Relightable 3D Head Portraits from a Smartphone Video

Artem Sevastopolsky[1,2] Savva Ignatiev[2] Gonzalo Ferrer[2]
Evgeny Burnaev[2] Victor Lempitsky[1,2]

[1] Samsung AI Center, Moscow, Russia
[2] Skolkovo Institute of Science and Technology (Skoltech), Moscow, Russia

{a.sevastopol, v.lempitsky}@samsung.com

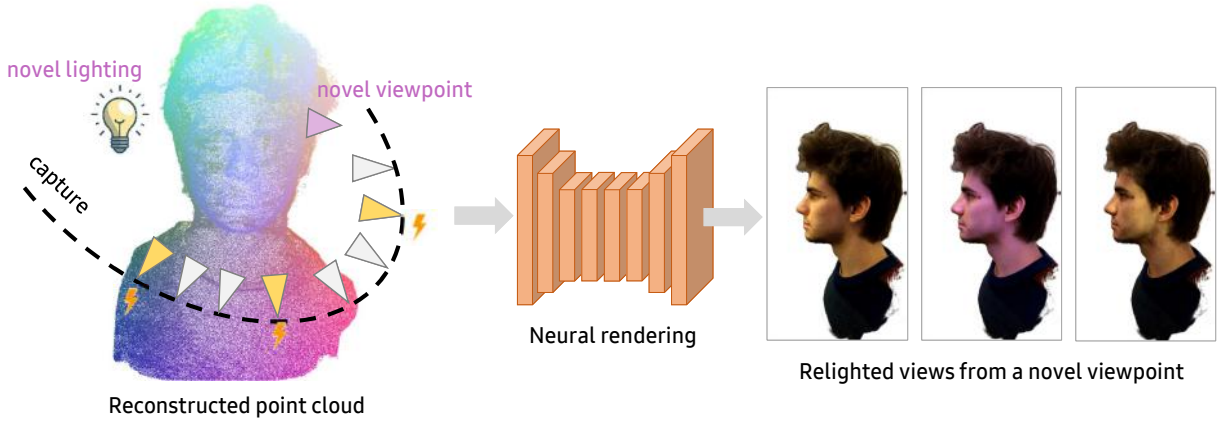

Figure 1: Our method generates relightable 3D portraits. It takes a smartphone video of a person taken with blinking flash. Using Structure-from-Motion (SfM) software, a camera pose is estimated for each of the frames, and a dense point cloud is reconstructed. After fitting, our neural rendering pipeline receives the point cloud rasterized onto a novel camera view. Several maps of lighting-related properties are then predicted. Based on these maps, images relighted with novel lighting conditions can be rendered.

Abstract

In this work, a system for creating a relightable 3D portrait of a human head is presented. Our neural pipeline operates on a sequence of frames captured by a smartphone camera with the flash blinking ("flash-no flash" sequence). A coarse point cloud reconstructed via structure-from-motion software and multi-view denoising is then used as a geometric proxy. Afterwards, a deep rendering network is trained to regress dense albedo, normals, and environmental lighting maps for arbitrary new viewpoints. Effectively, the proxy geometry and the rendering network constitute a relightable 3D portrait model, that can be synthesized from an arbitrary viewpoint and under arbitrary lighting, e.g. directional light, point light, or an environment map. The model is fitted to the sequence of frames with human face-specific priors that enforce the plausability of albedo-lighting decomposition and operates at the interactive frame rate. We evaluate the performance of the method under varying lighting conditions and at the extrapolated viewpoints and compare with existing relighting methods.

1. Introduction

The rise of mobile photography comes hand-in-hand with the pervasiveness of two-dimensional displays. As three-dimensional display devices such as VR headsets, AR glasses, 3D monitors are becoming wide-spread, expanding mobile photography to 3D content acquisition becomes an interesting research direction. In this work, we describe

