

3D morphable model [18] is among the earliest studies that successfully model the shape and appearance variations of human faces as a linear combination of bases. Over the years, 3DMM has influenced a number of technical advancements (e.g., [19,20]). Egger et al. [21] provided a detailed survey of research in this direction. Thies et al. [22] and Garrido et al. [23] utilized single or multiple images to iteratively fit the linear coefficients of faces. Cao et al. [24] accelerated the process by using the Cascaded Pose Regression. With the rise of deep learning, Refs. [6–8] used pre-trained CNN-based models to regress facial coefficients. The main limitation of these techniques is that the expressiveness of a model is restricted to the linear space it spans, which may not cover certain facial variations. Tewari et al. [25] proposed a non-linear correction module to overcome the limiation of linear models. Saito et al. [26] fitted deep multi-scale features to acquire high quality skin albedo. Sengupta et al. [5] used pixel-wise image transformation to acquire skin albedo and facial shape. Tran et al. [9,10] proposed a nonlinear 3DMM. Gecer et al. [27] used GAN to train a decoder for skin color. These methods can model the non-linearness of human faces to some extent. But the reconstructed results still lack high fidelity. Yamaguchi et al. [11] modeled the facial reflectance as skin albedo, specular, and displacement components. Their learning-based method can infer facial reflectance from a single image with visually plausible details. While our studies share a similar goal, there is one important difference between these approaches: Yamaguchi et al. trained their networks by minimizing the loss defined in the reflectance texture space and directly used the output of the networks. In our pipeline, the encoder's sole job is to provide a good initialization. The decoder will work together with the renderer to iteratively minimize the image-space loss.

Huynh et al. [28] focused on the mesoscopic facial geometry and proposed a CNN-based method. Different from the above methods, Ref. [29–31] embedded the input facial image into a hidden space, which allows the facial image to be relighted directly without the inference of facial shape or reflectance.

2.3 Differentiable renderer

A traditional graphics rasterizer produces 2D images from 3D scenes. By making the rasterization differentiable, it is possible to infer 3D shapes from the rendered 2D images in reverse [32,33]. In the research of differentiable render equation, Refs. [5,8,25] assumed the Lambertian surface and approximated the radiance by spherical harmonics (SH) basis functions. Shu et al. [34] utilized this assumption to design their facial image editing framework. However, since this assumption oversimplifies the facial reflectance, and high-quality, photorealistic rendering is not their main purpose, their reconstructed results appear blurred and in many cases fail to preserve the unique facial characteristics of the original subjects. Refs. [7,27,35] based their methods on the Phong model and assumed point lights when designing their differentiable renderers. Gao et al. [36] also assumed point lights, but replaced the Phong model with bidirectional reflectance distribution functions (BRDF) [37]. Calian et al. [38] assumed faces were Lambertian surface and exploited the shadow caused by self-occlusion to infer the environment light in an input image. We also utilize shadow cues to better infer facial reflectance and lighting. However, our goal is to reconstruct high quality facial reflectance, and a Lambertian surface assumption is not sufficient. We need to take into account not only diffuse BRDF, but also the specular. Another related topic is differentiable ray tracing [39,40]. Dib et al. [39] used a raytracer in inverse rendering, but the expensive computation hinders its use in high-quality reconstruction tasks. The differentiable renderers mentioned above are commonly

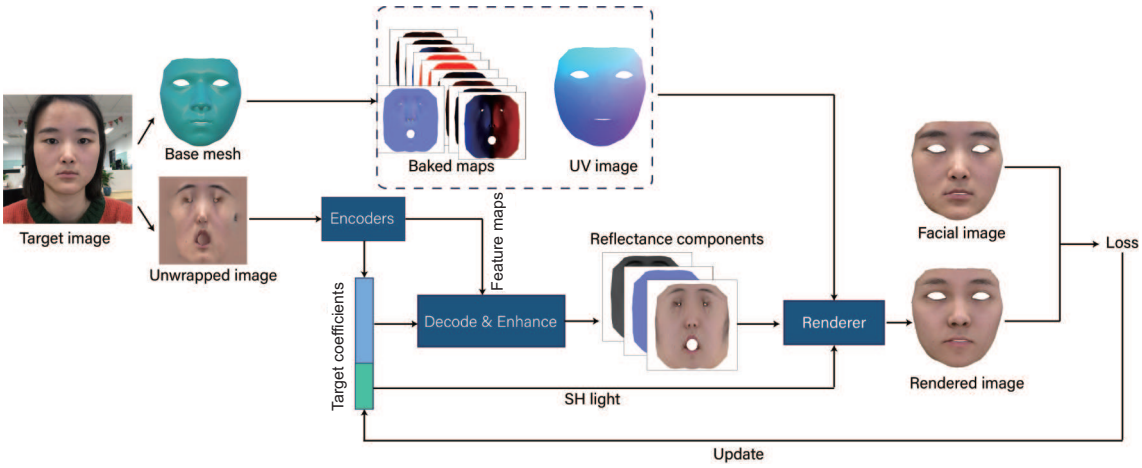


Figure 2 (Color online) The overall algorithm pipeline. The whole process can be divided into two phases. In the initialization phase, we extract a textured 3D facial mesh from the input image. Multiple encoders then convert the unwrapped color texture to a latent representation. In the optimization phase, separate reflectance components are decoded from the latent representation and fed to a differentiable renderer to produce a reconstructed image, based on which a reconstruction loss can be computed and guide the iterative updating of latent coefficients for skin reflectance and lighting via backprop.

used for two distinct purposes. One is in the training process to achieve self- or semi-supervised learning [5, 8, 25]. The other is at run time to facilitate iterative optimization [27, 35, 36, 38]. Our use of the differentiable renderer belongs to the second purpose, but differs from all the methods above in that we use a more complex facial reflectance model to suit our need for high-fidelity reconstruction. To our best knowledge, we are the first to take into account the diffuse, specular BRDF and self-occlusion in a differentiable renderer in the context of facial reflectance inference. The proposed renderer achieves a good balance between rendering realism and performance.

3 Overview

Given a single portrait image, our goal is to automatically estimate a 3D facial mesh with UV coordinates (Section 4) and a set of high quality reflectance textures, namely the albedo, specular, and normal components. Together they enable photorealistic rendering of the subject's face under arbitrary lighting conditions.

To decouple the facial reflectance textures from a single frontal portrait, we adopt an image-to-image translation framework [41] where three dedicated encoders convert the input image into low-dimensional latent representations, from which three corresponding decoders can infer the albedo, specular, and normal components respectively (Subsection 5.1).

We also employ a pretrained regression network to simultaneously estimate the lighting condition in the input image as a series of SH coefficients (Subsection 5.2).

Due to the high variability of facial images in the wild, the initial inferences made by the decoder networks are unlikely to yield a final rendering that precisely matches the original. While the quality of the inferred reflectance textures can be measured in UV space, pixelwise reconstruction loss in image space is trickier to define and largely ignored [11, 26]. In contrast to previous work, we integrate a physically based differentiable renderer that uses the inferred textures, the 3D facial mesh, as well as the estimated lighting coefficients to render an output image. This allows us to calculate differentiable image-space reconstruction loss and thus optimize the initial latent representation of the facial reflectance, greatly reducing the discrepancy between the final rendering and the input portrait (Section 6).

