AvatarMe^{++}: Facial Shape and BRDF Inference with Photorealistic Rendering-Aware GANs

Alexandros Lattas, Stylianos Moschoglou, Stylianos Ploumpis,
Baris Gecer, Abhijeet Ghosh, Stefanos Zafeiriou

Abstract—Over the last years, with the advent of Generative Adversarial Networks (GANs), many face analysis tasks have accomplished astounding performance, with applications including, but not limited to, face generation and 3D face reconstruction from a single "in-the-wild" image. Nevertheless, to the best of our knowledge, there is no method which can produce render-ready high-resolution 3D faces from "in-the-wild" images and this can be attributed to the: (a) scarcity of available data for training, and (b) lack of robust methodologies that can successfully be applied on very high-resolution data. In this paper, we introduce the first method that is able to reconstruct photorealistic render-ready 3D facial geometry and BRDF from a single "in-the-wild" image. To achieve this, we capture a large dataset of facial shape and reflectance, which we have made public. Moreover, we define a fast and photorealistic differentiable rendering methodology with accurate facial skin diffuse and specular reflection, self-occlusion and subsurface scattering approximation. With this, we train a network that disentangles the facial diffuse and specular reflectance components from a mesh and texture with baked illumination, scanned or reconstructed with a 3DMM fitting method. As we demonstrate in a series of qualitative and quantitative experiments, our method outperforms the existing arts by a significant margin and reconstructs authentic, 4K by 6K-resolution 3D faces from a single low-resolution image, that are ready to be rendered in various applications and bridge the uncanny valley.

Index Terms—3D Reconstruction, Reflectance, Differentiable Rendering, Face, GAN, 3DMM, Computer Vision, Graphics

✦

1 INTRODUCTION

3D Face reconstruction from a single image is one of the most popular and well-studied problems in the intersection of computer vision, graphics and machine learning. Apart from its countless applications, it demonstrates the power of recent developments in scanning, learning, and synthesizing 3D objects [2], [3]. Recently, mainly due to the advent of deep learning, tremendous progress has been made in 3D face reconstruction from images captured even in arbitrary recording conditions (also referred to as "in-the-wild") [4], [5], [6], [7]. Nevertheless, even though the geometry can be inferred somewhat accurately, in order to render a reconstructed face in arbitrary virtual environments, much more information than a 3D smooth geometry is required, i.e., skin reflectance as well as high-frequency surface normals. In this paper, we propose a meticulously designed pipeline for the reconstruction of high-resolution render-ready faces from "in-the-wild" images captured in arbitrary poses, lighting conditions, and occlusions. A result from our pipeline is showcased in Fig. 1.

The seminal work in the field is the 3D Morphable Model (3DMM) fitting algorithm [2]. The facial texture and shape that is reconstructed by the 3DMM algorithm always lies in a space that is spanned by a linear basis, which is learned by Principal Component Analysis (PCA). The linear basis, even though remarkable in representing the basic characteristics of the reconstructed face, fails in reconstructing high-frequency details in texture and geometry. Furthermore, the

All authors are with the Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ, UK.
Corresponding author is A.L. (a.lattas@imperial.ac.uk). Other emails available at https://ibug.doc.ic.ac.uk/people.
The dataset, project page and supplemental materials are available at https://github.com/lattas/avatarme.

PCA model fails in representing the complex structure of facial texture captured in "in-the-wild" conditions. Therefore, 3DMM fitting usually fails in "in-the-wild" images. In the years that followed, 3DMM fitting was extended so that it could use a PCA model on robust features, i.e., Histogram of Oriented Gradients (HOGs) [8], for representing facial texture [9], with improved results in "in-the-wild" images. The recently proposed, Morphable Face Albedo model [10] additionally reconstructs diffuse and specular albedo with PCA. Nevertheless, these methods cannot reconstruct high-resolution facial textures. Finally, the non-linear facial and head albedo and normals models [11], [12], generate high-resolution facial textures but have not been shown in "in-the-wild" fitting.

With the advent of deep learning, many regression methods using an encoder-decoder structure have been proposed to infer 3D geometry, reflectance and illumination [3], [4], [5], [6], [7], [13], [14]. Some of the methods demonstrate that it is possible to reconstruct shape and texture, even in real-time on a CPU [3]. However, the methods [3], [5], [6], [7], [14] fail to reconstruct highly-detailed texture and shape, due to various factors such as the use of basic reflectance models (e.g., the Lambertian reflectance model), the use of synthetic data, or mesh-convolutions on colored meshes. Their results are not render-ready, and cannot be used directly in industrial rendering applications for photorealistic results. Furthermore, in many of the above methods, the reconstructed texture and shape lose many of the identity characteristics of the original image.

Arguably, the first generic method which demonstrated that it is possible to reconstruct high-quality texture and shape from single "in-the-wild" images is the recently proposed GANFIT method [4]. GANFIT can be described as an
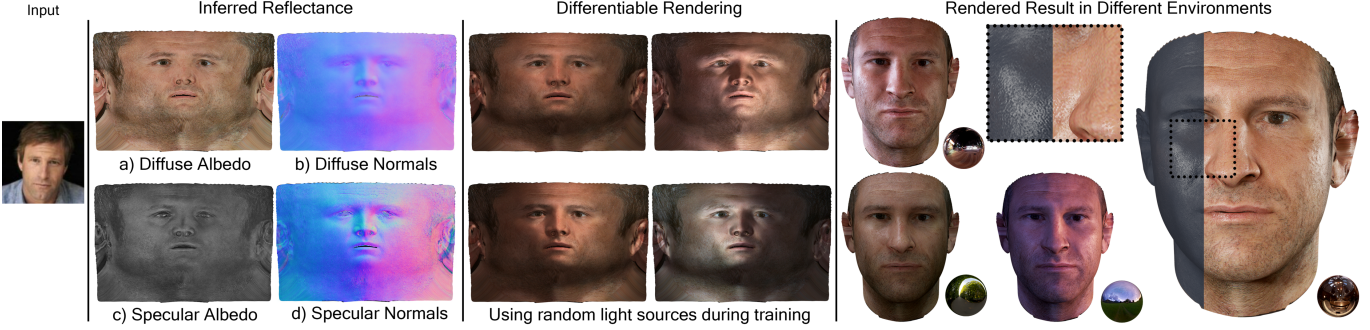
Fig. 1: From left to right: Input image from LFW [1]; AvatarMe++ predicted reflectance (diffuse albedo, diffuse normals, specular albedo and specular normals); Rendered predictions with our photorealistic differentiable rendering; Rendered reconstruction in different environments.

extension of the original 3DMM fitting strategy but with the following differences: (a) instead of a PCA texture model, it uses a GAN [15] trained on high-resolution UV-maps, and (b) in order to preserve the identity in the reconstructed texture and shape, it uses features from a state-of-the-art face recognition network [16]. However, the reconstructed texture and shape is not render-ready due to (a) the texture containing baked illumination, and (b) not being able to reconstruct high-frequency normals or specular reflectance.

Inverse rendering of an "in-the-wild" image or 3DMM-reconstructed texture to acquire its shape and its Bidirectional Reflectance Distribution Function (BRDF) parameters is an ill-posed problem and hence statistical priors are needed. Numerous works have proposed the use differentiable rendering loss while reconstructing 3DMMs [4], [14], [17], [18], [19]. However none of them photorealistically render the reconstructed face and acquire its relfectance properties, but project the reconstructed shape and use a simplistic shading model. This is mostly because of the low availability in facial reflectance data, and the inherent challenges in differentiable rendering. Recent works in differentiable rendering [19], [20] enables us to capitalize on our large reflectance dataset to implement a fast photorealistic facial differentiable rendering framework and use it in reconstructing high-resolution facial shapes and BRDF.

Early attempts to infer photorealistic render-ready information from single "in-the-wild" images have been made in some works [13], [21], [22], [23]. Arguably, some of the results showcased in the above papers are of decent quality. Nevertheless, the methods do not generalize since they directly manipulate and augment the low-quality and potentially occluded input facial texture. As a result, the quality of the final reconstruction always depends on the input image. Even more, the employed 3D model may not be very representative, and a very small number of subjects (e.g., 25 [23], 122 [13]) were available for training for the high-frequency details of the face. While closest to our work, these approaches focus on easily creating a digital avatar rather than high-quality render-ready face reconstruction from "in-the-wild" images, which is the goal of our work.

We present an elaborate methodology for high-quality 3D facial geometry and reflectance reconstruction from a single "in-the-wild" image. In particularly, we collect a big dataset of facial reflectance, and use an end-to-end reflectance inference network with a photorealistic differentiable rendering loss. Our method builds upon recent reconstruction methods (e.g., GANFIT [4]) and applies super-resolution and domain-adaption algorithms to GAN-generated generated high-quality facial textures. We show that this methodology is superior to the previous state-of-the-art (e.g. [13], [23]), who apply algorithms for high-frequency estimation of the original input, which could be of low quality and are affected by environment illumination. We demonstrate that it is possible to produce render-ready faces from arbitrary faces (pose, occlusion, etc.) including paintings, which can be realistically re-rendered in any environment. Specifically, our contributions are:

- A dataset of facial reflectance and geometry collected using state-of-the-art methods from over 200 subjects, which is now available to the community.
- A differentiable rendering framework that fully utilizes both diffuse and specular reflectance data, and enables the fast approximation of subsurface scattering and occlusion shadows.
- An image-translation network that transforms a facial geometry with baked-in illumination to diffuse and specular albedo and normals using the above differentiable rendering framework.
- An end-to-end algorithm for reconstructing high-resolution 3D faces including their shape and BRDF, from a single "in-the-wild" image.

This work is an extension of AvatarMe [24]. Compared to the conference paper, AvatarMe++ adds: a) a photorealistic differentiable rendering method; b) online data augmentation of the training data in randomized illumination environments; c) a single image-translation network for BRDF inference, that utilizes a stochastic rendering loss, geometrical and global information; d) an extensive ablation study and experiments for the aforementioned additions.

2 RELATED WORK

2.1 Facial Geometry and Reflectance Capture

Debevec et al. [25] first proposed to employ a specialized light stage setup to acquire a reflectance field of a human face for photorealistic image-based relighting. Weyrich et al. [26] used an LED sphere and 16 cameras to densely record facial reflectance and computed view-independent estimates of facial reflectance from the acquired data, including per-pixel diffuse and specular albedos, and per-region

specular roughness parameters. These initial works required cumbersome and impractical dense capturing.

Ma et al. [27] introduced polarized spherical gradient illumination (using an LED sphere) for efficient acquisition of separated diffuse and specular albedos and photometric normals of a face using just eight photographs. They demonstrated high quality facial geometry, including skin mesostructure as well as realistic rendering with the acquired data. However, the method was restricted to frontal viewpoint acquisition, as the polarization pattern used on the LED sphere was view-dependent. Subsequently, Ghosh et al. [28] extended polarized spherical gradient illumination for multi-view facial acquisition by employing two orthogonal spherical polarization patterns. This allows capturing separated diffuse and specular reflectance and photometric normals from any viewpoint around the equator of the LED sphere. Until today, it can be considered the state-of-the art in terms of high quality facial capture. In the recent years, significant progress has also been made in passive facial capture, from high quality facial geometry capture [29] to even detailed facial appearance estimation [30]. However, the quality of the acquired data with such methods is lower compared to active illumination techniques.

Recently, Kampouris et al. [31] demonstrated how to utilize unpolarized binary spherical gradient illumination for estimating separated diffuse and specular albedo and photometric normals using color-space analysis. The method does not require polarization and hence needs half the number of photographs compared to polarized spherical gradients. Moreover, it enables completely view-independent reflectance separation, making it faster and more robust for high quality facial capture [32]. For our work, we use the unpolarised active illumination-based multi-view facial capture method [31], [32] for acquiring high quality facial reflectance data in order to build our training data.

2.2 Facial Geometry and Texture Estimation

Over the years, numerous methods have been introduced in the literature that tackle the problem of 3D facial reconstruction from a single input image [2], [9], [33], [34], [35], [36], [37], [38]. Early methods required a statistical 3DMM both for shape and appearance, usually encoded in a low dimensional space constructed by PCA [2], [9], [33]. A 3DMM is typically fitted on a 2D image using an energy based cost optimization with respect to the model's identity and expression parameters as well as the parameters of the camera and scene illumination, as thoroughly explained in the 3DMM review of [34]. Moreover, many approaches have tried to leverage the power of Convolutional Neural Networks (CNNs) to either regress the latent parameters of a PCA model [35], [36] or utilize a 3DMM to synthesize images and formulate an image-to-image translation problem using CNNs [37], [38]. Similar works have also modeled complete head topologies [39], [40], [41], [42]. Finally, the recent Morphable Face Albedo Model [10] separately models diffuse and specular albedo with a PCA model.

2.3 Image-to-Image Translation

Image-to-image translation refers to the task of translating an input image to a designated target domain (e.g., turning sketches into images, or day into night scenes). With the introduction of GANs [43], image-to-image translation improved dramatically [44], [45]. Recently, with the increasing capabilities in the hardware, image-to-image translation has also been successfully attempted in high-resolution data [46]. In this work, we improve on pix2pixHD [46], by building an image-translation network that incorporates the photorealistic differentiable rendering of its results, so that a rendering loss can be introduced. Our model successfully learns to disentangle relfectance components from textures with rendered illumination and occlusion shadows.

2.4 Differentiable Rendering

Multiple works in the past have attempted to differentiate rendering models to solve inverse rendering problems with limited success [47]. OpenDR [48] was the first complete framework in Python for differentiable rendering, built using an auto-differentiation framework. Neural 3D Mesh Renderer [49] introduced a rasterization approximation which enables differentiation for non-occluded gradients. TF Mesh Renderer [19] introduced a differentiable rasterizer which interpolates per-vertex attributes in viewspace, using positive and negative barycentric coordinates to overcome discontinuities. SoftRasterizer [50] introduced a fully differentiable rendering framework, by using a probabilistic rasterization function with an aggregation function for z-buffering. It significantly improves the gradient flow over [48], [49] and can be used with differentiable local-illumination models. Similarly, [51] separate the fore- and background rasterization and use barycentric coordinates to propagate the gradient only for the foreground pixels. For global illumination models, Li et al. [52] introduced a differentiable ray tracer, which uses an edge-sampling-based method to provide a continuous rendering function and importance sampling to improve on performance. Moreover, Loubet et al. [53] introduced spherical rotations that remove the discontinuities, with respect to visibility, cameras, lights, and geometry.