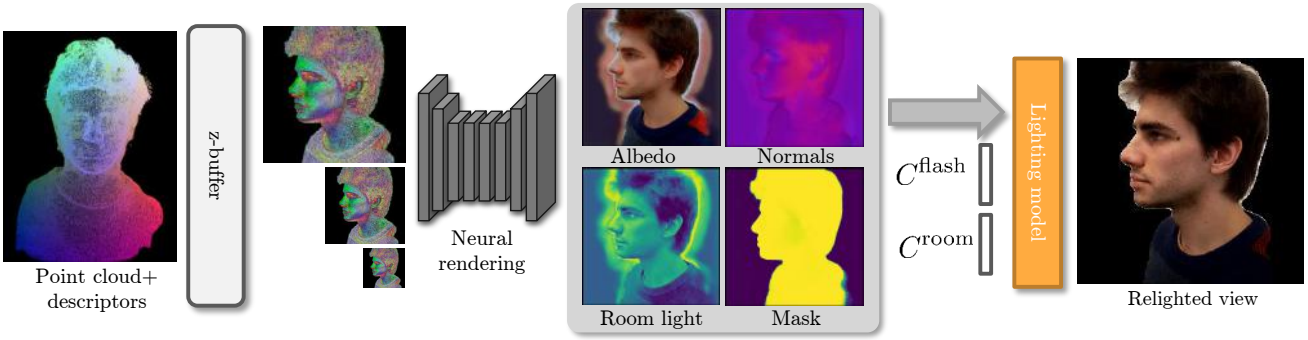

Figure 2: The overview of our rendering process. The point cloud with the neural descriptors is rasterized according to the desired camera position at several resolutions using a simple z-buffer algorithm. The neural rendering process then translates the rasterizations into a set of maps, including the per-pixel albedo, the per-pixel normals, the per-pixel lighting intensity w.r.t. the original room, and the foreground mask. The lighting model can then reproduce the training image, while taking the dominant colors of the flash and the room lighting into account. At test time, new lighting parameters can be plugged into the lighting model instead.

a system for obtaining photorealistic 3D models of human heads or upperbodies (which we refer to as *3D human portraits*) from handheld video sequences, e.g. shot by a smartphone camera.

Generally, acquiring photorealistic 3D models from videos is a heavily-researched direction [26, 3, 2]. However, most 3D displays project the 3D models into either the user surrounding (AR glasses) or the virtual environment (VR headsets, 3D monitors). Thus, to enhance the realism of such models, they should be made *relightable* in a realistic way according to their environment (real or synthetic).

Building realistic *relightable* 3D models is a far less investigated area of research. Existing works either focus on single-view 3D relightable reconstruction [50, 48] (limiting the complexity of the model that can be obtained), or on the acquisition of relightable models using specialized equipment (*light stages*) [11, 47]. In contrast, our system expects only a handheld video of a person made by a smartphone with interleaved flash and no-flash frames. We assume that the person remains static during the video acquisition (which takes few dozens of seconds).

Given such a video, our system estimates a relightable model. We follow the neural point-based graphics approach [3] and use a point cloud estimated by structure-from-motion as a geometric proxy, whereas the photometric information is encoded in the vectorial neural descriptors of individual points in the cloud. The points and their descriptors can then be rasterized for novel camera viewpoints, and the rasterizations are processed by a rendering network [3]. In our case, the rendering network outputs per-pixel albedo values, the normal direction, and the shadow map.

During the scene reconstruction process, we fit this model to the video frames estimating the parameters of the scene lighting and the camera flash along the way. We use human face-specific priors to disentangle lighting and albedo factors. While the priors are only measured in the facial part of the scene, they facilitate disentanglement across the whole 3D portrait. Once reconstructed, the scene (portrait) can be rendered from new viewpoints and with new lighting at interactive speeds.

Our contributions are as follows:

- We propose a new neural rendering model that supports relighting and is based on point cloud geometry.
- We suggest a new albedo-lighting disentangling approach that uses face priors.
- Combining the two contributions, we build a system for relightable 3D model creation that is suitable for human heads and upperbodies, uses handheld videos, and supports realtime rendering.

We evaluate the resulting system on a number of videos, and compare it to the alternative approaches [50, 6].

2. Related work

Creating 3D models from registered video sequences is a problem with a long history [19, 40, 9]. Estimating the photometric properties for such models is known to be a challenging problem in itself [18, 38, 49]. Recently, several works have suggested to model objects using coarse geometry and *neural* photometric descriptors. The rasterization of such descriptors is processed into realistic RGB views using rendering networks. The geometry can take the form of a mesh [39] or a point cloud [3, 20, 41]. Our approach builds on the neural point-based graphics framework [3] and expands it significantly to add the relighting capability.

Estimating surface properties that allow arbitrary relighting is also a research area with long history [7, 27, 23]. There is an emerging direction to estimate surface properties from flash and no-flash images of the same scene [4, 5]. Several approaches focus on relighting 2D portraits without 3D head reconstruction [50, 48]. The recent work [21] estimates the relightable 3D portrait from a single image but restricts the reconstruction to the face (3DMM area) with hair and garment. Several approaches come up with an impressive full body relightable model, yet rely on dedicated light stage equipment [24, 11, 47, 6, 25]. Our work thus achieves a different balance between completeness of the model and the ease of acquisition. It is capable of reconstructing a 3D model of the entire head (and shoulders), and does so from a handheld video sequence that can be acquired from a mobile phone.

Even though our reconstruction is not restricted to the face region, the success of our method relies on the use of face priors. The face region thus serves as a "probe" facilitating the reconstruction. Similar idea has been proposed for the reconstruction of non-relightable head portraits in [43].