Figure 2 illustrates the overall pipeline of our method.

4 Fitting the facial geometry

As the first step of our pipeline, we need to calculate the template 3D mesh and the camera parameters of the input image. We adopt the morph-based algorithm by Thies et al. [22]. Although many other geometric reconstruction methods should also work.

The mesh provides a robust, albeit smooth, approximation of the actual facial geometry. More importantly, by projecting the mesh back to the original image, we can unwrap the facial region to texture space. The consistent UV parameterization effectively eliminates the geometric variations of different faces, allowing the rest of the pipeline to focus on the reflectance aspect of the skin region.

To facilitate physically based rendering in the later stage, we also precompute per-vertex self-occlusion and bent normals on the facial mesh. The self-occlusion is represented as 9-dimensional SH coefficients and used for calculating diffuse shadow [42]. The bent normal is computed in tangent space and affects the amount of specular occlusion. We use an offline raytracer to bake these two terms into additional mesh textures.

5 Decoupling the facial reflectance

The color of each pixel in a facial image is the product of multiple factors: the underlying geometry, the surface material, and the lighting condition. Photorealistic relighting of the face depends on successfully decoupling these components.

For the reflectance part, we introduce three dedicated autoencoders for albedo, specular, and normal. Each of them first encodes the facial color texture into a latent representation, then decodes to produce the respective reflectance component (Subsection 5.1). As for lighting, another convolutional neural network also takes the color texture as input but outputs an estimation of the environment lighting in the form of a 27-dimensional vector of SH coefficients (Subsection 5.2).

5.1 Albedo, specular, and normal

Inferring individual reflectance components from a color texture can be considered as an image-to-image translation problem [41]. We adopt the U-net framework [43] followed by a separate SRGAN network [44] for detail enhancement. Albedo, specular, and normal components are inferred by three dedicated networks of the above architecture without parameter sharing. Specifically, the network for inferring each of them consists of three modules: an encoder E_*, a decoder D_*, and a detail enhancer R_*. The subscript "$*$" can be "a" for albedo, "s" for specular, or "n" for normal.

5.1.1 *Encoder:* $E_*(T) \to F_*^l, z_*$

Given the unwrapped color texture T, the encoder produces a series of feature maps F_*^l via repeated 3×3 convolutional layers with stride 2 (where l denotes the layer index), each followed by a Leaky ReLU and a batch normalization. The four-channel color texture (the 4th channel being a binary mask of the skin region) is eventually encoded into a k_*-dimensional latent space vector z_*. For albedo, the input of E_a is resized to 512^2 and the dimension of z_a is 1024. For both specular and normal, the input size and output dimension are 256^2 and 512, respectively.

5.1.2 *Decoder:* $D_*(F_*^l, z_*) \to T_*$

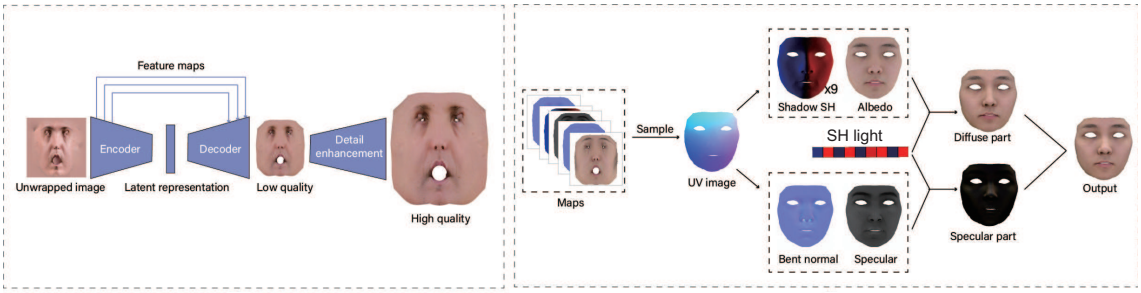
Following an almost opposite stack of layers, the decoder D_* infers the respective reflectance component T_* from both z_* and F_*^l. The multi-level feature maps skip-connected to the corresponding layers of the decoder allow it to better capture localized structural details (e.g., facial hair, pores, and moles) that are hard to encode in z_*. Note that unlike previous approaches [11, 41], we only allow the decoder to utilize feature maps from the first three layers (i.e., F_*^l where $l = \{1, 2, 3\}$). This prevents the encoding from leaning too much on the skip connections and undermining the expressiveness of z_*. The reasoning behind this modification will become apparent when we discuss optimization later (Section 6).

5.1.3 *Detail enhancer:* $R_*(T_*) \to \tilde{T}_*$

Once the decoder outputs an initial reflectance component, we employ SRGAN [44], a state-of-the-art super-resolution network, to further enhance the level of details. Our loss function combines pixel-wise MSE, VGG-based perceptual loss [45] and GAN loss. While it works well for the albedo, we need to make necessary adaptations when processing specular and normal. First, since VGG is pretrained on natural images, we discard the VGG term when computing losses for specular and normal, which tend to have totally different "color" distributions. Second, as SRGAN works in local patches, it is difficult

(a) (b)

Figure 3 (Color online) (a) Detail enhancement for normal and albedo. Top row: an area near the eyebrow; bottom row: an area near the mouth corner. From left to right: original normal, detail-enhanced normal, original albedo, detail-enhanced albedo. (b) Comparison of albedo detail enhancement results. From left to right: the low resolution input, output of a network that is trained with a downsampled target as input, output of a network using the decoder's output as training input.



(a) (b)

Figure 4 (Color online) (a) The encoder-decoder-enhancer network for albedo. Networks for specular and normal follow the same architecture. (b) Physically based rendering pipeline. Note that every step (arrow) in the pipeline is differentiable. Original photo courtesy of Andreas Serna.

for the model to distinguish certain low-resolution regions in the normal component (e.g., the patches for an eyebrow and the lips may look similar in low-resolution but drastically different in high-resolution). Therefore, when enhancing details for T_{s} and T_{n}, we also concatenate the color texture T as additional input channels. In this way, the network learns to synthesize details with regard to local facial features. Figure 3 compares the normal and albedo before and after detail enhancement.

As shown in Figure 4, the encoders E_* convert the input color texture to a 2048-dimensional vector $z = \{z_{\mathrm{a}}, z_{\mathrm{s}}, z_{\mathrm{n}}\}$, from which the decoders D_* and detail enhancers R_* can infer the individual reflectance components. In order to reconstruct the original facial image, there is one more thing to estimate lighting.

5.2 Lighting as SH coefficients

In order to model the complex lighting environments in the real world while keeping the rendering cost acceptable, we use low-order SH [42] as a powerful yet computationally efficient lighting representation, and propose a regression network that predicts the SH coefficients from the facial texture T.

The network adopts a VGG-like architecture [45]: the facial image is first resized to 256^2 before passed through 10 convolutional layers, followed by a mean pooling and a fully connected layer. The output of the network is a 27-dimensional vector, denoted as z_{e}, that corresponds to 9 SH coefficients for each of the color components.