Several complete frameworks exist that combine deep learning with differentiable rendering. TF Mesh Renderer [19] is integrated with Tensorflow, which also includes a library for graphics and differentiable rendering. Kaolin [54] is a library of Pytorch implementations including [14], [50]. Finally, Pytorch3D [20] is a complete modular differentiable rendering framework, based on SoftRasterizer, with additional modules for shading, performance, and compatibility improvements. To the best of our knowledge, our method is the first to show fast photorealistic differentiable rendering for human skin. We extend Pytorch3D framework [20] for accurate facial skin diffuse and specular reflection, self-occlusion and subsurface scattering approximation, and integrate it with a high-resolution image-translation GAN.

2.5 Facial BRDF Estimation

Many approaches have been successful in acquiring the reflectance of materials from a single image, using deep networks with an encoder-decoder architecture [55], [56], [57], [58]. However, they only explore planar surfaces in a constrained environment, typically assuming a single point-light source. Similar principles have also been successfully

applied to "in-the-wild" outdoor images [59]. Early applications on human faces [14], [60] used image translation networks to infer facial reflection from an "in-the-wild" image, producing low-resolution results. Recent approaches attempt to incorporate additional facial normal and displacement mappings resulting in representations with high frequency details [13]. Although this method demonstrates impressive results in geometry inference, it tends to fail in conditions with challenging illumination and extreme head poses, and does not produce re-lightable results. Saito et al. [22] proposed a deep learning approach for data-driven inference of high resolution facial texture map of an entire face for realistic rendering, using an input of a single low-resolution face image with partial facial coverage. This has been extended to inference of facial mesostructure, given a diffuse albedo [21], and even complete facial reflectance and displacement maps besides albedo texture, given a partial facial image as input [23]. The above methods are the most related to our work and achieve the creation of digital avatars from "in-the-wild" images. In this work, we show high quality facial reflectance reconstruction, from such images and existing models and datasets.

Various alternative paradigms that produce renderable human faces have been recently proposed. [61] introduce a coarse-to-fine optimization utilizing differentiable ray-tracing for facial geometry and albedo reconstruction. [62] reconstruct neural face reflectance fields that enable rendering with complex physical effects. Finally, [63] generate dynamic head textures and shapes with an encoder-decoder, that are relightable and animatable from VR-headset views.

The state-of-the-art facial 3DMM fitting method GANFIT [4] uses a GAN-generated texture with an iterative optimization method of the 3DMM's weights. Each iteration utilizes lambertian differentiable rendering and a deep face recognition network, achieving high quality texture with fine identity characteristics. In this work, we use an image-translation network that learns to disentangle the reflectance components reconstructed from a 3DMM fitting method, guided by our high-resolution photorealistic differentiable rendering loss. Our facial geometry and spatially-varying BRDF textures are high-resolution and ready to be rendered with high-quality photorealistic results.

3 DATA ACQUISITION

3.1 Training Data Capturing Setup

We use our facial capturing system [64], comprising of an LED sphere with 168 lights (partitioned into two polarization banks) and 9 DSLR cameras. Half of the LEDs on the sphere are vertically polarized (for parallel polarization), and the other half are horizontally polarized (for cross-polarization) in an interleaved pattern. On this setup, we can employ the state-of-the-art method of [28] for capturing high resolution pore-level reflectance maps of faces. On the same apparatus, we remove the polarizers and use the color-space analysis for diffuse-specular separation and multi-view facial capture [31], [32], to acquire reflectance of similar quality (Fig. 2). This is our preferred method, since it requires less than half of the data captured (hence reduced capture time) and provides a simpler setup without polarizers, enabling the acquisition of larger datasets.

Following the capture and diffuse-specular separation with [31], we develop a pipeline to prepare the training data as follows: A base geometry is reconstructed from the full-on images using structure-from-motion [65] and multi-view stereo [66]. The geometry is fitted with landmarks [67] using its rendering and registered to a template mesh [68]. Finally, the camera-space reflectance is projected to a uniform UV map, manually constructed for minimal distortion, at a resolution of $\hat{H}, \hat{W} = 6144, 4096$ pixels, as shown in Fig. 2.

3.2 Data Collection

In this work, we capture faces of over 200 individuals of different ages and characteristics under 7 different expressions. We curate a dataset called RealFaceDB, by sampling square 512×512 pixels patches, of (a) diffuse albedo $\mathbf{A_D}$, (b) specular albedo $\mathbf{A_S}$, (c) diffuse normals $\mathbf{N_D}$, (d) specular normals $\mathbf{N_S}$ and (e) shape $\mathbf{S}$ in UV space. The patches are anonymized, shuffled, and only the correspondence between same patches of different type is kept. We have made RealFaceDB public for the research community [1]. The captured subjects are 63.2% male and 36.8% female; 61.1% White, 26.3% Asian, 5.5% Black and 7.1% other; 55.8% 0-25 years old, 37.3% 25-40 years old, 6.9% over 40 years old.

Diff.Alb. $\mathbf{A_D}$ Spec.Alb. $\mathbf{A_S}$ Diff.Nor. $\mathbf{N_D}$ Spec.Nor. $\mathbf{N_S}$ Shape $\mathbf{S}$

Fig. 2: Example of a captured subject data using [31], [32], registered and projected to a standard UV topology.

3.3 Data Augmentation

In our initial method AvatarMe [24], we only rendered our dataset in the environment of the target 3DMM. In AvatarMe++ (Sec. 4.7.2), we can augment the training data by rendering them in random environments, centered around the target environment. This does not only improve the model's accuracy on the target environment, but enables it to successfully generalize to other domains (Figs. 11,13,14). Moreover, our captured dataset is imbalanced on race, due to the demographic limitations in our area and the immobility of our capturing device. In an attempt to balance the dataset, we use the albedo measurements of [69] to augment our acquired albedos. Specifically, we use a patch from the forehead of the captured diffuse albedo, match it to the closest albedo from [69] and then apply a transformation to another albedo from their chart. Since all our albedos are in the same UV space, a manually constructed "skin" mask ensures that common non-skin areas remain unchanged.

4 METHOD

4.1 Overview

We aim to reconstruct the shape and reflectance properties of a subject from a single "in-the-wild" image, that can be used for photorealistic rendering. These are the shape $\mathbf{S}$,

---

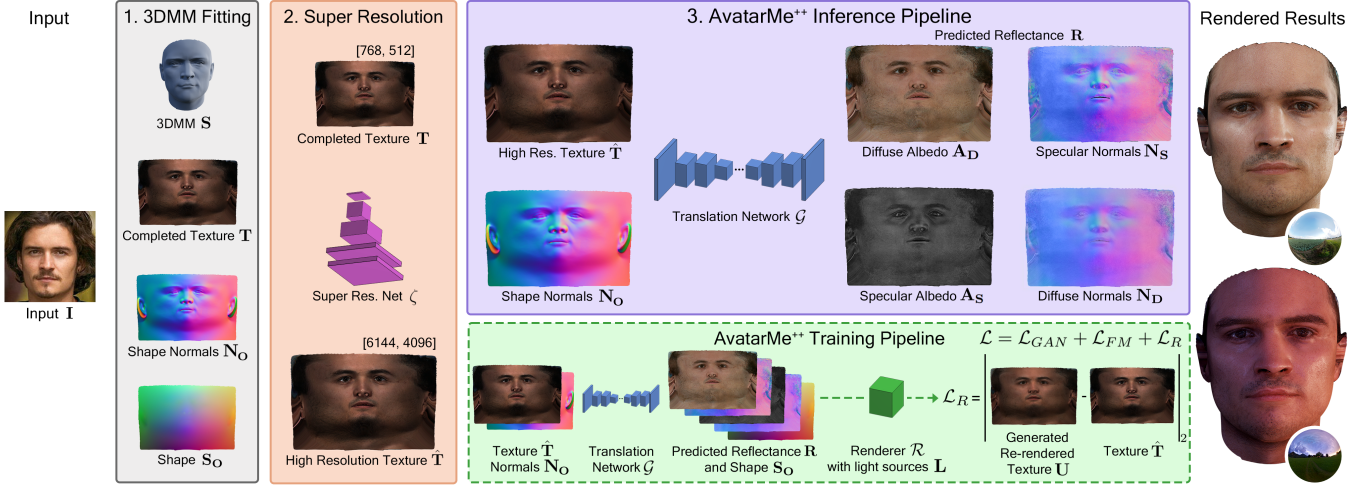1. Dataset available at https://github.com/lattas/avatarme .

Fig. 3: Summary of the AvatarMe++ method: Given an "in-the-wild" image $\mathbf{I}$, we first fit a 3D Morphable Model (3DMM) to acquire the shape $\mathbf{S_O}$, texture $\mathbf{T}$ and shape normals $\mathbf{N_O}$ in UV space. Then, we upscale the texture $\mathbf{T}$ using a state-of-the-art super resolution network ζ, trained on synthetic data rendered in the texture's $\mathbf{T}$ domain. A deep network $\mathcal{G}$ is then used to transform the upscaled texture $\hat{\mathbf{T}}$ and normals $\mathbf{N_O}$) to reflectance maps, namely the diffuse albedo $\mathbf{A_D}$, specular albedo $\mathbf{A_S}$, diffuse normals $\mathbf{N_D}$ and specular normals $\mathbf{N_S}$. The deep image-translation network is trained on high-resolution captured facial BRDF, which we have made public as RealFaceDB. To train AvatarMe++, we define a photorealistic differentiable rendering module $\mathcal{R}$, with subsurface-scattering and self-occlusion approximation. During training, $\mathcal{R}$ is used to create synthetic data pairs, by rendering the captured data in the target's environment $\mathbf{L}$ and random ones. The loss $\mathcal{L}$ used during training, is comprised of an adversarial loss $\mathcal{L}_{GAN}$, a feature-matching loss $\mathcal{L}_{FM}$ and our photorealistic differentiable loss $\mathcal{L}_R$. The complete high resolution (up to $6\text{k} \times 4\text{k}$) BRDF maps can be used for photorealistic rendering, while the specular normals $\mathbf{N_S}$ can be used to enhance the 3DMM's geometry.

diffuse albedo $\mathbf{A_D}$, specular albedo $\mathbf{A_S}$, diffuse normals $\mathbf{N_D}$ and specular normals $\mathbf{N_S}$, facial reflectance components that can be used for photorealistic rendering (i.e. [27], [28]).

As shown in Fig. 3, we fit a 3DMM to an "in-the-wild" image (Sec. 4.2), obtaining a 3D facial geometry $\mathbf{S}$ with a texture $\mathbf{T}$, which is typically of low-resolution and contains baked-in illumination and shadows. We upsample $\mathbf{T}$ using a deep super-resolution network trained on textures of the same domain as the ones from the 3DMM (Sec. 4.4). Then, the AvatarMe models (Sec. 4.5) or the AvatarMe++ model (Sec. 4.7) transform the upsampled texture $\hat{\mathbf{T}}$ into the reflectance components $\mathbf{A_D}, \mathbf{A_S}, \mathbf{N_D}, \mathbf{N_S}$. AvatarMe utilizes four image-translation networks for the above transformation. AvatarMe++, an extension to AvatarMe, uses a combined image-translation network and incorporates a photorealistic differentiable renderer (Sec. 4.6), achieving improved results, generalization and computational speed.

4.2 Initial Geometry and Texture Estimation

The first step of our method is the acquisition of a facial shape $\mathbf{S}$ and texture $\mathbf{T}$ from a single image. In our approach, we adopt the 3DMM-based fitting method GANFIT [4]. Apart from the usage of deep identity features, GANFIT synthesizes consistent realistic UV texture maps, using a GAN as a statistical representation of the facial texture. Alternatively, our method and training can easily be modified to use other methods (i.e. [51], [70]) as long as they produce a consistent shape and texture (results in Fig. 11). We reconstruct the initial base shape $\mathbf{S} \in \mathbb{R}^{n \times 3}$ of n vertices and texture $\mathbf{T} \in \mathbb{R}^{W \times H \times 3}$ from the input image $\mathbf{I}$ as follows:

$$\mathbf{T}, \mathbf{S} = \mathcal{F}(\mathbf{I}) \qquad (1)$$

where $\mathcal{F} : \mathbb{R}^{k \times m \times 3} \mapsto \mathbb{R}^{W \times H \times 3}, \mathbb{R}^{n \times 3}$ denotes the GANFIT reconstruction method for an $\mathbf{I} \in \mathbb{R}^{k \times m \times 3}$ arbitrary sized image, and n number of vertices on a fixed topology.

The acquired shape is of adequate quality for rendering, however the texture $\mathbf{T}$ is of limited resolution and most importantly, contains significant baked-in illumination and self-shadows. We proceed by upsampling $\mathbf{T}$ in the next section, and then discuss the ways to learn the disentanglement of the baked-in illumination in $\mathbf{T}$, into high-resolution spatially-varying reflectance parameter UV maps.

4.3 3DMM Capturing Environment Estimation

A drawback of the texture modeled by typical 3DMMs is that they reproduce the environment conditions of their training data (i.e. reflection and shadows), which inhibits rendering. In our case, the textures generated by [4] contain sharp highlights and shadows, made by point-light sources, as well as environment illumination. In order to alleviate this problem, we model the illumination conditions of the dataset used in [4] and synthesize UV maps with the same illumination, in order to train a transformation between texture with baked-illumination $\mathbf{T}$ and reflectance maps $\mathbf{R}$.

Initially, we acquire random texture and mesh outputs from GANFIT, by fitting random facial images. Using a cornea model [71], we estimate the average view direction $\mathbf{v_d}$, the direction for the apparent 3 point light sources $\mathbf{l_d}$ and their intensity $\mathbf{c_d}$ used in the 3DMM texture data, including an environment color defined as $\mathbf{e_d}$. Then, we render our acquired subjects (Section 3.2), as if they were samples from the dataset used in the training of the 3DMM used, in our case [4]. In this way, we also have accurate ground truth of their reflectance. We compute a rendering ρ for each subject with reflectance $\mathbf{R} = \{\mathbf{A_D}, \mathbf{A_S}, \mathbf{N_D}, \mathbf{N_S}\}$, directly in a UV map $\mathbf{T_d}$, using the predicted environment parameters. We denote this rendering process by $\rho : \mathbf{R}, \mathbf{c_d}, \mathbf{l_d}, \mathbf{v_d}, \mathbf{e_d} \mapsto \mathbf{T_d} \in \mathbb{R}^{\hat{W} \times \hat{W} \times 3}$ which renders the captured reflectance to the domain of the 3DMM textures with baked illumination.