3. Method

We start by discussing the lighting model used in our approach. After that, we detail our approach to geometric modeling based on neural point-based graphics. In the end of this section, we discuss the model construction (fitting process), including the loss terms and the prior terms.

3.1. Lighting model

For a given point $\mathbf{x}$ in space which belongs to the surface of a volumetric object, the level of radiance of the light emitted at the location $\mathbf{x}$ in direction ω_o is commonly described by the rendering equation [15]:

$$L_o(\mathbf{x},\omega_o) = \int_S f_r(\mathbf{x},\omega_i,\omega_o) L_i(\mathbf{x},\omega_i) v(\mathbf{x},\omega_i) \cdot \langle -\omega_i, \mathbf{n}(\mathbf{x})\rangle d\omega_i, \tag{1}$$

where $L_i(\mathbf{x},\omega_i)$ defines the incoming radiance to $\mathbf{x}$ in direction ω_i, S stands for the upper hemisphere w.r.t. the surface tangent plane at $\mathbf{x}$ with the unit normal $\mathbf{n}(\mathbf{x})$. Also, $f_r(\mathbf{x},\omega_i,\omega_o)$ is the ratio of scattered light intensity in the direction ω_o and the incoming at direction ω_i, which is usually referred to as the bidirectional reflectance distribution function (BRDF). Furthermore, $v(\mathbf{x},\omega_i)$ is a visibility term (equals to 1 if a point $\mathbf{x}$ is reachable by the light from direction ω_i, or 0 if there is an occlusion in that direction), and $L_o(\mathbf{x},\omega_o)$ defines the total radiance coming from $\mathbf{x}$ in direction ω_0. In terms of this model, BRDF is a material property that describes its spatially-varying light scattering properties. Since all images in our experiments are RGB, values $L_o(\mathbf{x},\omega_o)$, $L_i(\mathbf{x},\omega_i)$, $f_r(\mathbf{x},\omega_i,\omega_o)$ are all in $\mathbb{R}^3$, with each component (channel) calculated independently.

In our setup, we capture the subject using a set of frames taken with an environment lighting, and another set frames in the same environment additionally lighted by a camera flash. To model these two sets of frames, we decompose the incoming radiance with visibility into two terms:

$$L_i(\mathbf{x},\omega_i)\, v(\mathbf{x},\omega_i) = L_i^{\text{room}}(\mathbf{x},\omega_i) v^{\text{room}}(\mathbf{x},\omega_i) + \mathrm{F}\cdot L_i^{\text{flash}}(\mathbf{x},\omega_i) v^{\text{flash}}(\mathbf{x},\omega_i)\,, \tag{2}$$

where $L_i^{\text{room}}(\mathbf{x},\omega_i)$ stands for the environmental (room) light, $L_i^{\text{flash}}(\mathbf{x},\omega_i)$ stands for the flash light, and F indicated if the photo is taken with the flash turned on ($\mathrm{F}=1$) or turned off ($\mathrm{F}=0$). Accordingly, v^{room} and v^{flash} model the light occlusion for the environmental and the flash lighting respectively.

We further assume that the BRDF is Lambertian (constant at each point) $f_r(\mathbf{x},\omega_i,\omega_o)=\rho(\mathbf{x})$ (with the normalizing constant omitted), i.e. corresponds to a diffuse surface, and $\rho(\mathbf{x})$ stands for the surface albedo at $\mathbf{x}$. We note that through the use of neural rendering, our system can model some amount of non-Lambertian effects since the albedo in our model can be effectively made view-dependent. Applying the modifications to (1), we get:

$$\begin{aligned} L_o(\mathbf{x},\omega_o) = \\ \rho(\mathbf{x}) \int_S L_i^{\text{room}}(\mathbf{x},\omega_i) v^{\text{room}}(\mathbf{x},\omega_i) \cdot \langle -\omega_i, \mathbf{n}(\mathbf{x})\rangle d\omega_i \\ + \mathrm{F}\cdot\rho(\mathbf{x}) \int_S L_i^{\text{flash}}(\mathbf{x},\omega_i) v^{\text{flash}}(\mathbf{x},\omega_i) \cdot \langle -\omega_i, \mathbf{n}(\mathbf{x})\rangle d\omega_i \end{aligned} \tag{3}$$