6 Optimizing with a differentiable renderer

With the aforementioned neural networks we are able to infer the reflectance components (T_{a}, T_{s}, and T_{n}) and lighting coefficients (z_{e}) from the input image. However, the rendered image with these inference results is not guaranteed to match the original, as the reconstruction error is not directly minimized.

Therefore, we introduce a physically based differentiable renderer Φ that produces a rendered facial image $\tilde{I}$ from the initially inferred reflectance components and lighting coefficients. It allows us to compute the loss between $\tilde{I}$ and the original portrait and propagate the gradients all the way back to the latent

representations z. More formally, we can now iteratively improve z by solving the following optimization problem:

$$\begin{aligned} z' &= \arg\min_z \mathcal{L}(\tilde{I}, I) \\ &= \arg\min_z \mathcal{L}\big(\Phi(T_{\rm a}, T_{\rm s}, T_{\rm n}, z_{\rm e}), I\big), \end{aligned} \tag{1}$$

where according to the previous definition (Subsection 5.1):

$$T_{\rm a} = R_{\rm a}\big(D_{\rm a}(F_{\rm a}^l, z_{\rm a})\big), \tag{2}$$

$$T_{\rm s} = R_{\rm s}\big(D_{\rm s}(F_{\rm s}^l, z_{\rm s}), T\big), \tag{3}$$

$$T_{\rm n} = R_{\rm n}\big(D_{\rm n}(F_{\rm n}^l, z_{\rm n}), T\big). \tag{4}$$

$\mathcal{L}$ can be any appropriate loss function. We have found that the L2 norm worked well in our experiments.

6.1 Physically based differentiable renderer

The renderer determines the color of a pixel x according to the following equation:

$$\tilde{I}(x) = t_{\rm a} \cdot (1 - t_{\rm r}) \cdot L_{\rm d} + L_{\rm s}. \tag{5}$$

$L_{\rm d}$ and $L_{\rm s}$ are the diffuse and specular components of the reflected light. They are computed using two separate BRDFs (Subsections 6.1.1 and 6.1.2).

$t_{\rm a}$ and $t_{\rm r}$ are the albedo and specular properties of the subject's face (sampled from $T_{\rm a}$ and $T_{\rm r}$ using the pixel's UV coordinates respectively). The $(1 - t_{\rm r})$ term ensures the conservation of total energy.

6.1.1 *Diffuse light*

The classic rendering equation for diffuse light is

$$L_{\rm d} = \frac{1}{\pi} \int_\Omega L(\omega_i) V(\omega_i) \max(0, N \cdot \omega_i) \, {\rm d}\omega_i, \tag{6}$$

where $L(\omega_i)$ is the incident radiance in direction ω_i, V is the visibility function, and N is the surface normal at the point of interest. The diffuse BRDF models a perfectly diffuse surface that reflects incident equally in all directions, and for all ω_o, ω_i, f_r is equal to $\frac{1}{\pi}$.

The computation of (6) can be significantly accelerated by spherical harmonics approximation [42]. Given the SH basis functions $Y_i(N)$, the lighting coefficients $z_{{\rm e},i}$ (Subsection 5.2), and the precomputed self-occlusion coefficients v_i (Section 4), we can approximate L and V with $\sum_{i=0}^9 z_{{\rm e},i} \cdot Y_i(N)$ and $\sum_{i=0}^9 v_i \cdot Y_i(N)$, respectively. The clamped cosine $\max(0, N \cdot \omega_i)$ can also be expressed as $\sum_{i=0}^9 c_i \cdot Y_i(N)$, where c_i is the SH coefficient of the cosine function rotated to N. More details behind these equations can be found in [46].

To eliminate the need for actually computing the integral in (6), we utilize the product projection of SH to re-project $z_{\rm e}$ and v to a new coefficient w using the technique proposed by Snyder [47]. The rendering equation now has a much simpler form:

$$L_{\rm d} = \sum_{i=0}^{9} w_i \cdot c_i \cdot Y_i(N). \tag{7}$$

6.1.2 *Specular light*

The rendering equation for specular light starts with:

$$L_{\rm s} = \int_\Omega f_r(\omega_i, \omega_o) L(\omega_i) V(\omega_i) \max(0, N \cdot \omega_i) \, {\rm d}\omega_i, \tag{8}$$

where ω_o is the view direction, and f_r of specular BRDF follows GGX distribution [48]:

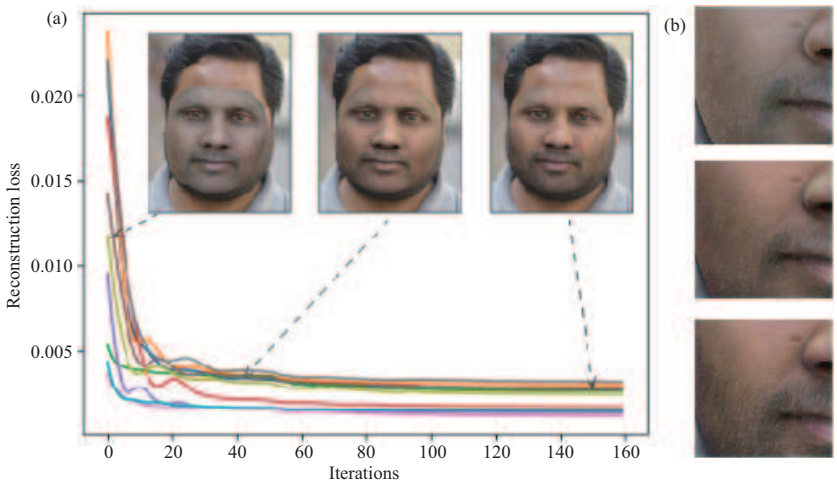


Figure 5 (Color online) (a) Reconstruction loss vs. iterations. Results of ten random input images are plotted as curves in different colors. The insets show reconstructions of one input at the 1st, 40th, and 150th iterations. (b) Zoom-ins of the insets of (a), showing reconstructed images at different optimization iterations. Note how the overall skin tone stabilizes quickly after the first few iterations before the facial hair becomes gradually clearer. Original photo courtesy of Jaymis Loveday.

$$f_r(\omega_i, \omega_o) = \frac{D(h, r)F(\omega_o, h, t_r)G(\omega_o, \omega_i, r)}{4(N \cdot \omega_i)(N \cdot \omega_o)}. \tag{9}$$

D is the normal distribution function. F is the Schlick formulation of the Fresnel term. G is the geometrical attenuation term. h is the half angle between ω_i and ω_o. r is the roughness.

Using the method by Lagarde and de Rousiers [49], it is possible to decompose the time-consuming integration of (8) into a product of two pre-integrals T_{DFG} and L:

$$L_{\mathrm{s}} = T_{\mathrm{DFG}}(N, \omega_o, r) \cdot L(\omega_i). \tag{10}$$

The D, F, G terms are independent of the light, and for a specific BRDF they are functions of only the surface roughness and the cosine between the view direction and the surface normal. Thus we can pre-integrate them into a 2D texture T_{DFG} for efficient lookups.

$$T_{\mathrm{DFG}}(N, \omega_o, r) = \frac{1}{n}\sum_{i=0}^{n} \frac{F(\omega_o, h, t_r)G(\omega_o, \omega_i, r)}{(N \cdot \omega_o)(N \cdot h)}\omega_o \cdot h. \tag{11}$$

The L term is simply the inner product of the SH light coefficients z_{e} and c:

$$L(\omega_i) = \sum_{i=1}^{n} L(\omega_i)\max(0, N \cdot \omega_i) = \sum_{i=0}^{9} z_{\mathrm{e},i} \cdot c_i \cdot Y_i(\omega_i). \tag{12}$$

The full rendering pipeline is illustrated in Figure 4.