The above estimation of the target 3DMM environment can be further improved, after an initial training of the AvatarMe^{++} network $\mathcal{G}$. We acquire a number of random 3DMM-generated textures in the target environment and use $\mathcal{G}$ to acquire their reflectance. Then, we initialize our differentiable renderer (Sec. 4.6) with the calculated parameters $(\mathbf{v_d}, \mathbf{l_d}, \mathbf{c_d})$. In an iterative process, we render the acquired reflectance maps in the current best environment parameters and compare the rendering with the initial 3DMM-generated textures, using an L1 rendering loss. Since the renderer is differentiable, we use gradient descent to further optimize the estimated parameters $\mathbf{v_d}, \mathbf{l_d}, \mathbf{c_d}$. Then, we can re-train $\mathcal{G}$ in the optimized environment estimation.

4.4 Super Resolution

Although the texture $\mathbf{T} \in \mathbb{R}^{W \times H \times 3}$ from GANFIT [4] has reasonably good quality and resolution ($H, W = 768, 512$) it is below par compared to artist-made render-ready 3D faces. On the contrary, the facial reflectance textures we capture in (Sec. 3.1) are in a resolution of $\hat{W}, \hat{H} = 6144, 4096$. Therefore, we train a state-of-the-art super resolution network, RCAN [72], that increases the resolution of $\mathbf{T} \in \mathbb{R}^{W \times H \times 3}$ to $\mathbf{T} \in \mathbb{R}^{\hat{W} \times \hat{H} \times 3}$, using $\times 4$ upsampling twice. We define the super-resolution network ($\zeta : \mathbb{R}^{H_p \times H_p \times 3} \mapsto \mathbb{R}^{\hat{H}_p \times \hat{H}_p \times 3}$), which is trained on square patches of $H_p = 64 \mapsto \hat{H}_p = 512$, given the large size of the results and the low number of available training data. At testing time, the whole texture from GANFIT $\mathbf{T}$ is up-scaled by the following:

$$\hat{\mathbf{T}} = \zeta(\mathbf{T}) \qquad (2)$$

To train the super-resolution ζ to upsample the textures generated from Eq. 1, we use our estimation of GANFIT's illumination environment (Sec.4.3) to render our captured data with the same environment.

4.5 Reflectance Inference with AvatarMe

The significant issue of the texture $\mathbf{T}$ produced by typical 3DMMs is that they are trained on data with ambient illumination (i.e. reflection, shadows), which they reproduce. GANFIT-produced textures contain sharp highlights and shadows, made by strong point-light sources, as well as baked environment illumination, which prohibits photorealistic rendering. In order to alleviate this problem, we first model the illumination conditions of the dataset used in [4] and then synthesize UV maps with the same illumination $\mathbf{T_d}$ (Sec. 4.3). We can then use the pairs of $\mathbf{T_d}$ with the ground truth reflectance data $(\mathbf{A_D}, \mathbf{A_S}, \mathbf{N_D}, \mathbf{N_S})$ to train image-translation networks in a supervised manner. Finally, following [46], we formulate the networks' objective as:

$$\min_G \left(\max_D \mathcal{L}_{GAN}(G, D_k) + \lambda_{FM} \mathcal{L}_{FM}(G, D_k) \right) \qquad (3)$$

where $\mathcal{L}_{GAN}(G, D_k)$ is the sum of adversarial loss and $\mathcal{L}_{FM}(G, D_k)$ is the feature matching loss, for all 3 discriminators of pix2pixHD [46]. The feature matching term is controlled by λ_{FM}.

We find that we can improve learning by giving the network an insight into the geometry of the reconstructed shape. In this manner, for each training data pair, we extract the shape $\mathbf{S_O}$ and its normals $\mathbf{N_O}$ in the same UV parameterization as the textures, and complete the 2D RGB texture, by using bilinear interpolation in the 2D UV space. Below we describe the baseline pipeline, *AvatarMe*, in which 4 image-translation networks are used, to first acquire the diffuse albedo $\mathbf{A_D}$ from the upsampled reconstructed texture $\hat{\mathbf{T}}$ and then the specular albedo $\mathbf{A_S}$, diffuse normals $\mathbf{N_D}$ and specular normals $\mathbf{N_S}$ from the diffuse albedo $\mathbf{A_D}$. The better performing pipeline *AvatarMe^{++}*, with a single rendering-aware network is described in Sec. 4.7.

4.5.1 Diffuse Albedo Extraction

We formulate de-lighting as a domain adaptation problem and train an image-to-image translation network. To do this, we follow two strategies different from the standard image translation approaches. Firstly, the shading and occlusion on the skin surface is geometry dependent and thus use both the texture and geometry of the 3DMM as input to the network. We find that this improves not only the network's accuracy, but also the consistency between patches. Instead of using the 3-channel shape texture $\mathbf{S_O}$, we define the 1-channel texture $\mathbf{D_O}$, that contains only the Z axis of $\mathbf{S_O}$. To do so, we concatenate the texture $\mathbf{T_d}$ with the UV map of the depth of the mesh in object space $\mathbf{D_O}$. We feed the network with a 4D tensor of $[\mathbf{T_{d_R}}, \mathbf{T_{d_G}}, \mathbf{T_{d_B}}, \mathbf{D_O}]$ and predict the resulting 3-channel albedo $\mathbf{A_D}$. Instead of $\mathbf{D_O}$, the shape normals ($\mathbf{N_O}$) can also be used. Secondly, we split the original high-resolution data into overlapping patches of $\hat{H}_p \times \hat{H}_p$ pixels, in order to augment the number of data samples and fit the data into the available GPU memory.

Therefore, in order to de-light $\hat{\mathbf{T}}$, and acquire the diffuse albedo $\mathbf{A_D}$, we train an image-to-image translation network $\mathcal{G}_{A_D} : \mathbf{T_d}, \mathbf{D_O} \mapsto \mathbf{A_D} \in \mathbb{R}^{\hat{H}_p \times \hat{H}_p \times 3}$ and then extract the diffuse albedo $\mathbf{A_D}$ by the following:

$$\mathbf{A_D} = \mathcal{G}_{A_D}(\hat{\mathbf{T}}, \mathbf{D_O}) \qquad (4)$$

4.5.2 Specular Albedo Extraction

Predicting the entire specular BRDF and the per-pixel specular roughness from the illuminated texture $\hat{\mathbf{T}}$ or the inferred diffuse albedo $\mathbf{A_D}$, poses an unnecessary challenge. As shown in [28], [31] a subject can be realistically rendered using only the intensity of the specular reflection (specular albedo) $\mathbf{A_S}$, which is consistent on a face due to the skin's refractive index. The spatial variation is correlated to facial skin structures such as skin pores, wrinkles, or hair, which are apparent in both the baked texture $\mathbf{T}$ and the diffuse albedo $\mathbf{A_D}$. Both can be used as input to the network, and we empirically found that our predicted high quality diffuse albedo $\mathbf{A_D}$ produces more accurate and consistent results. Therefore, having inferred $\mathbf{A_D}$ with $\mathcal{G}_{A_D}$, we infer the specular albedo $\mathbf{A_S}$ by a similar patch-based image-to-image translation network from the diffuse albedo ($\mathcal{G}_{A_S} : \mathbf{A_D} \mapsto \mathbf{A_S} \in \mathbb{R}^{\hat{H}_p \times \hat{H}_p \times 1}$):

$$\mathbf{A_S} = \mathcal{G}_{A_S}(\mathbf{A_D}) \qquad (5)$$

4.5.3 Specular and Diffuse Normals Extraction

The specular normals exhibit sharp surface details, such as fine wrinkles and skin pores, and are challenging to estimate, as the appearance of some high-frequency details

is dependent on the lighting conditions and viewpoint of the texture. Therefore, much detail may not be apparent in the input image or reconstructed texture. Previous works fail to predict high-frequency details [13], or rely on separating the mid- and high-frequency information in two separate maps, as a generator network may discard the high-frequency as noise [23]. Instead, we show that it is possible to employ an image-to-image translation network with feature matching loss [46] on a large high-resolution training dataset, which produces more detailed and accurate results.

Similarly to the specular albedo inference with $\mathcal{G}_{A_S}$, we feed the network with the predicted diffuse albedo $\mathbf{A_D}$. Using $\mathbf{A_D}$ instead of the 3DMM texture $\mathbf{T}$ produces more consistent results. Even though $\mathbf{T}$ contains some specular highlights, these are always concentrated on a small subset of the image, since they're reconstructed using [4]. We can also luma-transform (in sRGB) the diffuse albedo to grayscale $\mathbf{A_D^{(gray)}}$, in order to reduce the number of channels. Moreover, the consistency of the results is greatly improved when also feeding the network with the 3DMM geometry, in this case, the shape normals. Finally, we also transform the shape normals $\mathbf{N_O}$ in tangent space $\mathbf{N_T}$, where the basis is a vector pointing to $[0, 0, 1]$. We find that in this multiple-network approach, the inferred specular normals details are better accentuated, when using both the input shape normals $\mathbf{N_T}$ and the predicted specular normals $\mathbf{N_S}$ in the tangent space.

Therefore, we train another image-translation network $\mathcal{G}_{N_S} : \mathbf{A_D^{gray}}, \mathbf{N_T} \mapsto \mathbf{N_S}, \in \mathbb{R}^{\hat{H}_p \times \hat{H}_p \times 3}$ to transform the concatenation of the grayscale diffuse albedo $\mathbf{A_D^{gray}}$ and the shape normals in tangent space $\mathbf{N_T}$ to the specular normals $\mathbf{N_S}$. The specular normals are extracted by the following:

$$\mathbf{N_S} = \mathcal{G}_{N_S}(\mathbf{A_D^{gray}}, \mathbf{N_T}) \qquad (6)$$

The diffuse normals $\mathbf{N_D}$ are highly correlated with the 3DMM-reconstructed shape normals $\mathbf{N_O}$, as the evenly scattered light blurs most skin details. Similarly fpr $\mathcal{G}_{N_S}$, we train a network $\mathcal{G}_{N_D} : \mathbf{A_D^{gray}}, \mathbf{N_O} \mapsto \mathbf{N_D} \in \mathbb{R}^{\hat{H}_p \times \hat{H}_p \times 3}$ to map the concatenation of the grayscale diffuse albedo $\mathbf{A_D^{gray}}$ and the shape normals in object space $\mathbf{N_O}$ to the diffuse normals $\mathbf{N_D}$. The diffuse normals are extracted as:

$$\mathbf{N_D} = \mathcal{G}_{N_D}(\mathbf{A_D^{gray}}, \mathbf{N_O}) \qquad (7)$$

Finally, the inferred specular normals can enhance the mesoscopic structure of the reconstructed geometry $\mathbf{S}$, by refining its features and adding plausible details. Based on [73], we integrate the specular normals in the tangent space $\mathbf{N_S}$ to produce a height UV map, which describes high-resolution per-pixel surface elevation. The height map can be then be embossed on a subdivided 3DMM-reconstructed geometry $\mathbf{S}$, to produce a higher-resolution shape.

4.6 Photorealistic Differentiable Facial Rendering

Here, we formulate a photorealistic differentiable rendering methodology, that can be incorporated in our image-translation networks during training and render the training data and results in different illumination environments.

Shading can be modeled as *local illumination*, which only models the surface reflection of light sources on objects and *global illumination*, which models light propagation in a scene, including indirect illumination, and produces more realistic results at a higher computational cost. We choose to rely on local illumination shading, since most such models are differentiable and much faster to compute than global illumination. Despite producing more realistic results, rendering high-resolution human skin with global illumination takes several minutes, and would be impractical to use while training a deep neural network like ours. There exist various local illumination models appropriate for rendering human skin, on which we capitalize on to compose the following methodology for photorealistic facial rendering. We achieve fast and differentiable photorealistic rendering, by using a local illumination model and approximations for self-occlusion and subsurface scattering. Additionally, ambient occlusion is inherently baked in the captured diffuse albedo and is thus reproduced during rendering.

4.6.1 Shading Model

We use Lambertian shading for the diffuse component $\mathbf{U_D}$ and Blinn-Phong [74] shading for the specular component $\mathbf{U_S}$, given their photorealistic results for human skin and their cheap computation. For the specular exponent s, we use a common spatially varying UV map. For a reflectance and shape set $(\mathbf{A_D}, \mathbf{A_S}, \mathbf{N_D}, \mathbf{N_S}, \mathbf{S})$, a camera with view direction $\mathbf{v}$ and a set of n_l light sources with $\mathbf{l_j}$ direction and $\mathbf{c_j}$ intensity and an ambient illumination intensity $\mathbf{c_a}$, we evaluate the shading for each pixel i as follows:

$$\mathbf{U_{D_i}} = \mathbf{c_a}\mathbf{A_{D_i}} \sum_{j=1}^{n_l} (\mathbf{N_{D_i}} \cdot \mathbf{l_j})\mathbf{c_j} \qquad (8)$$

$$\mathbf{U_{S_i}} = \mathbf{A_{S_i}} \sum_{j=1}^{n_l} (\chi^+(\mathbf{N_{S_i}} \cdot \mathbf{h_j}))^s \mathbf{c_j}, \qquad \mathbf{h_j} = \frac{\mathbf{l_j} + \mathbf{v_j}}{||\mathbf{l_j} + \mathbf{v_j}||} \qquad (9)$$

where $\chi^+(x)$ a piece-wise function that returns $\max\{0, x\}$, since negative angles between the normals and light source direction do not contribute to specular reflection.

4.6.2 Rendering Directly in UV Space

Rasterization, the process of transforming the geometrical shape and texture to pixels visible by a camera, is traditionally non-differentiable, and also computationally expensive. The various methods that have been proposed for differentiable rasterization are based on sub-optimal approximations [49], [50], [51] or are very expensive [52], [53]. This motivated our pipeline to completely avoid rasterization, by rendering directly on the UV space.

The geometry shape vertices $\mathbf{S}$ are projected and interpolated in the same UV space as the reflectance textures $\mathbf{S_O}$, creating texels with a one-to-one correspondence between normals, shape and reflectance pixels. Hence, each texel's $S_{O_{u,v}}$ are used to evaluate the view direction $\mathbf{v}$ used in Eq. 9. This way (a) is faster than using rasterization, (b) is differentiable, (c) can be used with small patches of a larger texture and shape and (d) creates a pixel-to-pixel correspondence between reflectance and rendering.

4.6.3 Fast Differentiable Facial Subsurface Scattering

Subsurface scattering (SSS) describes the light that exits a translucent medium at a different point from where it had entered. Human skin, a dielectric material, exhibits

such properties and the travel distance can be further than that covered by the lambertian model (Eq.8). Subsurface scattering in the skin has a smoothing effect, with predominantly red color bleed and is required for the photorealistic rendering of skin [75]. These effects also vary across the skin and are stronger in more translucent areas such as the nose. Accurate SSS requires the expensive measurements of light transport, however we find that the following modifications to our renderer produce a photorealistic approximation that improves the results of our method.