Suppose now that the integral in the first part is equal to $\overline{s}(x) \in \mathbb{R}^3$ and defines the shadowing caused both by the room lamps and occlusions (e.g. a nose casting shadow on a cheek, in case of a human head). We are going to model it as a product of color temperature and grayscale shadowing $\overline{s}(x) = C^{\text{room}} \cdot s(x)$, $C^{\text{room}} \in \mathbb{R}^3$, $s(x) \in \mathbb{R}$. As for the second (flash) part, we explicitly model the incoming light radiance $L_i^{\text{flash}}(\mathbf{x},\omega_i) = \frac{C^{\text{flash}}}{d(\mathbf{x})^2}$, where $d(\mathbf{x})$ is the distance from the flash to $\mathbf{x}$, and $C^{\text{flash}} \in \mathbb{R}^3$ is a constant vector proportional to the color temperature and intensity of the flash. In our experiments we assume that the flash is far enough, and hence, $d(\mathbf{x}) \approx d$, with d being the distance from the camera to its closest point in the point cloud, and the light rays from the flash are approximately parallel. Since on a smartphone, flashlight and the camera lens are usually co-located at close proximity (at ω_o), we assume that $v^{\text{flash}}(\mathbf{x},\omega_i) = 1$ for all $\mathbf{x}$ observed in a flashlighted image. This brings us to the transformation of (3) into (4):

$$L_o(\mathbf{x}, \omega_o) = \rho(\mathbf{x}) C^{\text{room}} s(x) + \mathrm{F} \cdot \rho(\mathbf{x}) \frac{C^{\text{flash}}}{d^2} \langle \mathbf{n}(\mathbf{x}), -\omega_o \rangle \tag{4}$$

Note that in (4), the decomposition of the light intensity $L_o(\mathbf{x}, \omega_o)$ into separate components is ambiguous, as is common to most lighting-albedo estimation problems. In particular, there is an inverse proportional relationship between $\rho(\mathbf{x})$ and both C^{room} and C^{flash}. This ambiguity will be resolved in Section 3.3 through the use of appropriate priors.

3.2. Geometric modeling

After discussing the lighting model, we proceed to the geometric modeling. We assume that a set of RGB photographs of a human head $\mathcal{I} = \{I_1, \dots, I_P\}$, $I_k \in \mathbb{R}^{\mathrm{H} \times \mathrm{W} \times 3}$, all are taken by a mobile camera in the same environment and featuring the head from various angles. A subset $\mathcal{I}' = \{I_{s_1}, \dots, I_{s_M}\}$ of photos features the head additionally lighted by a camera flash. As a first step, we use structure-from-motion (SfM) methods to reconstruct camera poses $C_1, \dots, C_P$ of each image: $C_k = (K_k, [R_k, t_k])$. Then, similarly, we reconstruct a dense point cloud $\mathcal{P} = \{p_1, \dots, p_N\}$, $p_i = \{x_i, y_i, z_i\}$ from all images. In our experiments, we used Agisoft Metashape [1], while any other software such as COLMAP [32, 33], etc. could be employed instead.

Segmentation and filtering. As we aim to model the subject without background, we estimate the foreground segmentation of individual photographs, and filter out the estimated point cloud along the way. We use the newly proposed U^2-Net [28] segmentation network designed for a salient object segmentation task. We fine-tune the pretrained U^2-Net model with a warm-up learning schedule on the Supervisely [37] human segmentation dataset to make it more suitable for our task. After that, we pass photographs $\mathcal{I}$ through the fine-tuned model and obtain the sequence of initial 'soft' masks $\mathcal{M}^0 = \{M_1^0, \dots, M_P^0\}$, $M_k^0 \in \mathbb{R}^{\mathrm{H} \times \mathrm{W} \times 1}$. Finally, we add multi-view consistency of the silhouettes via visual hull estimation [22] using point cloud as a geometric proxy, and using the weights of the segmentation network as a parameterization. This is done by fine-tuning the weights of the segmentation network to minimize the inconsistency of segmentations across the views, which results in obtaining the refined masks $\mathcal{M} = \{M_1, \dots, M_P\}$.

3.2.1 Neural rendering.

Our rendering approach (Figure 2) is based on a deep neural network predicting albedo, normals, and room shadows from a point cloud rasterized onto each camera view, and follows the Neural Point-Based Graphics (NPBG) pipeline [3] to accomplish that. Following their approach, points in the point cloud are augmented with neural descriptors, i.e. multi-dimensional latent vectors characterizing point properties: $\mathcal{D} = \{d_1, \dots, d_N\}$, $d_k \in \mathbb{R}^L$ ($L = 8$ in all our experiments).

The neural rendering stage then starts with the rasterization of the descriptors onto the canvas associated with the camera C_k. This is performed the way suggested in [3]: the *raw image* $S[0] = S[0](\mathcal{P}, \mathcal{D}, C_k) \in \mathbb{R}^{H \times W \times L}$ is formed by Z-Buffering the point cloud (for each pixel, finding the closest to the camera point which projects to this pixel). The descriptor of each of the closest points is assigned to the respective pixel. In case there are no point which project to some pixel, a null descriptor is assigned to that pixel instead. Similarly, we construct a set of auxiliary raw images $S[1], \dots, S[T]$ of spatial sizes $\frac{H}{2^t} \times \frac{W}{2^t}$ and perform the rasterization of descriptors onto these images by the same algorithm. A pyramid of the raw images is introduced to cover the point cloud at several scales. The highest resolution raw image $S[0]$ features the richest spatial detail, while the lower resolution ones feature less surface bleeding (see [3] for the detailed explanation).