6.2 Optimizing the latent representation

Note that the mesh vertices are fixed throughout the process. The rendering pipeline, starting from bilinearly sampling the reflectance components using the mesh's UV coordinates all the way to the final composition of diffuse and specular light, is fully differentiable.

Following (1), we iteratively render the full facial image from z, calculate the reconstruction loss, and update z using gradient descending until convergence. As shown in Figure 5, by optimizing the latent representations for both facial reflectance and lighting environment, we are able to dramatically reduce both the reconstruction loss and the perceptual discrepancy between the original image and the rendered one within a few dozens of iterations.

7 Implementation

7.1 Raw data

Our raw training data consist of 84 high quality human head models acquired from 3D Scan Store[2)] and 2957 HDR environment maps collected from the Internet and the Laval HDR Dataset [50]. The ethnic composition is 64 Caucasians, 14 Africans, and 8 Asians. The male-female ratio is 52 : 32. Each of the head models contains a detailed 3D mesh and separate albedo and specular maps in 4K resolution. Some of them also come with normal maps. For the others, we use high resolution meshes to bake normals.

Since the topologies of the models are different, we fit a template mesh to the raw meshes using deformation transfer [51] so that all of them have a consistent topology and UV parameterization.

In order to cover a wider range of skin tones, we use a histogram-based algorithm to transfer additional skin tones from the Chicago Face Database [52] to our 84 models, augmenting the number of distinct albedo maps to 4000. Note that this kind of data augmentation has been used in other studies (e.g., [11]) as well.

7.2 Training

7.2.1 *Reflectance encoders and decoders*

Successful training of the encoders E_* and decoders D_* often requires sufficiently diverse facial images with ground truth reflectance (something portraits in the wild do not have). While synthetic data can be used in training, the resulting models often failed to learn subtle details that only present in real-world images. This, notably, does not pose an issue in our case, as the optimization stage will compensate the reconstruction difference.

Indeed, we generate a total of 100000 training examples by rendering high quality face models with varying poses, camera settings, and lighting environments. Each example consists of an unwrapped color texture T and three reflectance components T_a, T_s, and T_n. T and T_a both have a resolution of 1024^2, while T_s and T_n are 512^2. Any holes in T caused by occlusion are filled using Poisson blending [53].

7.2.2 *Detail enhancing networks*

Training the detail enhancers R_* follows the way of the original SRGAN [44]. The model for albedo R_a enhances a 512^2 input to the size of 1024^2. Both R_s and R_n take 256^2 input and output at 1024^2. One caveat in preparing the low-resolution input is that they should not be downsampled versions of the high-resolution targets. Instead, we should directly use the output of the corresponding decoders D_*. Otherwise, the input in training will not be representative of those at runtime, resulting in suboptimal details (see Figure 3).

8 Results

8.1 Hardware and timings

All the results shown in the paper are produced on a workstation with an Intel Xeon E5-4650 CPU and an NVidia GeForce RTX 2080Ti GPU. We implement all the neural networks, including the differentiable renderer, in TensorFlow [54].

Training all the autoencoders takes about 12 h. The SH light regressor takes about 4 h. Training the detail enhancing networks takes about nearly 50 h.

Given an input image with a facial region of 600×800, the mesh fitting and texture baking take about 30 s. The encoders provide the initial latent representations in about 10 ms. After that the pipeline iteratively updates the latent representations. In our experiments, we use Adam [55] to optimize latent representations. The updating step size, the exponential decay rates for the 1st and 2nd moment estimates are 0.1, 0.9, 0.999, respectively. In each iteration, the forward and backward propagations need about 250 ms, in which the rendering takes about 20 ms. For all the results in this paper the optimization converges within 150 iterations, i.e., less than 40 s in total.

2) https://www.3dscanstore.com/.