A local illumination BRDF as described in Eq. 8 cannot model light scattered over large areas. However, the scattering occurring in human skin travels only a few millimeters and can be modeled by separately modeling the normals for diffuse reflection $\mathbf{N_D}$ [27]. We separately capture (Sec. 3.1) and infer both $\mathbf{N_D}$ and $\mathbf{N_S}$, which are then separately used to evaluate the diffuse (Eq. 8) and specular (Eq. 9) components. Both normals are wavelength dependent [31] and we acquire $\mathbf{N_D}$ from the red channel and $\mathbf{N_S}$ from the blue channel of our captures. This method accurately models the SSS angular blur (Fig. 4) and does not impose a computational overhead during rendering or training.

Nevertheless, the above does not reproduce the spectrally-dependent spatial blur produced by SSS, which results in red-dominated color bleed and shadow smoothing. These can be accurately approximated by texture-space SSS [75], which blurs the diffuse component $\mathbf{U_D}$ under multiple kernels, based on the wavelength associated with the R, G, B channels. [75] uses the weighted combinations per channel for 6 different kernels, which is too expensive for our training requirements. In line with [76], we find that a single kernel is adequate and much faster. Empirically, for a gaussian filter $g()$, we calculate the mean kernel at $k = 1.4mm$ or 21 pixels in our standard topology and define a weighted combination of $\mathbf{w_D} = diag(0.5, 0.85, 0.95)$ for the diffuse component and $\mathbf{w_{SSS}} = diag(0.5, 0.15, 0.05)$ for the subsurface scattering. We use a manually created standard translucency map $\mathbf{C}$, which describes the amount of light absorbed and scattered on different facial areas. Finally, since we are using N_D, we localize this effect only on the darker areas, by multiplying the translucency map with an inverse brightness mask. Therefore, the diffuse component $\mathbf{U_D}$ with subsurface scattering $\mathcal{S}$ is defined as:

$$\mathcal{S}(\mathbf{U_{D_i}}) = (\mathbf{1} - \mathbf{C'_i}) \circ \mathbf{w_D} \mathbf{U_{D_i}} + \mathbf{C'_i} \circ \mathbf{w_{SSS}} \, g(\mathbf{U_{D_i}}) \quad (10)$$

$$\mathbf{C'} = \mathbf{C} \circ \left(\mathbf{1} - \sum_{j=1}^{n_L} (\mathbf{N_D} \cdot \mathbf{l}_j) \mathbf{c_j} \right)$$

where $\circ$ is the Hadamard product. As shown in Fig. 4, the usage of both SSS methods provides realistic results, with the minimum computational overhead.

4.6.4 Differentiable Shadows Simulation

The rendering framework so far does not include self-occlusion shadows, whose computation entails several challenges. Pytorch3D [20] does not support self-occlusion tracing, and efficient local illumination models, such as Blinn-Phong, inherently do not model it. On the other hand, differentiable global illumination algorithms [52], [53], that compensate for self-occlusion, are too expensive while training. Finally, the patches being rendered are often unaware

of the geometry that causes self-occlusion, as it may appear on other patches (i.e. patches of nose and cheek).

(a) Pytorch3D, using $\mathbf{S}, \mathbf{A_D}, \mathbf{N}$ (b) Blinn-Phong, using $\mathbf{S}, \mathbf{A_D}, \mathbf{A_S}, \mathbf{N_S}$ (c) b) with Subsurface Scattering   (d) c) with self-occlusion AE (in & left of nose)
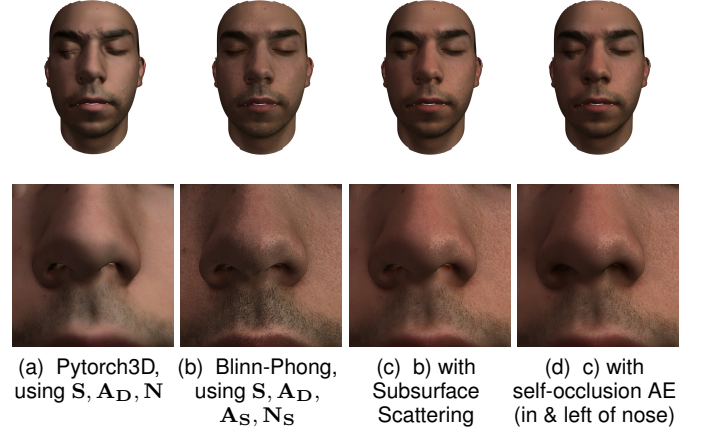
Fig. 4: The impact of our rendering modifications, to the Pytorch3D Mesh Renderer [20]. Top row: rasterized mesh with rendered texture, bottom: detail. Our improvements in realisticity, also improve the network's ability to recover the albedo and specular highlights.

As an efficient and simple solution, we propose a simple autoencoder network, that generates self-occlusion UV maps based on light source direction and intensity. We train the autoencoder on UV maps baked only with self-occlusion. We also train a linear regressor that maps light sources features $\mathbf{L}$ to the autoencoder's learned latent features $\mathbf{H_{\mathcal{O}}}$. For each training example, we create a set of n_l random light sources, with direction $\mathbf{l_j}$ and luminosity $\mathbf{c_{l_j}}$, which we stack in a matrix $\mathbf{L} = [\mathbf{l_j} \, \dots \, \mathbf{l_{N_l}} \, \mathbf{c_{l_j}} \, \dots \, \mathbf{c_{l_{n_l}}}]^\top$. We acquire self-occlusion UV-maps using a global illumination method, at the same topology of our main dataset, for the mean geometry of our dataset. We then pre-train the autoencoder and regressor on these data. On each rendering step of the main network training, the rendering environment's light source features $\mathbf{L}$ are regressed to the latent space of the autoencoder, using the learned weights $\mathbf{W_{\mathcal{O}}}$ from which the decoder $\mathcal{O}(\mathbf{LW_{\mathcal{O}}})$ generates the self-occlusion map. The result is multiplied with the rendered diffuse and specular components, to produce the final rendering:

$$\mathcal{R}(\mathbf{R}, \mathbf{L}) = \mathcal{O}(\mathbf{LW_{\mathcal{O}}}) \circ (\mathcal{S}(\mathbf{U_D}) + \mathbf{U_S}) \quad (11)$$

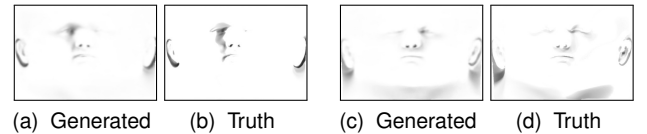(a) Generated   (b) Truth   (c) Generated   (d) Truth

Fig. 5: Prediction and ground truth for our self-occlusion autoencoder $\mathcal{O}$ (Sec. 4.6.4) for randomly sampled sets of 3 light sources, as input.

Self-occlusion on human faces (i.e. Fig. 5) does not exhibit sharp edges and is similar between different facial geometries. Therefore, we make the following simplifications: a) We use self-occlusion UV maps rendered from our dataset's mean shape, which enables the decoder $\mathcal{O}$ to correctly learn a meaningful latent representation, b) We train the autoencoder on low-resolution inputs and upsample the needed cropped patch. Thus, we have fast and differentiable
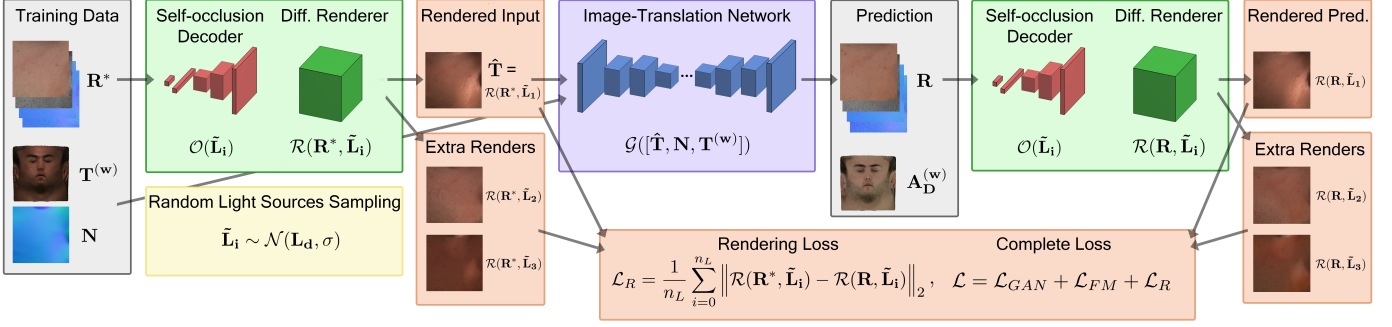
Fig. 6: AvatarMe^{++} training methodology: For each iteration we render reflectance patches $\mathbf{R}^* = [\mathbf{A_D}, \mathbf{A_S}, \mathbf{N_D}, \mathbf{N_S}]$ from the captured data (RealFaceDB), using our differentiable renderer $\mathcal{R}$ (Sec. 4.6) and occlusion autoencoder (Sec. 4.6.4). The rendering parameters $\tilde{\mathbf{L}}_\mathbf{i} \sim \mathcal{N}(\mathbf{L_d}, \sigma)$, $i = 1 \ldots n_L$ are sampled from a distribution with mean the target 3DMM environment $\mathbf{L_d}$. We pass one rendered patch $\hat{\mathbf{T}} = \mathcal{R}(\mathbf{R}, \tilde{\mathbf{L}}_\mathbf{i})$ to our main network $\mathcal{G}$ (Sec. 4.7) and produce the reflectance patches $\mathbf{R}$. The training $\mathbf{R}^*$ and generated $\mathbf{R}$ reflectance patches are used for the adversarial loss $\mathcal{L}_{GAN}$ and feature-matching loss $\mathcal{L}_{FM}$. Moreover, the consistency of the predicted patches is improved by including the down-scaled input texture $\mathbf{T^{(w)}}$ in $\mathcal{G}$'s input, and the down-scaled diffuse albedo $\mathbf{A_D^{(w)}}$ in the $\mathcal{G}$'s target. Additionally, we render the training and predicted data, with each $\tilde{\mathbf{L}}_\mathbf{i}$ and define the rendering loss $\mathcal{L}_R$, as the average loss for each random environment.

self-occlusion generation during training, with a minimal footprint, so that our main network can learn to remove it.

4.7 Reflectance Inference with AvatarMe^{++}

Here, we introduce a single rendering-aware image-translation network which jointly generates all reflectance components $(\mathbf{A_D}, \mathbf{A_S}, \mathbf{N_D}, \mathbf{N_S})$ from the upsampled 3DMM texture $\hat{\mathbf{T}}$. The motivation behind a single network is three-fold: jointly generating the above reflectance components: a) enables the introduction of a stochastic photorealistic rendering loss, which we show that improves the accuracy and generalization of the network, b) enables the network to share learned parameters between components, decreasing the size and memory requirement of the network, c) reduces the required inference time, since only a single forward pass is required. Fig. 6 shows an overview of this approach. In this manner, we introduce AvatarMe^{++} with a novel fast photorealistic differentiable rendering methodology (Sec. 4.6), an updated model architecture (Sec. 4.7.1) and a stochastic rendering loss (Sec. 4.7.2) which greatly improves the the results of AvatarMe.

4.7.1 Combined Image-to-Image Translation Model

We formulate the reflectance acquisition problem, as a domain adaptation problem and train a single image-translation network, The network $\mathcal{G}$ learns an inverse rendering function, on a UV texture $\hat{\mathbf{T}}$ with baked illumination.

An important challenge of this patch-based inference is producing consistent patches (especially of diffuse albedo $\mathbf{A_D}$) that can be seamlessly stitched together. We find that we can alleviate this issue by including the whole texture $\mathbf{T^{(w)}} = \mathbf{T}_{\downarrow_{\hat{H}_p \times \hat{H}_p}}$ to $\mathcal{G}$'s input and the diffuse albedo $\mathbf{A_D^{(w)}} = \mathbf{A_D}_{\downarrow_{\hat{H}_p \times \hat{H}_p}}$ to $\mathcal{G}$'s output, both downsampled ($\downarrow$) to the same size as the training patches. By including $\mathbf{T^{(w)}}, \mathbf{A_D^{(w)}}$ in the adversarial loss, we show that the network can learn the albedo color from the $\mathbf{T^{(w)}}$ and apply it when generating the high-resolution albedo patches.

The input to the generator $\mathcal{G}$ is formulated as the 9D tensor $\hat{\mathbf{T}}^+ = [\hat{\mathbf{T}}, \mathbf{N_O}, \hat{\mathbf{T}}^{(w)}] \in \mathcal{R}^{\hat{W} \times \hat{H} \times 9}$. The output is a 13D tensor $\mathbf{R}^+ = [\mathbf{A_D}, \mathbf{A_S}, \mathbf{N_D}, \mathbf{N_S}, \mathbf{A_D^{(w)}}] \in \mathcal{R}^{\hat{W} \times \hat{H} \times 13}$.

Due to the resolution of our textures, we train the network on randomized $\hat{H}_p \times \hat{H}_p$ patches of $\hat{\mathbf{T}}^+$ and $\mathbf{R}$. $\mathbf{A_D^{(w)}}$ is only used for the adversarial loss and ignored at testing. Therefore, the reflectance $\mathbf{R}$ is extracted by the following:

$$\mathbf{R} = [\mathbf{A_D}, \mathbf{A_S}, \mathbf{N_D}, \mathbf{N_S}] = \mathcal{G}([\hat{\mathbf{T}}, \mathbf{N}, \hat{\mathbf{T}}^{(w)}]) \quad (12)$$

4.7.2 Stochastic Rendering Loss

So far, we have described a single-network $\mathcal{G}$ that generates the facial reflectance $\mathbf{R} = [\mathbf{A_D}, \mathbf{A_S}, \mathbf{N_D}, \mathbf{N_S}]$ (Sec. 4.7.1) and a fast photorealistic differentiable rendering method (Sec. 4.6). Therefore, we can now introduce a rendering loss in the training of $\mathcal{G}$, where the input texture $\mathbf{T_d}$ is compared with the predicted reflectance, rendered in the domain $\mathbf{L_d}$ of the input texture. The photorealistic rendering loss is defined as $\mathcal{L}_R = \|\mathbf{T_d} - \mathcal{R}(\mathbf{R}, \mathbf{L_d})\|_2$, using Eq. 11. Please note that $\mathcal{R}$ also uses the shape $\mathbf{S_O}$ which remains static during training and we omit it to avoid cluttering. A similar approach has been proposed for planar surfaces under flash illumination in [55], [57]. To the best of our knowledge, this is the first attempt on facial BRDF and the first included in a GAN-based image-translation network. Other facial acquisition methods (i.e. [4], [14], [19], [60], [77]) that use a differentiable rendering loss, merely re-project the inferred texture or use the unrealistic Lambertian model.