The second step involves transforming the set of raw images $S[0], \dots, S[T]$ by a deep neural network $f_\phi(S[0] \dots, S[T])$ closely following the U-Net [31] structure with gated convolutions [44]. As suggested in [3], each of the raw images is passed as an input (or concatenated) to the first layer of the network encoder of the respective resolution. The output of the rendering network is the set of dense maps (in the case of [3], the maps contain output RGB values).

In our case, the last layer of the network outputs an eight-channel tensor with several groups:

- The first group contains three channels and uses sigmoid non-linearity. These channels contain the albedo values $A \in \mathbb{R}^{H \times W \times 3}$. Each pixel of A describes the spatially-varying reflectance properties (albedo $\rho(\mathbf{x})$) of the head surface (skin, hair, or other parts).

- The second group also has three channels, and uses groupwise L_2 normalization in the end. This group contains the rasterized normals $N \in \mathbb{R}^{H \times W \times 3}$, with each pixel containing a normal vector $\mathbf{n}(\mathbf{x})$ of the point at the head surface in the world space.

- Next, one-channel group that uses the sigmoid non-linearity, corresponds to the grayscale shadowing $S \in \mathbb{R}^{H \times W}$, caused by the variations in the room lighting and consequently occurring occlusions.

- Finally, the last one-channel group that also uses the sigmoid non-linearity, defines the segmentation mask $M \in \mathbb{R}^{H \times W}$, with each pixel containing the predicted

probability of the pixel belonging to the head and not the background.

Given the output of the rendering network, the final rendered image is defined by fusing the albedo, normals, and room shadows, as prescribed by the lighting model (4):

$$\mathcal{I} = A \cdot C^{\text{room}} \cdot S + \text{F} \cdot A \frac{C^{\text{flash}}}{d^2} \langle N, -\omega_o \rangle, \qquad (5)$$

where the scalar product $\langle N, -\omega_o \rangle$ is applied to each pixel. The color temperature vectors C^{room} and C^{flash} both in $\mathbb{R}^3$ are considered part of the model and are shared between all pixels. These vectors are estimated from training data as discussed below. At inference time, feature maps A and N, as well as $C^{\text{room}}, C^{\text{flash}}$ parameters, can be used to render the human head under a different modified lighting, such as directional lighting ($\mathcal{I}' = A \cdot \lfloor \langle N, -\omega \rangle \rfloor$) or other models such as spherical harmonics (SH, see [29, 10]). In the latter case, the albedo A is multiplied by a non-linear function of the pixel normals N that incorporates the SH coefficients. When a 360° panorama of a new environment is available, one can obtain the values of the SH coefficients (up to the predefined order) by the panorama integration. As described in [29], choosing the order of 3 or higher, which results in at least 9 coefficients per color channel, can often yield the expressive lighting effects. In case of the third-order SH, the relighted image is defined as a quadratic form:

$$\mathcal{I}'[i,j] = A[i,j] \cdot [N[i,j]\ 1]^T \cdot \mathbf{M}(\text{SH}_{\text{coef}}) \cdot [N[i,j]\ 1], \qquad (6)$$

where the 4×4 matrix $\mathbf{M}(\text{SH}_{\text{coef}})$ linearly depends on the 27 SH coefficients SH_{coef} (the expression for $\mathbf{M}$ can be found in [29]).

3.3. Model fitting

Generic scene losses. Our model contains a large number of parameters that are fitted to the data. During fitting, the obtained images $\mathcal{I}$ (5) are compared with the ground truth image $\mathcal{I}^{\text{gt}}$ by a loss constructed from several components that are described below.