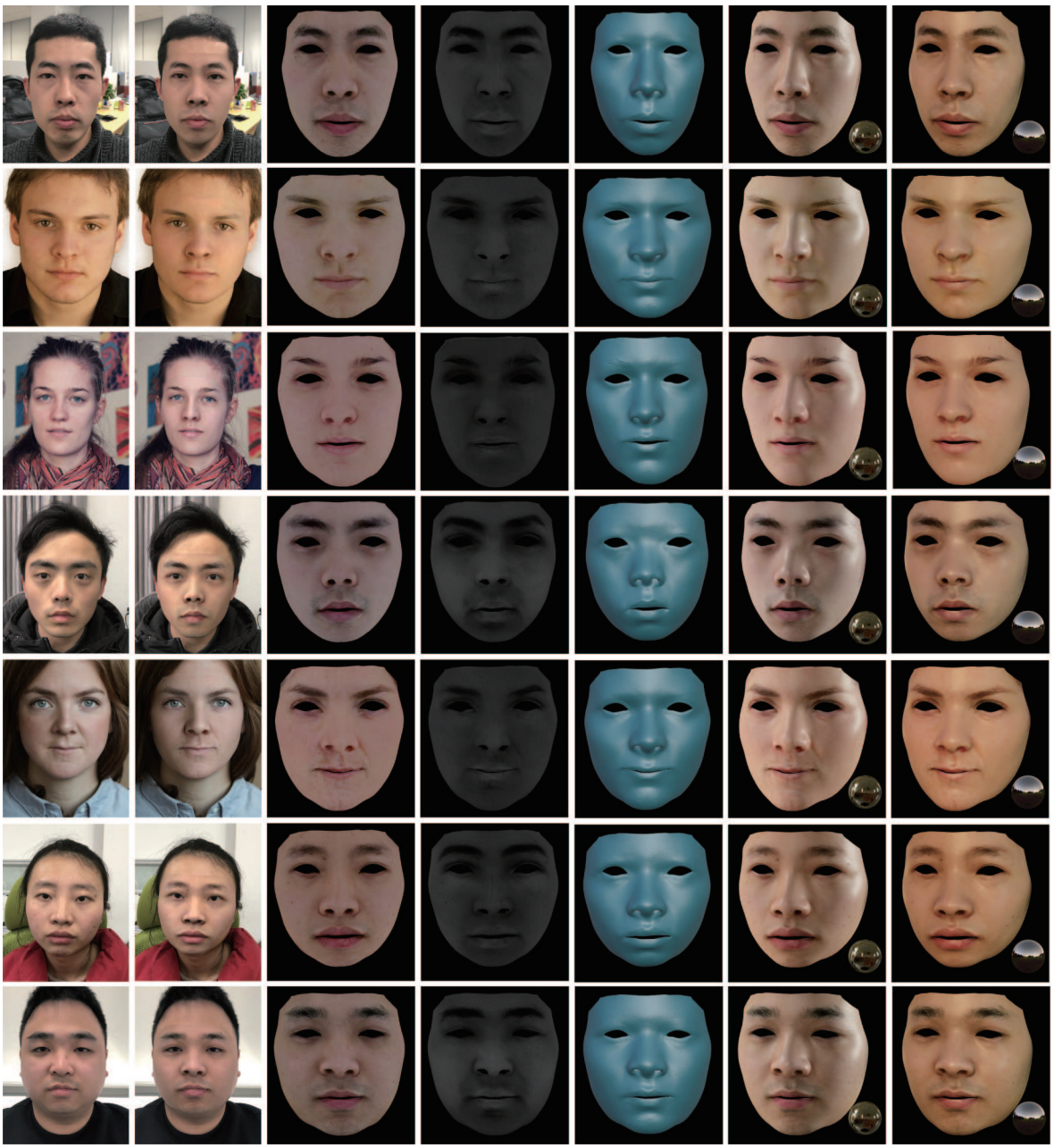


Figure 6 (Color online) Inferred reflectance components and rendering results. Each row from left to right: the input image, the reconstruction, the inferred albedo, specular, normal, and two relighting results. The original photos in the 2nd, 3rd, and 5th rows courtesy of David Kessler, Bekassine and Sterre van den Berge.

8.2 Qualitative evaluation

We have tested our algorithm extensively on images with varying subjects, lighting, image quality, etc. In Figures 1 and 6, we demonstrate each inference result in two ways.

8.2.1 *Reconstruction*

When the inferred reflectance and lighting are used in rendering the facial mesh directly on top of the original image, the rendered face blends naturally into the surrounding areas in most cases, indicating that our pipeline can faithfully reproduce the skin tone and the lighting environment.

A side-by-side comparison with the original image also shows that the characteristic facial features, such as facial hair, pores, lip color, are largely preserved with a high level of details.

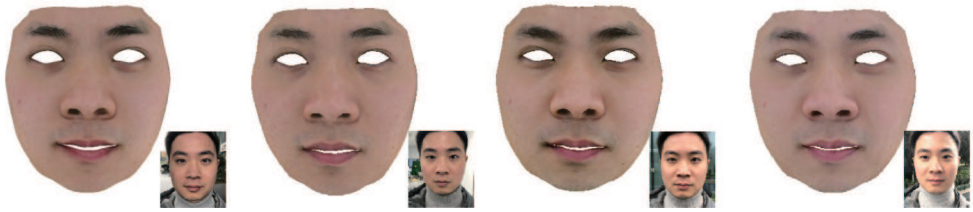


Figure 7 (Color online) Consistency of outputs under different lighting conditions.

Figure 8 (Color online) (a) is the input. (b) is the result by Yamaguchi et al. [11]. From left to right: the albedo, specular, displacement. (c) is our result. From left to right: the albedo, specular, normal. (d) Manual light tuning. When rendering the reconstructed facial images using Yamaguchi et al.'s results, we manually adjust the initially estimated SH lighting (left) until the rendered face matches the original image in overall skin tone (right). Original photo courtesy of Caique Silva.

8.2.2 *Relighting*

We also render the 3D face with a third-party real-time renderer[3)] in a new viewpoint and under different HDR image lighting. Even though the facial mesh is coarse, the final renderings look visually plausible.

8.2.3 *Consistency under different lightings*

We use photos of the same person captured under different lighting conditions to evaluate the consistency of our method with respect to lighting changes. As shown in Figure 7, our method can produce consistent output despite large lighting variation in the input photos.

For more results, please refer to the supplementary video.

8.3 Comparisons

We compare our results qualitatively with those generated by two state-of-the-art approaches [8, 11] and the ground truth.

8.3.1 *With Yamaguchi et al.'s results*

The input and output of their pipeline are almost the same as ours, except that in their reflectance model mesoscopic details are represented as displacement instead of normal, and their pipeline does not estimate the lighting in the scene. To compare the face reconstruction results of the two methods fairly, we cannot just apply the lighting estimated by our method while rendering their results. Otherwise one can expect an unnatural color shift in their results due to the highly ambiguous coupling of skin albedo and lighting. We therefore manually adjust the lighting coefficients for each of their reconstruction results until the rendered face matches the original image in overall skin tone (Figure 8).

Figure 9 compares the reconstruction and relighting results between the two methods. While the two groups of results have a comparable level of details, those generated by our method preserve the original facial features much better. It indicates that using our differentiable renderer in optimizing latent space representations can effectively infer high-fidelity facial reflectance from a photograph in the wild, even though the neural networks are trained with easy-to-acquire synthetic data.

In addition to the rendering results, we also compare the inference results component-wise. As shown in Figure 8, our method is able to recover more natural skin tone by decoupling the effect of environmental lighting. Furthermore, facial features like eyebrows, eyelashes, and lip textures are considerably clearer.

3) https://marmoset.co/toolbag/.