The introduction of the above rendering loss leaves the network unaware of specular features across the whole face, while it motivates the network to include shading elements in the diffuse albedo, which could be accurately reproduced in the rendering. This is mainly because all the training data are rendered in the same domain of the 3DMM textures used (Sec. 4.3) and the network can always expect shadows and highlights at the same places and intensities. Therefore we take 2 steps to introduce stochasticity to our training data, similar to [55], which improves our network's accuracy and generalization outside the target 3DMM domain.

Firstly, we sample random variations of the estimated target environment parameters $\tilde{\mathbf{L}} \sim \mathcal{N}(\mathbf{L_d}, \sigma)$ in each training iteration. We use it to render both the input to the network $\tilde{\mathbf{T}} = \mathcal{R}(\mathbf{R}^*, \tilde{\mathbf{L}})$ and the predicted reflectance $\mathcal{R}(\mathbf{R}, \tilde{\mathbf{L}})$ for the rendering loss. Secondly, for each training

iteration, we sample additional n_L environment parameters, $\tilde{\mathbf{L}}_\mathbf{i} \sim \mathcal{N}(\mathbf{L_d}, \sigma), i = 1 \dots n_L$ which are not fed to the network, but are only used to compute an average rendering loss for these environments. In this manner, we stochastically approximate all light source directions within the allowed variation, while also penalizing the network for over-fitting the target 3DMM environment. Then, for the ground truth $\mathbf{R}^*$ and the inferred reflectance $\mathbf{R}$, for each training iteration, the rendering loss $\mathcal{L}_R$ is defined as:

$$\mathcal{L}_R = \frac{1}{n_L} \sum_{i=0}^{n_L} \left\| \mathcal{R}(\mathbf{R}^*, \tilde{\mathbf{L}}_\mathbf{i}) - \mathcal{R}(\mathbf{R}, \tilde{\mathbf{L}}_\mathbf{i}) \right\|_2, \tilde{\mathbf{L}}_\mathbf{i} \sim \mathcal{N}(\mathbf{L_d}, \sigma) \tag{13}$$

Overall, the objective of our image-translation network, which is based on pix2pixHD [46] is defined as:

$$\min_G \left(\max_D \mathcal{L}_{GAN}(G, D_k) + \lambda_{FM}\mathcal{L}_{FM}(G, D_k) + \lambda_R\mathcal{L}_R \right) \tag{14}$$

where $\mathcal{L}_{GAN}(G, D_k)$ is the sum of adversarial loss and $\mathcal{L}_{FM}(G, D_k)$ is the feature matching loss, for all 3 discriminators of pix2pixHD [46]. The feature matching and rendering loss terms are controlled by λ_{FM} and λ_R.

5 EXPERIMENTS

5.1 Implementation Details

The task of disentagling the diffuse and specular components, from a given input image with baked illumination can be formulated as an image-to-image translation problem. Nevertheless, as discussed previously: (a) our captured data are of very high-resolution (more than 4K) and thus cannot be used for training "as-is', due to hardware limitations (note not even on a 32GB GPU we can fit such high-resolution data in their original format), (b) pix2pixHD [46] takes into account only the texture and optionally labels, and thus geometric details, in the form of the shape and shape normals cannot be exploited to improve the quality of the generated diffuse and specular components.

5.1.1 Patch-Based Image-to-Image Translation

To alleviate the aforementioned shortcomings, we split the original high-resolution data into smaller patches of $\hat{H}_p \times \hat{H}_p$ size. More specifically, using a stride of size 256, we derive the partially overlapping patches by passing through each original UV horizontally as well as vertically. For each translation task we utilize the shape or shape normals, projected and interpolated in the same UV parameterisation as the textures. This increases the accuracy and level of detail in the derived outputs as the geometry act as a "guide" to the network. Finally, we downsample the whole input texture to the patch size and include it as well, as we find it greatly improves the consistency of the predicted patches. For the AvatarMe^{++} pipeline, we concatenate them channelwise with the texture input and thus feed to the network a 9D tensor comprising of $\hat{\mathbf{T}}, \mathbf{N_O}, \hat{\mathbf{T}}^{(w)}$ and generate a 13D tensor comprising of $\mathbf{A_D}, \mathbf{A_S}, \mathbf{N_D}, \mathbf{N_S}$ (Eq. 12) and $\hat{\mathbf{T}}^{(w)}$, which is discarded. During inference, that patch size can be larger (e.g. $1k \times 1k$), since the network is fully-convolutional.

5.1.2 Training Setup

To train RCAN [72], we use the default hyper-parameters. For the rest of the translation of models, we use a custom translation network as described earlier, which is based on pix2pixHD [46]. More specifically, we use 9 and 3 residual blocks in the global and local generators, respectively. The learning rate we used is 0.0002, whereas the Adam betas are 0.5 for β_1 and 0.999 for β_2. In our best model, we use a feature matching loss controller of $\lambda_{FM} = 10.0$ and rendering loss controller of $\lambda_R = 0.3$, for which perform an ablation study in the following section. Finally, we use a variable number of input and outputs as $N-$dimensional tensors, based on the implementation (Sec. 4.5.1, 4.5.2, or 4.7). As mentioned earlier, this substantially improves the results by accentuating the details and enforcing patch consistency.

5.1.3 Rendering Setup

To implement the facial photorealistic differentiable rendering (Sec. 4.6), we extend the recently published PyTorch3D [20] (version 0.3.0), for its speed, easily modifiable modular design and its compatibility with our image-translation network. Specifically, we fully integrate it with the generator and discriminator networks of our image-translation network, implement objects for multiple reflectance textures and implement a texture-space shader, that uses our framework from Sec. 4.6. For the shader's parameters, i.e. shininess exponent and translucency masks, we use a single manually created UV map with spatially varying values.

For the self-occlusion prediction, we train an autoencoder and a linear regressor that maps light source features to the autoencoder's latent values. The encoder and decoder consist of 5 convolutional blocks, each block having 2 convolutional layers with ELU [78], batch normalization [79] and a down- or up-sampling layer. The hidden layer has 64 features and connects to the encoder and decoder with a fully connected layer of 256 features.

5.2 Evaluation

To evaluate our reconstruction pipeline, we compare reconstructed relfectance maps and renderings acquired with AvatarMe^{++}, with ground truth data captured in a similar manner as our dataset RealFaceDB, the digital Emily Project [80] and current state-of-the-art. We use the Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) [81]. In Table 1, we conduct quantitative comparisons against the state-of-the-art [23], [51]. As can be seen, our method outperforms [13] and [23] by a significant margin. All meshes were manually registered to the same topology and UV parameterization. Moreover using a state-of-the-art face recognition algorithm [16], we also find the highest match of facial identity compared to the input images when using our method. The input images were compared against renderings of the faces with reconstructed geometry and reflectance, including eyes added manually to [23]. We also present qualitative comparisons in Fig. 7 and Fig. 8.

For the qualitative comparisons, we perform 3D reconstructions of arbitrary images. As shown in Figs. 7 and 8, our method does not produce any artifacts in the final renderings and successfully handles extreme poses and occlusions
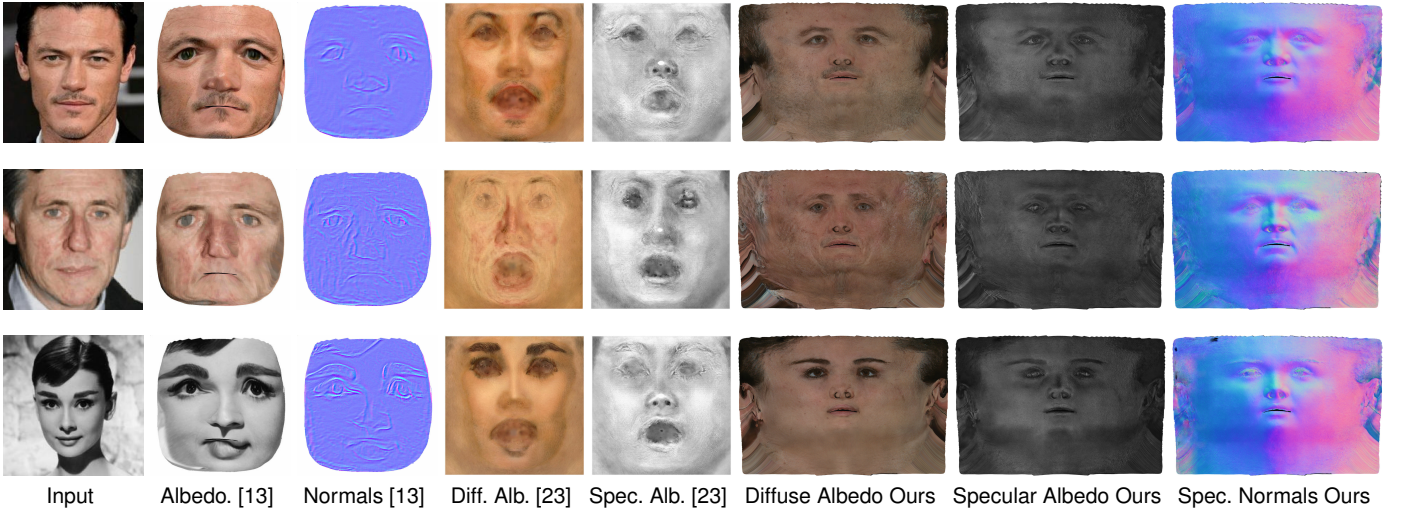
| Input | Albedo. [13] | Normals [13] | Diff. Alb. [23] | Spec. Alb. [23] | Diffuse Albedo Ours | Specular Albedo Ours | Spec. Normals Ours |

Fig. 7: Reflectance maps produced by our method AvatarMe++, against state-of-the-art methods. Reconstructions of [23] are provided by the authors and [13] are acquired using their open-sourced models.

Fig. 8: Qualitative comparison of rendered reconstructions from a frontal and challenging side image. [23] results are provided by the authors and [13] results are acquired from their open-sourced models.

TABLE 1: Quantitative comparisons with state-of-the-art, between 6 reconstructions of the same subject, from different "in-the-wild" images, and ground truth using [32]. We transform [13], [23] results to our UV topology and compute only for a $2K \times 2K$ centered crop, as they only produce the frontal part of the face and manually add eyes to [23].

Algorithm	[23]	[13]	AvatarMe	AvatarMe++
PSNR (Albedo)	11.225	14.374	24.05	**26.18**
PSNR (Normals)	21.889	17.321	26.97	**27.12**
MSE (Albedo)	0.0225	0.0140	0.0049	**0.0038**
MSE (Normals)	0.0047	0.0049	0.0031	**0.0025**
Rendered ID Score [16]	0.629	0.632	0.873	**0.881**

such as sunglasses. We infer the texture maps in a patch-based manner from high-resolution input, which produces higher-quality details than [13] and [23], who train on high-quality scans but infer the maps for the whole face, in lower resolution. This is also apparent in Fig. 9 and Fig. 16, which shows our reconstruction after each step of our process. Moreover, we can successfully acquire each component from black-and-white images (Fig. 7) and even paintings (Fig. 12).

Furthermore, we experiment with different environment conditions, in the input images and while rendering. As pre-
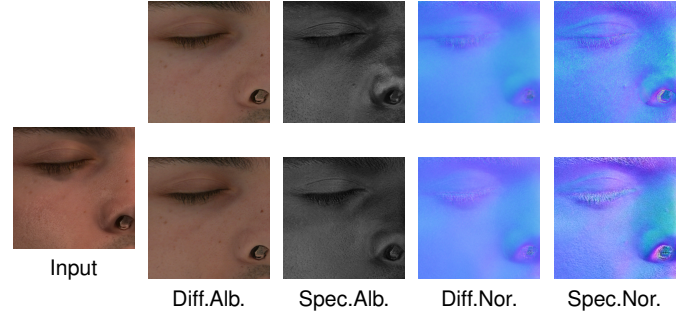

Fig. 9: Comparison of AvatarMe++ predicted reflectance with ground truth. Left: Patch of rendered test subject in target domain, Top row: predicted reflectance with our method, Bottom row: ground truth.

sented in Fig. 10, the extracted normals, diffuse and specular albedos are consistent, regardless of the illumination on the original input images. Moreover, Fig. 12 shows different subjects rendered under different environments. We can realistically illuminate each subject in each scene and accurately reconstruct the environment reflectance, including detailed specular reflections and subsurface scattering.
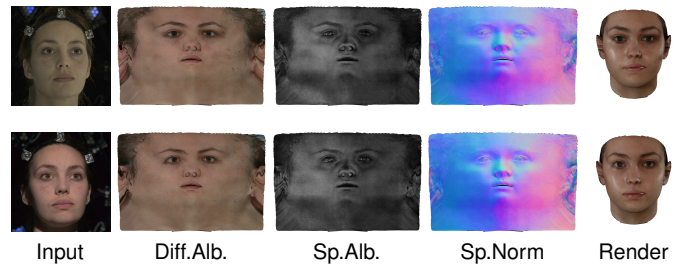

| Input | Diff.Alb. | Sp.Alb. | Sp.Norm | Render |

Fig. 10: Consistency of AvatarMe++ on varying conditions, from the Digital Emily Project [80]. We calculate on average 30.94 PSNR and 0.0007 MSE between our results. Compared to the ground truth from [80], we achieve on average 0.0083 MSE and 20.13 PSNR on albedo and and 0.011 MSE and 24.02 PSNR on normals.
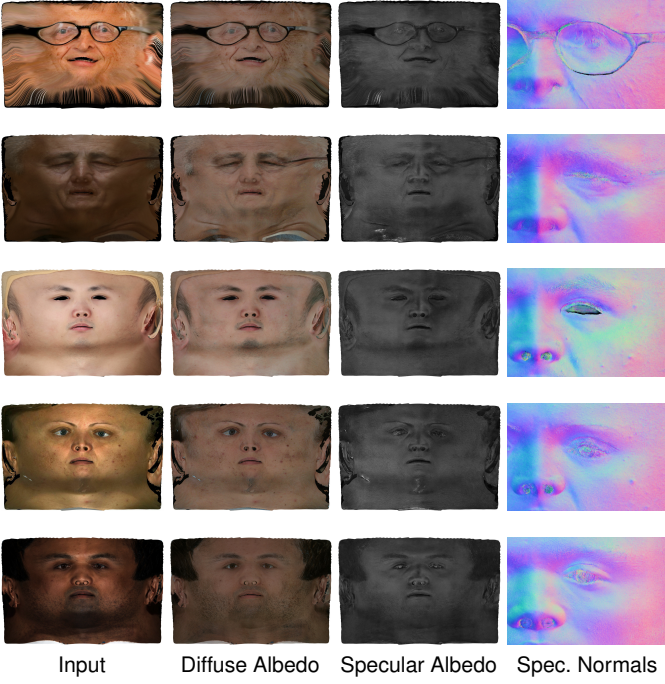
Input   Diffuse Albedo   Specular Albedo   Spec. Normals

Fig. 11: Generalization of AvatarMe++: Training AvatarMe++ with stochastically varied rendering scene parameters and our rendering loss (Sec. 4.7.2), makes the network domain-agnostic to an extend. Here we show results of a single AvatarMe++ network, on reconstructions and captured data, registered to our topology. From top to bottom: a) Reconstructed subject, with Facial Details Synthesis proxy and texture [13], b) Reconstructed subject with OSTeC fitting and texture completion [70], c) Captured subject from FaceScape [82], d) Captured subject from Superface [83], e) Captured subject with 3dMDface (3dmd.com).