The **main loss** equals to the evaluated mismatch between the estimated lighted image $\mathcal{I}$ and the ground truth image $\mathcal{I}^{\text{gt}}$:

$$\begin{aligned} &L_{\text{final}}(\phi, \mathcal{D}, C^{\text{room}}, C^{\text{flash}}) \\ &\quad = \Delta\big(\mathcal{I}(\phi, \mathcal{D}, C^{\text{room}}, C^{\text{flash}}), \mathcal{I}^{\text{gt}}\big), \end{aligned} \qquad (7)$$

whereas the mismatch function Δ is employed to compare pair of images:

$$\Delta(I_1, I_2) = \text{VGG}(I_1, I_2) + \beta \cdot L_1(\text{pool}(I_1), \text{pool}(I_2)) \qquad (8)$$

Here, $\text{VGG}(\cdot,\cdot)$ is a perceptual mismatch based on the layers of VGG-16 network [34], $L_1(\cdot,\cdot)$ refers to the mean absolute deviation, $\text{pool}(I)$ is the average pooling of an image I with $K \times K$ kernel ($K = 4$ in our experiments) and β is a balancing coefficient introduced to equalize the range of values of two terms ($\beta = 2500$ was selected). While VGG encourages matching of high-frequency details, L_1 rewards matching colors. A naive optimization of L_1 can lead to blurring and loss of details, therefore, we evaluate the L_1-term over the downsampled images.

Since the system must be able to segment the rendered head from an arbitrary viewpoint, we introduce the **segmentation loss** constraining the predicted mask M:

$$L_{\text{mask}}(\phi, \mathcal{D}) = -\log \text{Dice}(M(\phi, \mathcal{D}), M_i), \qquad (9)$$

where the Dice function is a common choice of the mismatch for segmentation [36] (evaluated as the pixel-wise F1 score). Effectively, with this loss, the network learns to extrapolate the precalculated masks $\mathcal{M}$ to new viewpoints.

For non-flashlighted images, the rendered image is equal to $\mathcal{I} = A \cdot C^{\text{room}} \cdot S$ according to (5). In practice, this creates a certain ambiguity between the learned maps A and S. Since A participates in both terms of (5), the high-frequency component of the renderings tends to be saved in S by default. The following **room shading loss** implicitly requires A to be fine-grained instead:

$$L_{\text{TV}}(\phi, \mathcal{D}) = \text{TV}(S(\phi, \mathcal{D})) \qquad (10)$$

where TV is the Total Variation loss based on L_1.

Face prior losses. To further regularize the learning process, we use the particular property of the scene that we handle the presence of face regions. For that, for each training image we perform face alignment using a pretrained PRNet system [8]. Given an arbitrary image I containing a face, PRNet estimates a *face alignment map*, i.e. a tensor Posmap of size 256 x 256 that maps the UV coordinates (in a predefined, fixed texture space associated with the human face) to the screen-space coordinates of the image I. Let $\text{Posmap}_1, \ldots, \text{Posmap}_P$ define the position maps calculated for each image in the training sequence. By the operation $I \odot \text{Posmap}$, we denote the bilinear sampling (backward warping) of an image I onto Posmap, which results into mapping the visible part of I into the UV texture space. The mapping thus constructs a colored (partial) face texture.

We employ the gathered face geometry data in two ways. Firstly, throughout the fitting we estimate an albedo half-texture $\mathcal{T}_A$ of size 256×128 (only for the left part of the albedo). $\mathcal{T}_A$ is initialized by taking a pixel-wise median of the projected textures for all flashlighted images $\mathcal{T}_F = \text{median}(I_{s_1} \odot \text{Posmap}_{s_1}, \ldots, I_{s_M} \odot \text{Posmap}_{s_M})$ and averaging left and flipped right halves. The **symmetry**

loss is a facial prior that encourages the symmetry of albedo by comparing it with the learned albedo texture:

$$L_{\text{symm}}(\phi, \mathcal{D}, \mathcal{T}_A) = \Delta(A(\phi, \mathcal{D}) \odot \text{Posmap}_i, \\ [\mathcal{T}_A,\ \text{flip}(\mathcal{T}_A)]) \tag{11}$$

(mismatch is evaluated only where $A(\phi, \mathcal{D}) \odot \text{Posmap}_i$ is defined). $[\mathcal{T}_A,\ \text{flip}(\mathcal{T}_A)]$ denotes the concatenated texture and its horizontally flipped version. Note that this loss both makes albedo more symmetric and matches albedo texture $\mathcal{T}_A$ colors with the ones of the learned albedo. The balance between these two factors is controlled by the learning rate for $\mathcal{T}_A$. Incorporation of this loss is mainly inspired by the recent work [42] and employed here in a simpler setting (without confidence estimation), as the UV texture space of PRNet is symmetric by design. For our task, the symmetry loss helps resolving the decomposition of an image into albedo and shadows, as the opposite points of the face (e.g. left and right cheeks) can have similar albedo, while casted shadows are most often non-symmetric.

Additionally, to select the gamma for albedo, we introduce the **albedo color matching loss** $L_{\text{cm}}(\phi, \mathcal{D}) = L_1(A(\phi, \mathcal{D}) \odot \text{Posmap}_i, \mathcal{T}_F)$ (also calculated only for valid texels of $A(\phi, \mathcal{D}) \odot \text{Posmap}_i$).