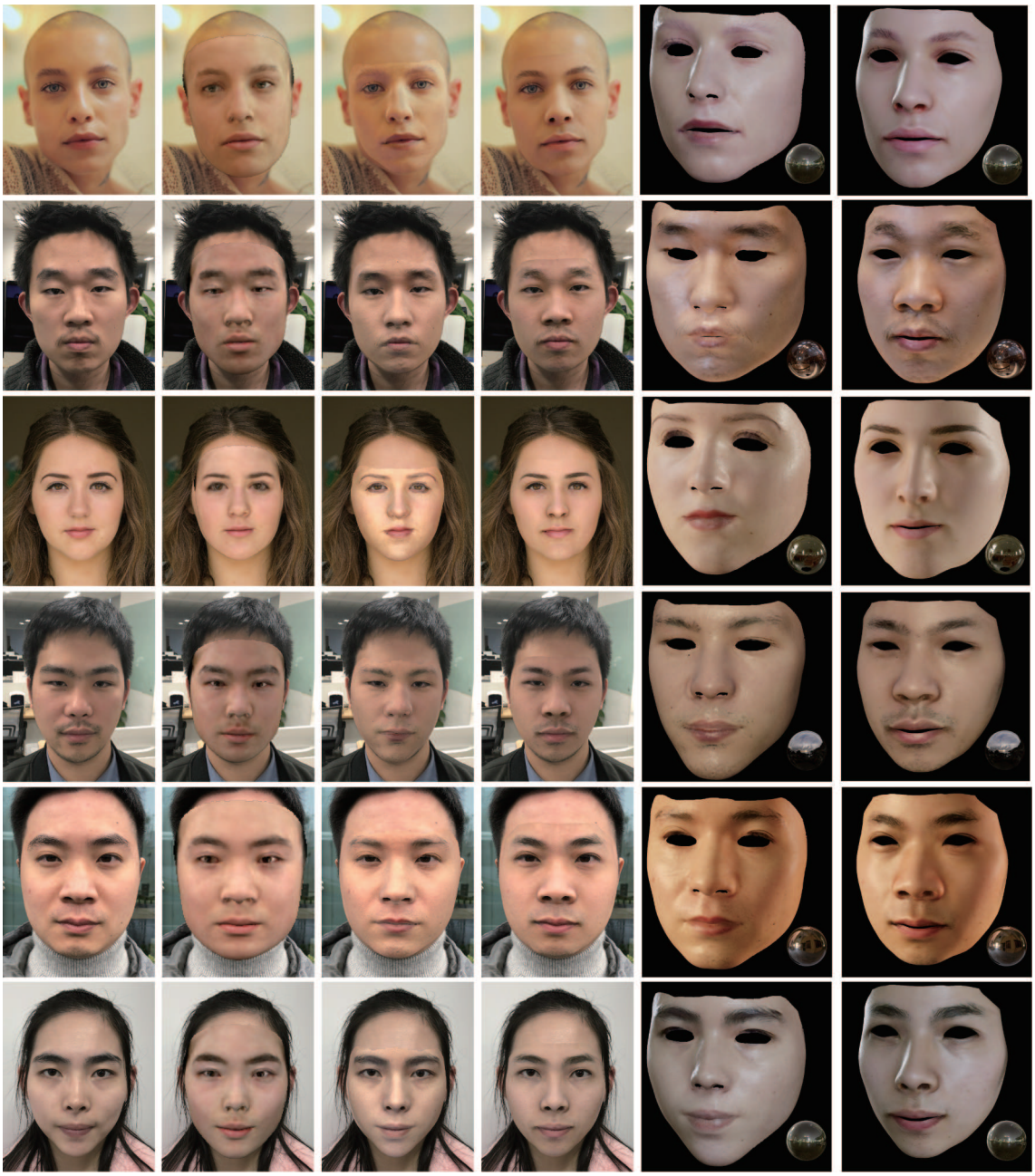


Figure 9 (Color online) Comparisons with prior approaches. Each row from left to right: the input, reconstructed by [8], reconstructed by [11], reconstructed by our method, the relighting result of [11], and our relighting result. The original photos in the 1st and 3rd rows courtesy of Caique Silva and Heath Cajandig.

8.3.2 *With PCA-based methods*

We compare our results with those produced by Deng et al.'s PCA-based facial reconstruction algorithm. As shown in Figure 9, limited by the expressiveness of the linear space, their method cannot produce enough geometric or appearance details. We also take some examples from [25] as our inputs. The comparison is shown in Figure 10. Even though they have improved on the basis of linear space, the expressiveness is still limited.

8.3.3 *With ground truth*

To further evaluate the correctness of reflectance decomposition, we utilize the synthetic data for training the autoencoders. Since all the ground truth reflectance components are known, we can compare them directly with those inferred by our pipeline. As shown in Table 1, our method is able to recover the albedo,

Figure 10 (Color online) Comparison with [25] @Copyright 2021 IEEE. Each row from left to right of (a) and (b): the input, reconstructed by [25], reconstructed by our method, the albedo of [25], the albedo of ours.

Table 1 Quantitative evaluation[a)]

	Albedo	Normal	Specular	Light
PSNR	25.10	26.61	22.68	20.29
RMSE	0.07	0.06	0.08	0.11

a) We measure the peak signal-to-noise ratio (PSNR) and the root-mean-square error (RMSE) between the ground truth and the inferred results for 100 test images.

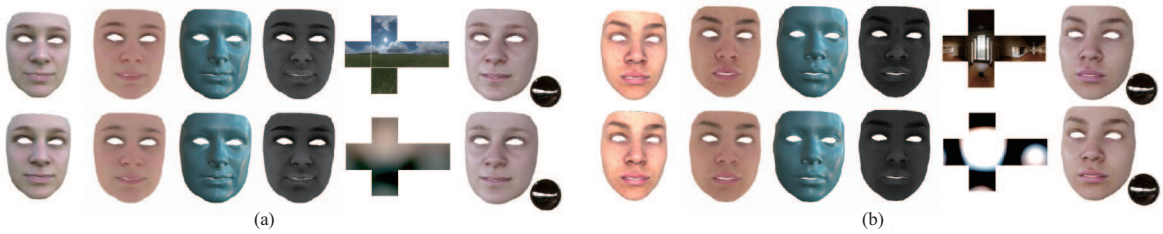


Figure 11 (Color online) Comparisons with the ground truth. The 1st row is the ground truth, the 2nd row is our result. For each row from left to right in (a) and (b): the target images or the rendering of our reconstruction, albedo, normal, specular, environmental lighting, and a relighting result.

normal, and specular well. However, our low-frequency light assumption prevents our method from recovering the light perfectly and affects the accuracy of specular slightly, but as shown in Figure 11, the inferred reflectance components are comparable to the ground truth. The reconstruction and relighting results are faithful and compelling.

8.4 Ablation study

We conduct a series of ablation experiments to evaluate some of our design choices.

8.4.1 *Skip connections*

To evaluate how much influence the skip connections have in the autoencoders, we train the autoencoders without any skip connection and compare the output. As can be seen in Figure 12 (the 2nd row), without the feature maps passing through the skip connections, the decoders struggle to localize facial features, resulting in smoother and even erroneous albedo. Such results confirm the assumption that the latent space alone cannot encode all the high-frequency, structural features of human faces, especially when the training data is insufficient or not diverse enough.

8.4.2 *Differentiable renderer*

In this experiment we use the inferred reflectance directly from the decoders' output without employing the renderer and iterative optimization.

As the results in Figure 12 (the 3rd row) indicate, the differentiable renderer and the iterative optimization of latent coefficients play a critical role in inferring the accurate skin tone and preserving important facial features. Even though each reflectance component and the lighting are initialized independently by its own encoder, the differentiable renderer successfully brings them together into a joint optimization. This makes our pipeline more capable of decoupling reflectance and lighting.

8.4.3 *Detail enhancement*

We also compare results generated without and with the detail enhancement networks. Reconstruction and relighting results are shown in Figure 13. Although the overall facial feature and skin tone look

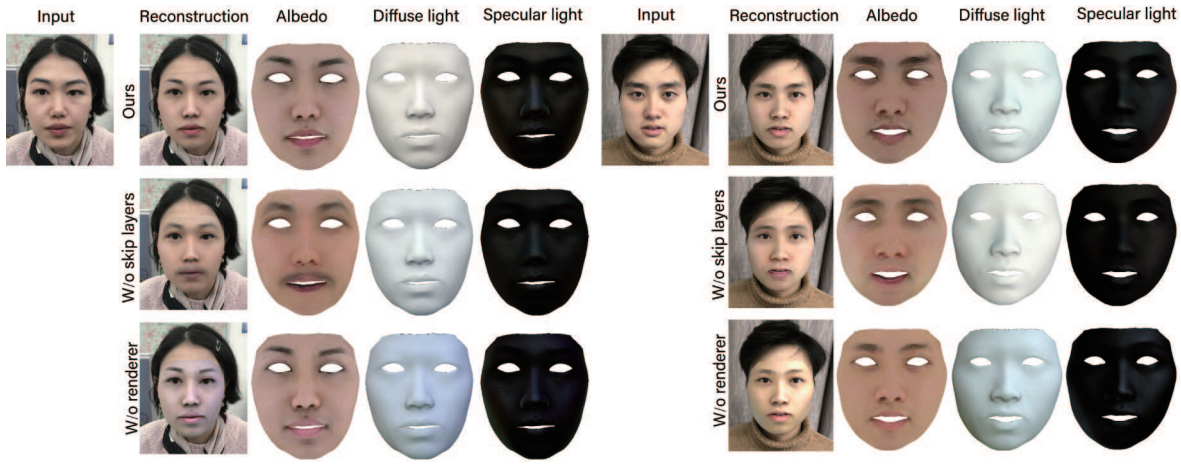


Figure 12 (Color online) Ablation study. From top down: our full pipeline, a pipeline without skip connections, a pipeline without the differentiable renderer.