TABLE 2: Method components ablation. Training time is given per training iteration, testing time is given for the inference of whole $\hat{W} \times \hat{H}$ textures. AvatarMe time includes the training time for all 4 of its networks. We measure the mean squared error (MSE) between ground truth and inferred reflectance maps, for the test set $\mathcal{T}$ (Sec 5.3.1).

Method	Train time	Test time	MSE
AvatarMe (Sec. 4.5)	22.8	15.98	0.0079
Single Network (Sec. 4.7.1)	8.0	6.35	0.0080
+ Rendering loss (Sec. 4.7.1)	14.4	6.48	0.0075
+ Whole low-res texture (Sec. 4.7.1)	14.8	9.31	0.0066
+ Random environment (Sec. 4.7.2)	14.8	9.35	0.0055
+ Multiple random env. (Sec. 4.7.2)	24.0	9.35	0.0048
+ Occl. AE (Sec. 4.6.4) (AvatarMe++)	28.8	9.35	0.0043

5.3 Ablation

5.3.1 Method Components and Variants

We investigate the importance of the various components of our method, when added on the base network. We create a test set $\mathcal{T} = \{\mathcal{R}(\mathbf{R}_i, \mathbf{L}_j)\}$, by rendering 5 test subjects from our captured dataset, RealFaceDB, with reflectance $\mathbf{R}_j$, in 10 total illumination environments $\mathbf{L}_j$ including: a) the 3DMM-target environment (Sec. 4.3), b) $-/+30\%$ light source variation in position, c) $-/+30\%$ in intensity from a), and d) directional front and side illumination. We compare the mean squared error, between the ground truth and the model's predicted reflectance and the rendering of the prediction in the source testing environment, as well as in the target environment. Testing in various illumination environments evaluates the generalization abilities of our networks,
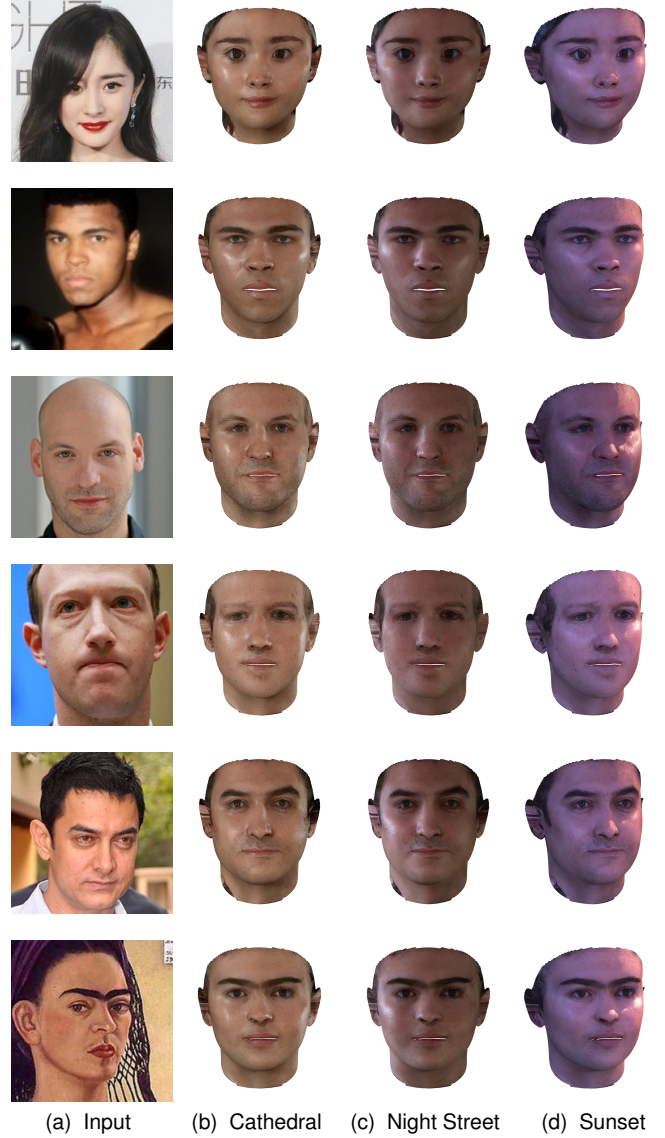

(a) Input   (b) Cathedral   (c) Night Street   (d) Sunset

Fig. 12: AvatarMe++ results rendered in different environments.

outside the target environment. We plot both inference and re-rendering error in Fig. 13 and show the reconstructed diffuse albedo $\mathbf{A_D}$ in Fig. 14. Moreover, although using a single network reduces the training and inference time by about 75%, the additional rendering adds a significant overhead in training time, but does not significantly increase testing time. We show this trade-off between reconstruction quality and training and testing time Table 2.

The introduction of the stochastic rendering loss (Sec. 4.7.2) enables AvatarMe++ to generalize to facial textures with different environments, while AvatarMe is trained only on the target 3DMM environment. Fig. 13 shows the quantitative improvement of AvatarMe++ on the test set $\mathcal{T}$ with 10 different environments, two of which are shown in Fig. 14. Finally, Fig. 11 shows the network's ability to generalize to textures obtained from different datasets (i.e. [82], [83]) and acquisitions methods (i.e. [46], [70]).

5.3.2 Network and Rendering Hyper-parameters

Training the network on 4 Tesla V100 GPUs takes about one day for a base single network (Sec. 4 or Sec. 4.7.1) and
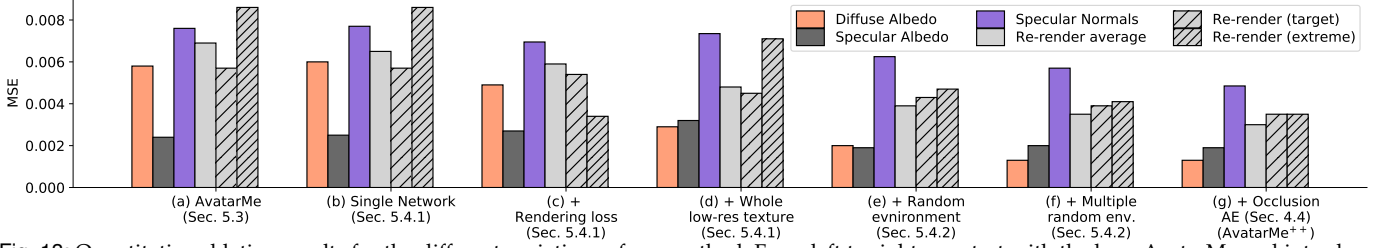
Fig. 13: Quantitative ablation results for the different variations of our method. From left to right, we start with the base AvatarMe and introduce each component of AvatarMe^{++}. Each variation is applied on test set $\mathcal{T}$ (Sec. 5.3.1). We measure the mean squared error (MSE) between ground truth and prediction's relfectance, and re-rendering in the target environment and an extreme side illumination environment used in $\mathcal{T}$.
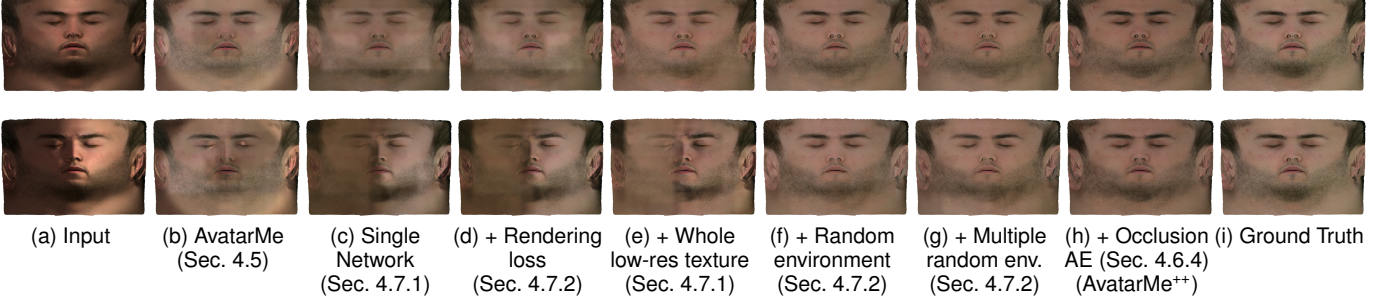

Fig. 14: From left to right, rendered test image, predicted diffuse albedo $\mathbf{A_D}$ from each of the main method variations and the ground truth diffuse albedo $\mathbf{A_D^*}$. Top row input is rendered in our target environment, bottom row in an extreme side illumination environment from $\mathcal{T}$.

Fig. 15: Predicted diffuse normals $\mathbf{N_D}$, specular albedo $\mathbf{A_D}$, and specular normals details $\mathbf{N_S}$ in tangent space, between AvatarMe (Sec.4.5) and AvatarMe^{++}. Diffuse albedo $\mathbf{A_D}$ comparison included in Fig. 14.

Fig. 17: The effect of different values of the rendering loss controller λ_R (Eq. 14) in the average reflectance $\mathbf{R}$ reconstruction for 5 subjects rendered in 10 different environments and the average re-rendering mean squared truth with the ground-truth.
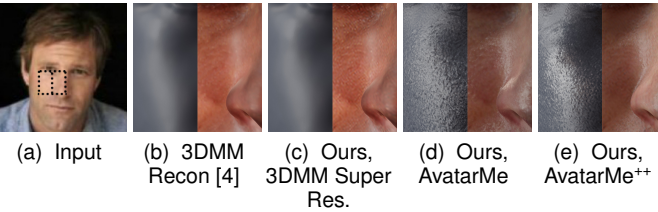

Fig. 16: Comparison of our shape reconstruction (left) and rendering (right) between the 3DMM fitting with GANFIT [4], upsampled texture (Sec. 4.4), AvatarMe (Sec. 4.5) and AvatarMe^{++} (Sec. 4.7).

up to three days for the complete method, with multiple rendering loss computations and self-occlusion in rendering (Sec. 4.7.2). Given these restrictions, we perform a study on the effect of the important hyper parameters. We train the complete method with different seeds and record an average 3.14% standard deviation in the error of reflectance reconstruction and 5.37% standard deviation in the error of re-rendered textures. All the other results are reported using the same seed for all stochastic operations. The impact of the rendering loss controller λ_R (Eq. 14) is shown in Fig. 17. We
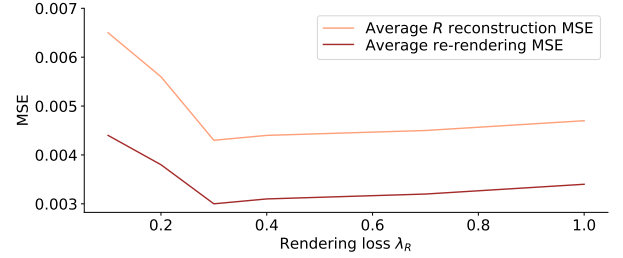
find $\lambda_R = 0.3$ to yield the best results.

Table 3 shows the effect of the scene variation hyper-parameters when training with randomized light source $\mathbf{l}$, camera direction $\mathbf{v}$ and light source intensity $\mathbf{c_l}$. We find the best results at up to 50% variation in light source position, and up to 25% in camera position and light source intensity. Finally, Table 3 shows the effect of the number of random scenes n_L used to evaluate the rendering loss (Eq. 13). We find a significant improvement in using 2 scenes instead of 1, and a smaller improvement when adding additional scenes. However each additional rendering loss evaluation introduces a trade-off with computational time.

5.4 Limitations

While our dataset contains a relatively large number of subjects, it does not contain sufficient examples of subjects from certain ethnicities (Sec. 3.2). Despite our effective training data augmentation, we find that the predicted diffuse albedo for subjects of the lightest or darkest skin types, produces patches of inconsistent intensity and the specular normals of older subjects are reconstructed sub-optimally.

TABLE 3: Ablation study for rendering loss hyper-parameters. Comparison of reflectance prediction and re-rendering error for variation in scene parameters (light source position $\mathbf{l}$, camera view direction $\mathbf{v}$ light source intensity $\mathbf{c_l}$) around the target environment (top table), and number of random scenes evaluated n_L in the rendering loss (bottom table) (Sec. 4.7.2). Average error reported for test set $\mathcal{T}$ (Sec. 5.3.1).

Scene Variation			Results	
$\mathbf{l}$	$\mathbf{v}$	$\mathbf{c_l}$	Recon. MSE	Render MSE
0%	0%	0%	0.0055	0.0059
50%	25%	25%	**0.0043**	**0.0030**
75%	50%	50%	0.0051	0.0036

Evaluations n_L	Train Time	Recon. MSE	Render MSE
1	**15.6**	0.0054	0.0037
2	22.4	0.0045	0.0032
3	28.8	**0.0043**	**0.0030**

Moreover, the accuracy of facial reconstruction is not completely independent of the quality of the input photograph, and well-lit, higher resolution photographs produce more accurate results, depending on the 3DMM method used. Additionally, we show that our method can generalize to various reconstruction and capturing methods, however, the model expects their light sources to be in front of the subject. Finally, our renderer models self-occlusion but not occlusion from foreign objects. This could be modeled by augmenting our dataset with randomized occlusions.

6 CONCLUSION

In this paper, we propose the first methodology that produces high-quality rendering-ready face reconstructions from arbitrary "in-the-wild" images. We build upon recently proposed 3D face reconstruction techniques and train an image translation network that can perform estimation of high quality (a) diffuse and specular albedo, and (b) diffuse and specular normals. This is made possible with a large training dataset of 200 faces acquired with high quality facial capture techniques and a fast photorealistic differentiable rendering framework. We demonstrate that it is possible to produce rendering-ready faces from arbitrary face images varying in pose, occlusions, etc., including black-and-white and drawn portraits. Our results exhibit unprecedented level of detail and realism in the reconstructions, while preserving the identity of subjects in the input photographs.