Another type of data which can be inferred from PRNet outputs is the normals for the face part. Each face alignment map, along with the estimated depth (also by PRNet) and a set of triangles, defines a triangular mesh. We render the meshes estimated for each view, smoothen them, calculate the face normals and render the projected normals onto the respective camera view (technically, all performed in the Blender rendering engine). Then, the rendered normals are rotated by the camera rotation matrix for their conversion to the world space. We will introduce the notation $N_1, \dots, N_P$ for the estimated normal images for the facial part of size $H \times W$. The normals predicted by the network are matched with the PRNet normals at the facial region (defined by the mask M_{face}) by the **normal loss**:

$$L_{\text{normal}}(\phi, \mathcal{D}) = \Delta(N(\phi, \mathcal{D}) \cdot M_{face},\ N_i \cdot M_{face}) \tag{12}$$

The composite loss is expressed as follows:

$$\begin{aligned} L(\phi, \mathcal{D}, C^{\text{room}}, C^{\text{flash}}, \mathcal{T}_A) &= L_{\text{final}}(\phi, \mathcal{D}, C^{\text{room}}, C^{\text{flash}}) \\ &+ \alpha_{\text{normal}} L_{\text{normal}}(\phi, \mathcal{D}) + \alpha_{\text{symm}} L_{\text{symm}}(\phi, \mathcal{D}, \mathcal{T}_A) \\ &+ \alpha_{\text{cm}} L_{\text{cm}}(\phi, \mathcal{D}) + \alpha_{\text{TV}} L_{\text{TV}}(\phi, \mathcal{D}) \\ &+ \alpha_{\text{mask}} L_{\text{mask}}(\phi, \mathcal{D}) \end{aligned} \tag{13}$$

In our experiments, the separate losses were balanced as follows: $\alpha_{\text{normal}} = 0.1$, $\alpha_{\text{symm}} = 0.02$, $\alpha_{\text{cm}} = 100$, $\alpha_{\text{TV}} = 50$, $\alpha_{\text{mask}} = 10^3$.

Optimization. The learnable part of the system is trained for one scene by the backpropagation of the loss to the rendering network parameters ϕ, point descriptors $\mathcal{D}$, and to the auxiliary parameters $C^{\text{room}}, C^{\text{flash}}, \mathcal{T}_A$. We use Adam [17] with the same learning rate for ϕ and $\mathcal{D}$, while the rest of learnable parameters feature different learning rates, empirically selected according to the range of their possible values. At each step, a sample training image is selected, and a forward pass followed by the gradient step is made. Train-time augmentations include random zoom in/out and subsequent cropping of a small patch.

4. Experiments

4.1. Real head portraits

We now show results for our method. Our experimental dataset consists of five sequences taken by a mobile camera of Samsung Galaxy S8 smartphone. Capturing was made in Open Camera app [12] with controlled white balance, 30 FPS, 1/200 s shutter speed and ISO of approximately 200 (selected for each sequence according to the lighting conditions). During each recording, the camera flash was blinking regularly once per second, each time turning on for $\sim$ 0.1 s. Each subject was photographed for 15-25 seconds and was asked to remain still during shooting. The individual frames were extracted from the resulting videos at 6 FPS, which included all the flashlighted photos. The camera poses and point clouds were reconstructed by Agisoft Metashape [1] with *Highest* precision of cameras alignment and *High* quality of point cloud reconstruction; the latter is a tradeoff between the geometry completeness and the number of points. Normals and meshes for the face part were collected by inferring PRNet [8] with MTCNN [45] face detector for each of the captured frames. The meshes estimated by PRNet were smoothened by five iterations of Laplacian smoothing [35].

For each subject, ten frames (five flashlighted and five non-flashlighted) covering a head from all sides were selected for validation, while the rest were used for training. Our method is fitted to each sequence separately. The pipeline is trained for 80'000 steps. At each step, a random frame (and the respective camera viewpoint) is selected, and a random patch of size 512×512 is extracted by sampling the patch center inside from the foreground region of the ground truth mask. In Fig. 3, the results of simultaneous relighting and view interpolation are depicted. For this evaluation, we show the results at novel viewpoints and directional+ambient lighting (three sample light directions were selected as up, right and forward vectors naturally associated with the frontal head view). The final rendered image under such conditions is given by: $\mathcal{I}' = A \cdot (\alpha + (1 - \alpha) \cdot \lfloor \langle N, -\omega \rangle \rfloor)$, where α stands for the ambient light, emitted regardless of the normal direc-

tion ($\alpha = 0.5$ was taken for the evaluation), ω defines the novel light direction, and the dot product $\langle N, -\omega \rangle$ is applied pixel-wise. Note that the normals are challenging to estimate for regions which remain completely black under the flashlight (e.g. a black jacket).