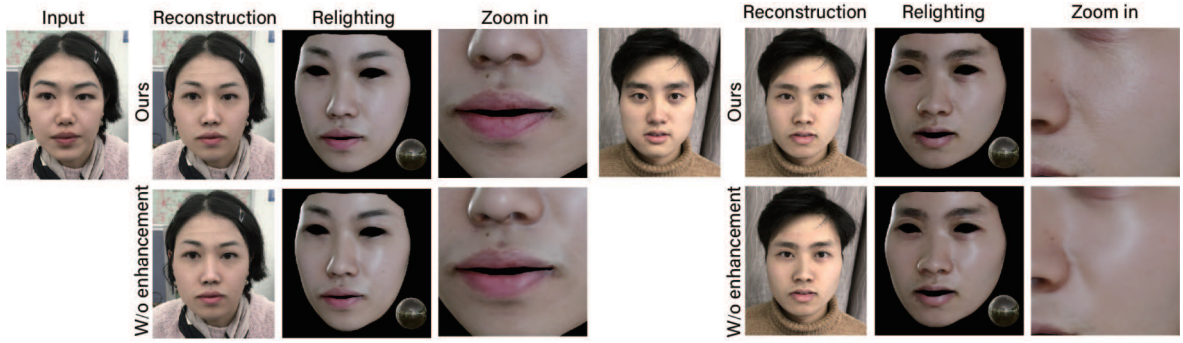


Figure 13 (Color online) Ablation study of detail enhancement.

Figure 14 (Color online) Limitation of our work. From left to right: the input image, the reconstruction result, the relighting result. Original photo courtesy of Steven Damron.

similar, when zoomed in, the detail-enhanced results clearly look more realistic.

8.5 Limitations

Our method has a few limitations. The current implementation does not handle faces with large occlusion. This includes both self-occlusion (due to non-frontal poses) and facial accessories (e.g., glasses). A data-driven approach that can robustly detect and inpaint the occluded regions in UV space is a direction that is worth investigation. Due to the limited training data, our method cannot recover faces with deep wrinkles or makeup. It is possible to expand the dataset to cover such cases.

In the reflectance components we use normal maps to represent fine-scale details, which unfortunately cannot model dramatic shape variations, e.g., thick mustache or beard. Figure 14 shows one such failure case.

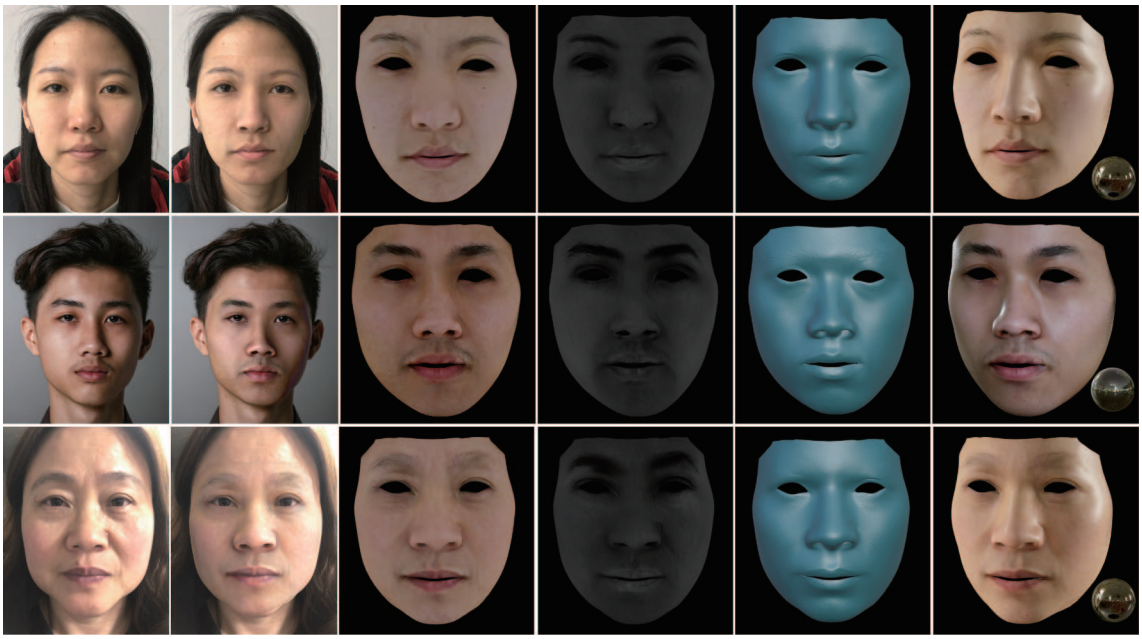


Figure 15 (Color online) Limitation of our work. From left to right: the input images, the reconstruction results, the albedo components, the specular components, the normal, and the relighting results. The original photo of the 2nd row courtesy of Imansyah Muhamad Putera.

Our pipeline assumes the environmental lighting in the scene is low-frequency and can be well approximated by SH. If it is not the case, e.g., there are sharp shadows or specularities on the face, our method may fail to completely eliminate the high-frequency shading, as shown in Figure 15. In the future, we plan to use more complex lighting models in our pipeline to support a wider range of real world environments.

Our differentiable renderer currently ignores subsurface scattering. Implementing a differentiable screen-space subsurface scattering algorithm will be our future work. Due to the baking of shadow and bent normal maps, our method does not handle gradients regarding the facial shape for now. However, it should be possible to learn a mapping from facial coefficients to these baked maps using neural networks. This would enable the design of a fully differentiable pipeline.

9 Conclusion

This paper introduces a novel method for single-view facial reflectance inference. Our algorithm pipeline makes use of the encoder-decoder neural networks. In contrast to previous approaches, the encoder is only used to give a reasonable initialization of the latent representation, and the decoder, followed by a detail enhancing network, is connected to a physically based differentiable renderer. This configuration allows us to directly minimize the reconstruction loss by iteratively updating the latent representation. Compared with the state-of-the-art, our method can better capture both the large-scale skin tone and characteristic local features of the subject's face, achieving a higher level of photorealism.