ACKNOWLEDGMENTS

REFERENCES

[1] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[2] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*. ACM Press, 1999, pp. 187–194.

[3] Y. Zhou, J. Deng, I. Kotsia, and S. Zafeiriou, "Dense 3D Face Decoding Over 2500FPS: Joint Texture & Shape Convolutional Mesh Decoders," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1097–1106.

[4] B. Gecer, S. Ploumpis, I. Kotsia, and S. Zafeiriou, "GANFIT: Generative Adversarial Network Fitting for High Fidelity 3D Face Reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1155–1164.

[5] M. Sela, E. Richardson, and R. Kimmel, "Unrestricted Facial Geometry Reconstruction Using Image-To-Image Translation," in *IEEE International Conference on Computer Vision*, 2017.

[6] A. Tewari, M. Zollhöfer, P. Garrido, F. Bernard, H. Kim, P. Pérez, and C. Theobalt, "Self-Supervised Multi-Level Face Model Learning for Monocular Reconstruction at Over 250 Hz," in *Conference on Computer Vision and Pattern Recognition*, 2018.

[7] L. Tran and X. Liu, "On Learning 3D Face Morphable Model from In-the-wild Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[9] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniah, and D. Dunaway, "A 3D Morphable Model Learnt From 10,000 Faces," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[10] W. A. P. Smith, A. Seck, H. Dee, B. Tiddeman, J. B. Tenenbaum, and B. Egger, "A Morphable Face Albedo Model," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.

[11] B. Gecer, A. Lattas, S. Ploumpis, J. Deng, A. Papaioannou, S. Moschoglou, and S. Zafeiriou, "Synthesizing Coupled 3D Face Modalities by Trunk-Branch Generative Adversarial Networks," in *Computer Vision – ECCV 2020*, ser. Lecture noteComments in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Springer International Publishing, 2020.

[12] R. Li, K. Bladin, Y. Zhao, C. Chinara, O. Ingraham, P. Xiang, X. Ren, P. Prasad, B. Kishore, J. Xing, and H. Li, "Learning Formation of Physically-Based Face Attributes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[13] A. Chen, Z. Chen, G. Zhang, K. Mitchell, and J. Yu, "Photorealistic facial details synthesis from single image," in *IEEE/CVF International Conference on Computer Vision*, October 2019.

[14] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras, "Neural Face Editing With Intrinsic Image Disentangling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5541–5550.

[15] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *International Conference on Learning Representations*, 2018.

[16] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *Conference on Computer Vision and Pattern Recognition*, 2019.

[17] A. Tewari, F. Bernard, P. Garrido, G. Bharaj, M. Elgharib, H.-P. Seidel, P. Perez, M. Zollhofer, and C. Theobalt, "FML: Face Model Learning From Videos," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2019.

[18] W. Zhu, H. Wu, Z. Chen, N. Vesdapunt, and B. Wang, "Reda:reinforced differentiable attribute for 3d face reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[19] K. Genova, F. Cole, A. Maschinot, A. Sarna, D. Vlasic, and W. T. Freeman, "Unsupervised Training for 3D Morphable Model Regression," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2018, pp. 8377–8386.
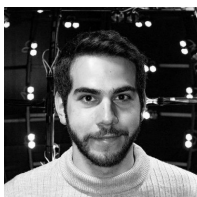
[20] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, Georgia Gkioxari, and Nikhila Ravi; Jeremy Reizenstein; David Novotny; Taylor Gordon; Wan-Yen Lo; Justin Johnson; Georgia Gkioxari, "PyTorch3D," 2020.

[21] L. Huynh, W. Chen, S. Saito, J. Xing, K. Nagano, A. Jones, P. Debevec, and H. Li, "Mesoscopic Facial Geometry Inference Using Deep Neural Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2018, pp. 8407–8416.

[22] S. Saito, L. Wei, L. Hu, K. Nagano, and H. Li, "Photorealistic Facial Texture Inference Using Deep Neural Networks," in *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5144–5153.

[23] S. Yamaguchi, S. Saito, K. Nagano, Y. Zhao, W. Chen, K. Olszewski, S. Morishima, and H. Li, "High-fidelity facial reflectance and geometry inference from an unconstrained image," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 162:1–162:14, Jul. 2018.

[24] A. Lattas, S. Moschoglou, B. Gecer, S. Ploumpis, V. Triantafyllou, A. Ghosh, and S. Zafeiriou, "AvatarMe: Realistically Renderable 3D Facial Reconstruction "In-the-Wild"," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 760–769.

[25] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar, "Acquiring the reflectance field of a human face," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH*. ACM Press, 2000, pp. 145–156.

[26] T. Weyrich, W. Matusik, H. Pfister, B. Bickel, C. Donner, C. Tu, J. McAndless, J. Lee, A. Ngan, H. W. Jensen, and M. Gross, "Analysis of human faces using a measurement-based skin reflectance model," *ACM Transactions on Graphics*, vol. 25, no. 3, Jul. 2006.

[27] W.-C. Ma, T. Hawkins, P. Peers, C.-F. Chabert, M. Weiss, and P. Debevec, "Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination," in *Proceedings of the 18th Eurographics conference on Rendering Techniques*, ser. EGSR'07. Eurographics Association, Jun. 2007, pp. 183–194.

[28] A. Ghosh, G. Fyffe, B. Tunwattanapong, J. Busch, X. Yu, and P. Debevec, "Multiview face capture using polarized spherical gradient illumination," *ACM Transactions on Graphics*, Dec. 2011.

[29] T. Beeler, B. Bickel, G. Noris, P. Beardsley, S. Marschner, R. W. Sumner, and M. Gross, "Coupled 3D reconstruction of sparse facial hair and skin," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 117:1–117:10, Jul. 2012.

[30] P. Gotardo, J. Riviere, D. Bradley, A. Ghosh, and T. Beeler, "Practical dynamic facial appearance modeling and acquisition," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 232:1–232:13, Dec. 2018.

[31] C. Kampouris, S. Zafeiriou, and A. Ghosh, "Diffuse-Specular Separation using Binary Spherical Gradient Illumination," *Eurographics Symposium on Rendering*, p. 10, 2018.

[32] A. Lattas, M. Wang, S. Zafeiriou, and A. Ghosh, "Multi-view facial capture using binary spherical gradient illumination," in *ACM SIGGRAPH 2019 Posters*. ACM, Jul. 2019.

[33] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3d face model for pose and illumination invariant face recognition," *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS) for Security, Safety and Monitoring in Smart Environments*, 2009.

[34] B. Egger, W. A. P. Smith, A. Tewari, S. Wuhrer, M. Zollhoefer, T. Beeler, F. Bernard, T. Bolkart, A. Kortylewski, S. Romdhani, C. Theobalt, V. Blanz, and T. Vetter, "3D Morphable Face Models—Past, Present, and Future," *ACM Transactions on Graphics*, vol. 39, no. 5, pp. 1–38, Jun. 2020.

[35] A. Tuan Tran, T. Hassner, I. Masi, and G. Medioni, "Regressing Robust and Discriminative 3D Morphable Models With a Very Deep Neural Network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5163–5172.

[36] F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, and W. T. Freeman, "Synthesizing Normalized Faces from Facial Identity Features," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jul. 2017, pp. 3386–3395.

[37] Y. Guo, J. Zhang, J. Cai, B. Jiang, and J. Zheng, "CNN-Based Real-Time Dense Face Reconstruction with Inverse-Rendered Photo-Realistic Face Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 6, pp. 1294–1307, Jun. 2019.

[38] E. Richardson, M. Sela, and R. Kimmel, "3D face reconstruction by learning from synthetic data," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 460–469.

[39] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, pp. 194:1–194:17, 2017.

[40] S. Ploumpis, H. Wang, N. Pears, W. A. P. Smith, and S. Zafeiriou, "Combining 3D morphable models: A large scale face-and-head model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[41] S. Ploumpis, E. Ververas, E. O. Sullivan, S. Moschoglou, H. Wang, N. Pears, W. Smith, B. Gecer, and S. P. Zafeiriou, "Towards a complete 3D morphable model of the human head," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[42] H. Luo, K. Nagano, H.-W. Kung, Q. Xu, Z. Wang, L. Wei, L. Hu, and H. Li, "Normalized avatar synthesis using stylegan and perceptual refinement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 662–11 672.

[43] 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 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[44] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017.

[45] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2242–2251.

[46] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[47] G. Patow and X. Pueyo, "A Survey of Inverse Rendering Problems," *Computer Graphics Forum*, vol. 22, no. 4, 2003.

[48] M. M. Loper and M. J. Black, "OpenDR: An Approximate Differentiable Renderer," in *Computer Vision – ECCV 2014*, ser. Lecture noteComments in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014.

[49] H. Kato, Y. Ushiku, and T. Harada, "Neural 3D Mesh Renderer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3907–3916.

[50] S. Liu, T. Li, W. Chen, and H. Li, "Soft Rasterizer: A Differentiable Renderer for Image-Based 3D Reasoning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[51] W. Chen, H. Ling, J. Gao, E. Smith, J. Lehtinen, A. Jacobson, and S. Fidler, "Learning to Predict 3D Objects with an Interpolation-based Differentiable Renderer," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 9609–9619.

[52] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen, "Differentiable Monte Carlo ray tracing through edge sampling," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 222:1–222:11, Dec. 2018.

[53] G. Loubet, N. Holzschuch, and W. Jakob, "Reparameterizing discontinuous integrands for differentiable rendering," *ACM Transactions on Graphics*, vol. 38, no. 6, pp. 228:1–228:14, Nov. 2019.

[54] K. M. Jatavallabhula, E. Smith, J.-F. Lafleche, C. F. Tsang, A. Rozantsev, W. Chen, T. Xiang, R. Lebaredian, and S. Fidler, "Kaolin: A PyTorch Library for Accelerating 3D Deep Learning Research," *arXiv:1911.05063 [cs]*, Nov. 2019.

[55] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau, "Single-image SVBRDF capture with a rendering-aware deep network," *ACM Transactions on Graphics*, Jul. 2018.

[56] X. Li, Y. Dong, P. Peers, and X. Tong, "Modeling surface appearance from a single photograph using self-augmented convolutional neural networks," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 45:1–45:11, Jul. 2017.

[57] Z. Li, K. Sunkavalli, and M. Chandraker, "Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image," in *Proceedings of the European Conference on Computer Vision*, 2018.

[58] L. P. Asselin, D. Laurendeau, and J. F. Lalonde, "Deep svbrdf estimation on real materials," in *2020 International Conference on 3D Vision (3DV)*, 2020, pp. 1157–1166.

[59] Y. Yu and W. A. Smith, "Inverserendernet: Learning single image inverse rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3155–3164.

[60] S. Sengupta, A. Kanazawa, C. D. Castillo, and D. W. Jacobs, "SfSNet: Learning Shape, Reflectance and Illuminance of Faces 'in the Wild'," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6296–6305.

[61] A. Dib, G. Bharaj, J. Ahn, C. Thébault, P. Gosselin, M. Romeo, and L. Chevallier, "Practical face reconstruction via differentiable ray tracing," in *Computer Graphics Forum*, vol. 40, no. 2. Wiley Online Library, 2021, pp. 153–164.

[62] M. B. R, A. Tewari, T.-H. Oh, T. Weyrich, B. Bickel, H.-P. Seidel, H. Pfister, W. Matusik, M. Elgharib, and C. Theobalt, "Monocular reconstruction of neural face reflectance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 4791–4800.

[63] S. Bi, S. Lombardi, S. Saito, T. Simon, S.-E. Wei, K. Mcphail, R. Ramamoorthi, Y. Sheikh, and J. Saragih, "Deep relightable appearance models for animatable faces," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–15, 2021.

[64] C. Kampouris and A. Ghosh, "ICL multispectral light stage: building a versatile LED sphere with off-the-shelf components," in *Proceedings of the Eurographics 2018 Workshop on Material Appearance Modeling*, ser. EG MAM '18. Eurographics Association, Jul. 2018.

[65] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[66] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[67] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A Semi-automatic Methodology for Facial Landmark Annotation," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, Jun. 2013, pp. 896–903.

[68] B. Amberg, S. Romdhani, and T. Vetter, "Optimal Step Nonrigid ICP Algorithms for Surface Registration," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2007, pp. 1–8.

[69] C. Donner and H. W. Jensen, "A spectral bssrdf for shading human skin." *Rendering techniques*, vol. 2006, pp. 409–418, 2006.

[70] B. Gecer, J. Deng, and S. Zafeiriou, "Ostec: One-shot texture completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7628–7638.

[71] K. Nishino and S. K. Nayar, "Eyes for relighting," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 704–711, Aug. 2004.

[72] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image Super-Resolution Using Very Deep Residual Channel Attention Networks," 2018, pp. 286–301.

[73] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi, "Efficiently combining positions and normals for precise 3D geometry," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 536–543, Jul. 2005.

[74] J. F. Blinn and M. E. Newell, "Texture and reflection in computer generated images," *Communications of the ACM*, vol. 19, no. 10, pp. 542–547, Oct. 1976.

[75] G. Borshukov and J. P. Lewis, "Realistic human face rendering for "the matrix reloaded"," in *ACM SIGGRAPH 2005 Courses*. Association for Computing Machinery, July 2005.

[76] J. Hable, "Uncharted 2:, character lighting and shading," in *SIGGRAPH Advances in Real-Time Rendering in Games course*, July 2010.

[77] A. Tewari, M. Zollhofer, H. Kim, P. Garrido, F. Bernard, P. Perez, and C. Theobalt, "MoFA: Model-Based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction," 2017.

[78] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *4th International Conference on Learning Representations, ICLR*, 2016.

[79] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[80] O. Alexander, M. Rogers, W. Lambeth, J.-Y. Chiang, W.-C. Ma, C.-C. Wang, and P. Debevec, "The Digital Emily Project: Achieving a Photorealistic Digital Actor," *IEEE Computer Graphics and Applications*, vol. 30, no. 4, pp. 20–31, Jul. 2010.

[81] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *2010 20th international conference on pattern recognition*. IEEE, 2010.

[82] H. Yang, H. Zhu, Y. Wang, M. Huang, Q. Shen, R. Yang, and X. Cao, "FaceScape: A Large-Scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 601–610.