References

1 Debevec P, Hawkins T, Tchou C, et al. Acquiring the reflectance field of a human face. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, 2000. 145–156

2 Ghosh A, Fyffe G, Tunwattanapong B, et al. Multiview face capture using polarized spherical gradient illumination. ACM Trans Graph, 2011, 30: 1–10

3 Ichim A E, Bouaziz S, Pauly M. Dynamic 3D avatar creation from hand-held video input. ACM Trans Graph, 2015, 34: 1–14

4 Hu L, Saito S, Wei L, et al. Avatar digitization from a single image for real-time rendering. ACM Trans Graph, 2017, 36: 1–14

5 Sengupta S, Kanazawa A, Castillo C D, et al. SfSNet: learning shape, reflectance and illuminance of faces in the wild. 2018. arXiv:1712.01261

6 Tewari A, Zollhfer M, Kim H, et al. MoFA: model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. 2017. ArXiv:1703.10580

7 Genova K, Cole F, Maschinot A, et al. Unsupervised training for 3D morphable model regression. 2018. ArXiv:1806.06098

8 Deng Y, Yang J, Xu S, et al. Accurate 3D face reconstruction with weakly-supervised learning: from single image to image set. 2019. ArXiv:1903.08527

9 Tran L, Liu X. Nonlinear 3D face morphable model. In: Proceedings of IEEE Computer Vision and Pattern Recognition, Salt Lake City, 2018

10 Tran L, Liu F, Liu X. Towards high-fidelity nonlinear 3D face morphable model. In: Proceedings of IEEE Computer Vision and Pattern Recognition, Long Beach, 2019

11 Yamaguchi S, Saito S, Nagano K, et al. High-fidelity facial reflectance and geometry inference from an unconstrained image. ACM Trans Graph, 2018, 37: 1–14

12 Ma W C, Hawkins T, Peers P, et al. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In: Proceedings of the Eurographics Symposium on Rendering Techniques, Grenoble, 2007

13 Gotardo P, Riviere J, Bradley D, et al. Practical dynamic facial appearance modeling and acquisition. ACM Trans Graph, 2019, 37: 1–13

14 Beeler T, Bickel B, Beardsley P, et al. High-quality single-shot capture of facial geometry. ACM Trans Graph, 2010, 29: 1–9

15 Beeler T, Hahn F, Bradley D, et al. High-quality passive facial performance capture using anchor frames. ACM Trans Graph, 2011, 30: 1–10

16 Graham P, Tunwattanapong B, Busch J, et al. Measurement-based synthesis of facial microgeometry. In: Proceedings of ACM SIGGRAPH, 2013

17 von der Pahlen J, Jimenez J, Danvoye E, et al. Digital Ira and Beyond: Creating a Real-Time Photoreal Digital Actor. Technical Report, 2014

18 Blanz V, Vetter T. A morphable model for the synthesis of 3D faces. In: Proceedings of ACM SIGGRAPH, 1999

19 Kemelmacher-Shlizerman I. Internet based morphable model. In: Proceedings of IEEE International Conference on Computer Vision, 2013. 3256–3263

20 Booth J, Roussos A, Zafeiriou S, et al. A 3D morphable model learnt from 10000 faces. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 5543–5552

21 Egger B, Smith W A P, Tewari A, et al. 3D morphable face models-past, present, and future. ACM Trans Graph, 2020, 39: 1–38

22 Thies J, Zollhofer M, Stamminger M, et al. Face2face: real-time face capture and reenactment of RGB videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 2387–2395

23 Garrido P, Zollhöfer M, Casas D, et al. Reconstruction of personalized 3D face rigs from monocular video. ACM Trans Graph, 2016, 35: 1–15

24 Cao C, Hou Q, Zhou K. Displaced dynamic expression regression for real-time facial tracking and animation. ACM Trans Graph, 2014, 33: 1–10

25 Tewari A, Zollhöfer M, Garrido P, et al. Self-supervised multi-level face model learning for monocular reconstruction at over 250 Hz. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 2549–2559

26 Saito S, Wei L, Hu L, et al. Photorealistic facial texture inference using deep neural networks. 2017. arXiv:1612.00523

27 Gecer B, Ploumpis S, Kotsia I, et al. GANFIT: generative adversarial network fitting for high fidelity 3D face reconstruction. 2019. ArXiv:1902.05978

28 Huynh L, Chen W, Saito S, et al. Mesoscopic facial geometry inference using deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 8407–8416

29 Sun T, Barron J T, Tsai Y T, et al. Single image portrait relighting. ACM Trans Graph, 2019, 38: 1–12

30 Zhou H, Hadap S, Sunkavalli K, et al. Deep single-image portrait relighting. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 7194–7202

31 Meka A, Häne C, Pandey R, et al. Deep reflectance fields. ACM Trans Graph, 2019, 38: 1–12

32 Liu S, Li T, Chen W, et al. Soft rasterizer: a differentiable renderer for image-based 3D reasoning. 2019. ArXiv:1904.01786

33 Chen W, Ling H, Gao J, et al. Learning to predict 3D objects with an interpolation-based differentiable renderer. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 9605–9616

34 Shu Z, Yumer E, Hadap S, et al. Neural face editing with intrinsic image disentangling. 2017. ArXiv:1704.04131

35 Aittala M, Aila T, Lehtinen J. Reflectance modeling by neural texture synthesis. ACM Trans Graph, 2016, 35: 1–13

36 Gao D, Li X, Dong Y, et al. Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. ACM Trans Graph, 2019, 38: 1–15

37 Nicodemus F E. Directional reflectance and emissivity of an opaque surface. Appl Opt, 1965, 4: 767–775

38 Calian D A, Lalonde J F, Gotardo P, et al. From faces to outdoor light probes. In: Proceedings of Computer Graphics Forum, 2018. 51–61

39 Dib A, Bharaj G, Ahn J, et al. Face reflectance and geometry modeling via differentiable ray tracing. 2019. ArXiv:1910.05200

40 Li T M, Aittala M, Durand F, et al. Differentiable Monte Carlo ray tracing through edge sampling. ACM Trans Graph, 2019, 37: 1–11

41 Isola P, Zhu J Y, Zhou T, et al. Image-to-image translation with conditional adversarial networks. In: Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition, 2017. 1125–1134
42 Sloan P P, Kautz J, Snyder J. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. ACM Trans Graph, 2002, 21: 527–536
43 Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. 2015. ArXiv:1505.04597
44 Ledig C, Theis L, Huszár F, et al. Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 4681–4690
45 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv:1409.1556
46 Sloan P P. Stupid spherical harmonics (SH) tricks. In: Proceedings of Game Developers Conference, 2008. 42
47 Snyder J. Code Generation and Factoring for Fast Evaluation of Low-order Spherical Harmonic Products and Squares. Microsoft TechReport MSR-TR-2006-53, 2006
48 Walter B, Marschner S R, Li H, et al. Microfacet models for refraction through rough surfaces. In: Proceedings of the Eurographics Symposium on Rendering Techniques, Grenoble, 2007
49 Lagarde S, de Rousiers C. Moving frostbite to physically based rendering. In: Proceedings of SIGGRAPH 2014 Conference, Vancouver, 2014
50 Gardner M A, Sunkavalli K, Yumer E, et al. Learning to predict indoor illumination from a single image. 2017. ArXiv:1704.00090
51 Sumner R W, Popović J. Deformation transfer for triangle meshes. ACM Trans Graph, 2004, 23: 399–405
52 Ma D S, Correll J, Wittenbrink B. The Chicago face database: a free stimulus set of faces and norming data. Behav Res, 2015, 47: 1122–1135
53 Pérez P, Gangnet M, Blake A. Poisson image editing. ACM Trans Graph, 2003, 22: 313–318
54 Abadi M, Barham P, Chen J, et al. Tensorflow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, 2016. 265–283
55 Kingma D P, Ba J. Adam: a method for stochastic optimization. 2014. ArXiv:1412.6980