[83] S. Berretti, A. D. Bimbo, and P. Pala, "Superfaces: A super-resolution model for 3d faces," in *ECCV Workshops (1)*, 2012.

Alexandros Lattas is a PhD candidate in the Department of Computing, Imperial College London, under the supervision of Prof Stefanos Zafeiriou and Dr Abhijeet Ghosh. He received his BSc in Management & Technology (Software Engineering) from the Athens University of Economics and Business (AUEB), Greece, in 2017. He joined the department of computing at Imperial College London, in October 2017, where he pursued an MSc in Advanced Computing. His interests lie in the field of photorealistic 3D human modeling with Deep Learning, 3D Computer Vision and Graphics.

Stylianos Moschoglou received his Diploma/MEng in Electrical and Computer Engineering from Aristotle University of Thessaloniki, Greece, in 2014. In 2015-16, he pursued an MSc in Computing (specialization Artificial Intelligence) at Imperial College London, U.K., where he completed his project under the supervision of Dr. Stefanos Zafeiriou. He is currently a PhD student at the Department of Computing, Imperial College London, under the supervis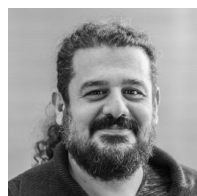ion of Dr. Stefanos Zafeiriou. His interests lie within the area of Machine Learning and in particular in Generative Adversarial Networks and Component Analysis.

Stylianos Ploumpis received the Diploma and MEng in Production Engineering & Management from Democritus University of Thrace, Greece (D.U.T.H.), in 2013. He joined the department of computing at Imperial College London, in October 2015, where he pursued an MSc in Computing specializing in Machine Learning. Currently, he is a PhD candidate/Researcher at the Department of Computing at Imperial College, under the supervision of Dr. Stefanos Zafeiriou. His research interests lie in the field of 3D Computer Vision, Pattern Recognition and Machine Learning.

Baris Gecer is a PhD. student in the Department of Computing, Imperial College London, under the supervision of Dr. Stefanos Zafeiriou. His main research interests are photorealistic 3D Face modeling and synthesis by Generative Adversarial Nets and Deep Learning. He obtained his M.S. degree from Bilkent University Computer Engineering department under the supervision of Prof. Selim Aksoy in 2016 and obtained his undergraduate degree in Computer Engineering from Hacettepe University in 2014.

Abhijeet Ghosh is a Reader (Sr. Associate Professor) in Graphics & Imaging within the Department of Computing at Imperial College London, and an Adjunct Professor of Computer Science at NTNU, Norway. He leads the Realistic Graphics and Imaging group and his current research interests include appearance modeling, and computational illumination and photography for graphics and vision. His research has been supported with a Royal Society Wolfson Research Merit Award, a Google Faculty Research Award, and an EPSRC Early Career Fellowship.

Stefanos Zafeiriou is a Professor in Machine Learning and Computer Vision with the Dept. of Computing, Imperial College London, London, U.K, and an EPSRC Early Career Research Fellow. Between 2016-2020 he was also a Distinguishing Research Fellow with the University of Oulu under Finish Distinguishing Professor Programme. He was a recipient of the Prestigious Junior Research Fellowships from Imperial College London in 2011. He was the recipient of the President's Medal for Excellence in Research Supervision for 2016. He served Associate and Guest Editor in various journals including IEEE Trans. Pattern Analysis and Machine Intelligence, International Journal of Computer Vision, IEEE Transactions on Affective Computing, Computer Vision and Image Understanding, IEEE Transactions on Cybernetics the Image and Vision Computing Journal.

Supplemental Materials for
AvatarMe++: Facial Shape and BRDF Inference with Photorealistic Rendering-Aware GANs

Alexandros Lattas, Stylianos Moschoglou, Stylianos Ploumpis,
Baris Gecer, Abhijeet Ghosh, Stefanos Zafeiriou

✦

1 AVATARME++ GENERALIZATION RESULTS

Training AvatarMe++ with stochastically varied rendering scene parameters, makes the network domain-agnostic to an extend, depending on the degree of variation. In this manner, the generalization of AvatarMe++ is significantly increased, when compared to AvatarMe [1]. Below, we show the results of Fig. 11 of the main manuscript, in high resolution, and compare them with the results of AvatarMe.

Specifically, we acquire various textures with baked illumination from different 3DMM fitting methods [2], [3], datasets [4], [5] and the 3DMD capturing system. We register them to the same template and transform the textures to our topology, before feeding them to the final AvatarMe and AvatarMe++ networks, used for the results in the main manuscript. Please note that only a single network is trained for AvatarMe++ and used for all examples on this document, having a single target environment and the stochastic rendering loss, as explained in our method section. Fig. 1 shows the comparison of generated diffuse albedo, Fig. 2 shows the comparison of generated specular albedo and Fig. 3 shows the comparison of generated specular normals.

REFERENCES

[1] A. Lattas, S. Moschoglou, B. Gecer, S. Ploumpis, V. Triantafyllou, A. Ghosh, and S. Zafeiriou, "AvatarMe: Realistically Renderable 3D Facial Reconstruction "In-the-Wild"," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 760–769.
[2] B. Gecer, J. Deng, and S. Zafeiriou, "Ostec: One-shot texture completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7628–7638.
[3] A. Chen, Z. Chen, G. Zhang, K. Mitchell, and J. Yu, "Photo-realistic facial details synthesis from single image," in *IEEE/CVF International Conference on Computer Vision*, October 2019.
[4] S. Berretti, A. D. Bimbo, and P. Pala, "Superfaces: A super-resolution model for 3d faces," in *ECCV Workshops (1)*, 2012.
[5] H. Yang, H. Zhu, Y. Wang, M. Huang, Q. Shen, R. Yang, and X. Cao, "FaceScape: A Large-Scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR*, 2020, pp. 601–610.
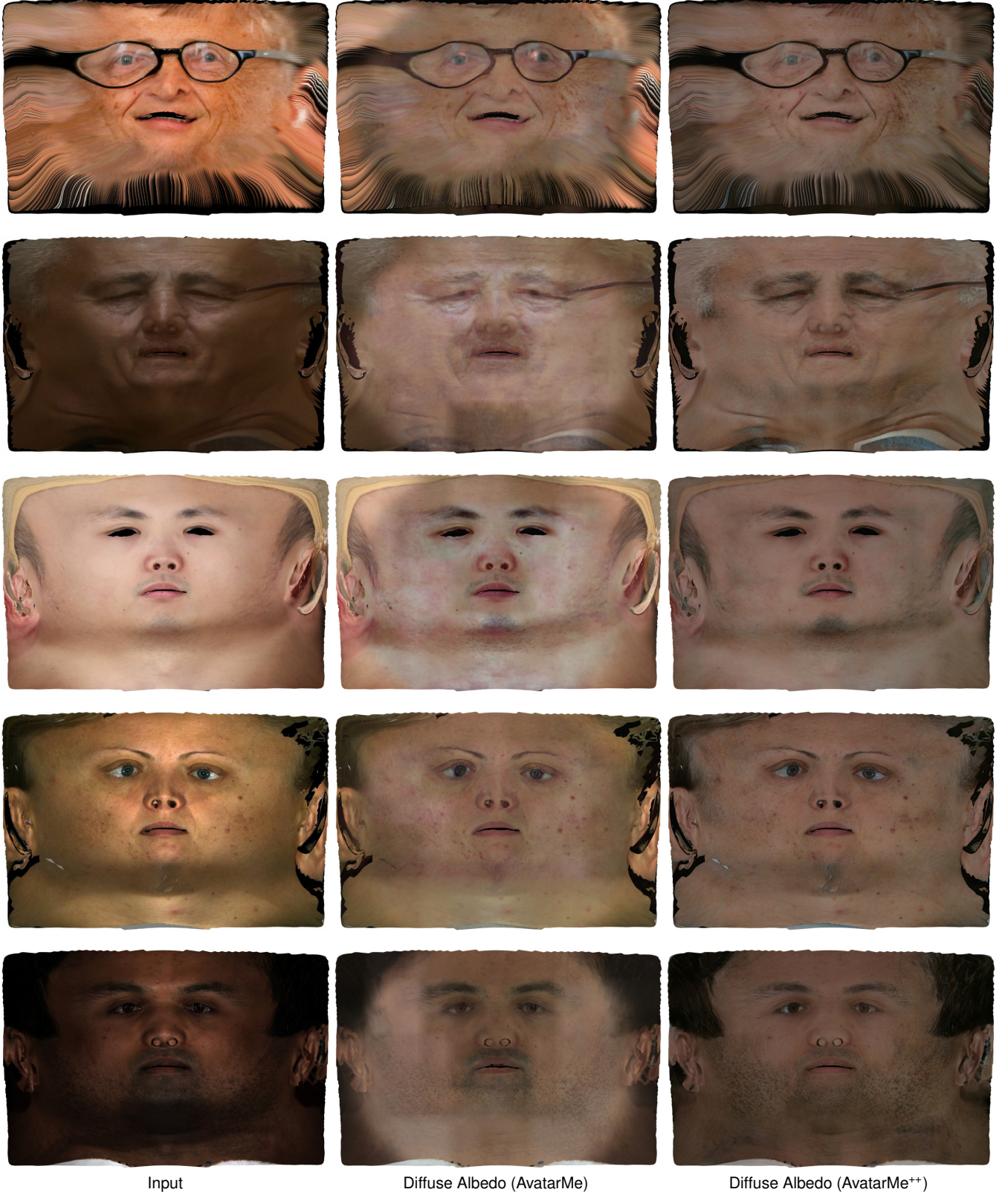
| Input | Diffuse Albedo (AvatarMe) | Diffuse Albedo (AvatarMe++) |

Fig. 1: Generalization of AvatarMe [1], compared to AvatarMe++: Diffuse Albedo from top to bottom: a) Reconstructed subject, with Facial Details Synthesis [3] proxy and texture, b) Reconstructed subject with OSTeC [2] fitting and texture completion, c) Captured subject from FaceScape [5] dataset, d) Captured subject from Superface [4] dataset, e) Captured subject with a 3dMDface system (https://3dmd.com).
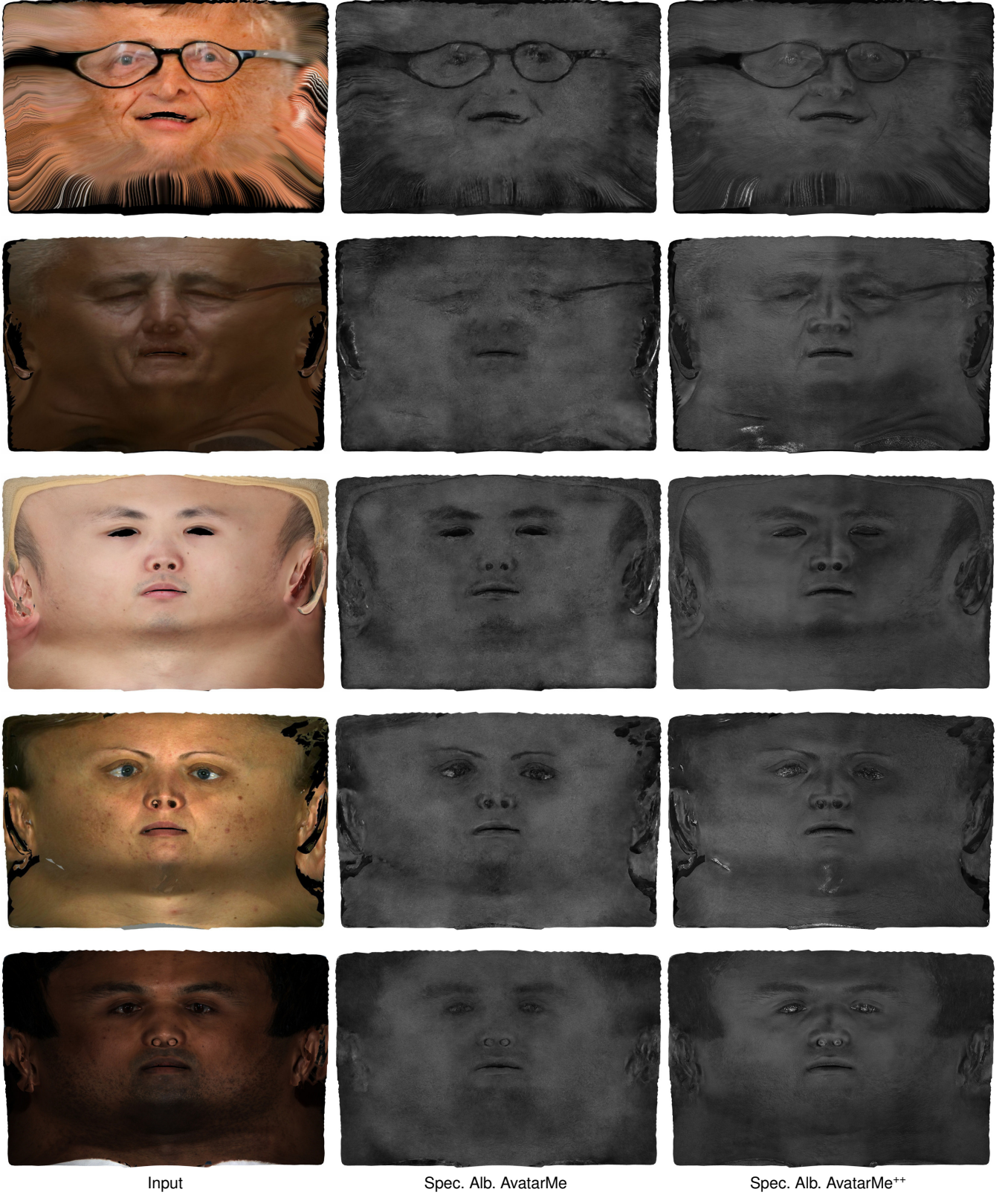
Fig. 2: Generalization of AvatarMe [1], compared to AvatarMe++: Specular Albedo from top to bottom: a) Reconstructed subject, with Facial Details Synthesis [3] proxy and texture, b) Reconstructed subject with OSTeC [2] fitting and texture completion, c) Captured subject from FaceScape [5] dataset, d) Captured subject from Superface [4] dataset, e) Captured subject with a 3dMDface system (https://3dmd.com).

Fig. 3: Generalization of AvatarMe [1], compared to AvatarMe++: Specular Normals (in tangent space) from top to bottom: a) Reconstructed subject, with Facial Details Synthesis [3] proxy and texture, b) Reconstructed subject with OSTeC [2] fitting and texture completion, c) Captured subject from FaceScape [5] dataset, d) Captured subject from Superface [4] dataset, e) Captured subject with a 3dMDface system (https://3dmd.com).