A similar comparison is shown in Fig. 4 for the case of *additional lighting*, i.e. adding a new point light source, similar to flash light, to the captured environmental lighting of the room. In this case, a head is relighted by plugging novel d (distance from the novel light source) and novel ω_o (direction of the novel light source) into (4). The renderings can look more appealing in the additional lighting setting than in the relighting setting in certain cases, as the estimated albedo is only rendered here in conjunction with soft shadows and shading.

In Fig. 5, the results of the comparison with a recently released DPR method [50] are presented. To make the comparison feasible, we relight the head by a combination of ambient light and 1^{st}-order Spherical Harmonics (SH) [29]. DPR is based on an end-to-end neural network that receives a single image, a new SH lighting to be applied, and returns a relighted image. We cropped inputs to DPR by a face localization network [45] and enlargening the bounding boxes to make them closer to the cropping of faces in the training set of DPR (CelebA-HQ [16]). In contrast to our method, DPR requires only one image and is able to realistically relight it, though it can be seen that some areas can be incorrectly reshaded at the side views. Our method operates on a video covering all head parts, but produces more consistent lighting and can additionally render a head from novel viewpoints. Performance of relighting with simultaneous view resynthesis is showcased in Fig. 6. There, the renderings in additional lighting setting are showcased along with the nearest views from the captured video. The nearest frame is selected by taking 5% of train viewpoints with the most similar view angle and then selecting the view with the closest viewpoint among those. Since random zoom-in/zoom-out and random cropping are used at the training stage, angle deviation is more critical when comparing the predicted result to the nearest frame.

4.2. Synthetically created people

In addition to the main experiments with smartphone videos, we conducted evaluations on synthetically rendered data. This comparison is based on the upper-body part of three full-body models from RenderPeople [30]. Each model is defined as a mesh with diffuse texture, normal texture, and other parameters. Using Blender Eevee rendering engine, we closely simulate the flash - no flash acquisition scheme employed for real portraits. The half-circular camera trajectory is drawn in front of the subject, with each view covering the head and shoulders of the model, and is evenly divided into 100 viewpoints. For every fifth viewpoint, a directional light source of constant power is added to the scene, co-located with the camera. The lighted subject is rendered for each of the 100 viewpoints in 3000×3000 resolution and 32-bit floating point HDR format without compression. Point clouds with 7 million points are uniformly sampled from the models meshes.

Similarly to the smartphone videos preprocessing pipeline, here we process each rendered frame with PRNet [8], following the same procedure conducted for real portraits (it performs at the similar level of accuracy on synthetic data from RenderPeople). Additionally, we precompute the renderings of each model colored with its diffuse texture (albedo render) and normals texture (normal render with world-space XYZ colored as RGB). Among 100 frames, we selected every 20^{th} (a flashlighted one), as well as one preceding and one subsequent frame for every one of them for validation (resulting in 30 frames total), while the rest were used for training (fitting). In contrast to the real head portraits, here we initialize the albedo texture $\mathcal{T}_A$ not by the median flashlighted texture $\mathcal{T}_F$, but by the median **non**-flashlighted texture $\mathcal{T}_{NF}$:

$$\begin{aligned} \mathcal{T}_{NF} = \text{median}(&I_{p_1} \odot \text{Posmap}_{p_1}, \ldots, I_{p_T} \odot \text{Posmap}_{p_T}), \\ &\{p_1, \ldots, p_T\} \cup \{s_1, \ldots, s_M\} = \{1, \ldots, P\}, \\ &\{p_1, \ldots, p_T\} \cap \{s_1, \ldots, s_M\} = \varnothing \end{aligned} \tag{14}$$

(less formally, $p_1, \ldots, p_T$ denote the indices of non-flashlighted captured frames). This is done here to make the comparison between the estimated albedo and the ground truth albedo more precise by correctly matching their color temperature.

The comparison of the relighted image with the ground truth on validation is presented in Table 1. To make the comparison with the DPR baseline [50] feasible, the images for the metrics evaluation in the Table 1 were relighted by a combination of ambient light and 1^{st}-order Spherical Harmonics [29] (see the subsection 4.1). Since DPR requires an image to be relighted, we only used non-flashlighted frames for the metrics evaluation. We cropped all images by a face localization network [45] and enlargened the bounding boxes to make them closer to the cropping of the faces in the training set of DPR (CelebA-HQ [16]). The predicted renderings of our method were cropped by the same bounding boxes for the comparison. Furthermore, we found that DPR significantly changes the color temperature of the original photo and alters the contribution of the direction light compared to the ambient light. Because of that, for the sake of fair comparison, the contribution of ambient and directional components of spherical harmonics passed to DPR were manually reweighted to match the color temperature of the ground truth as much as possible. The evaluation is based on three commonly used perceptual metrics

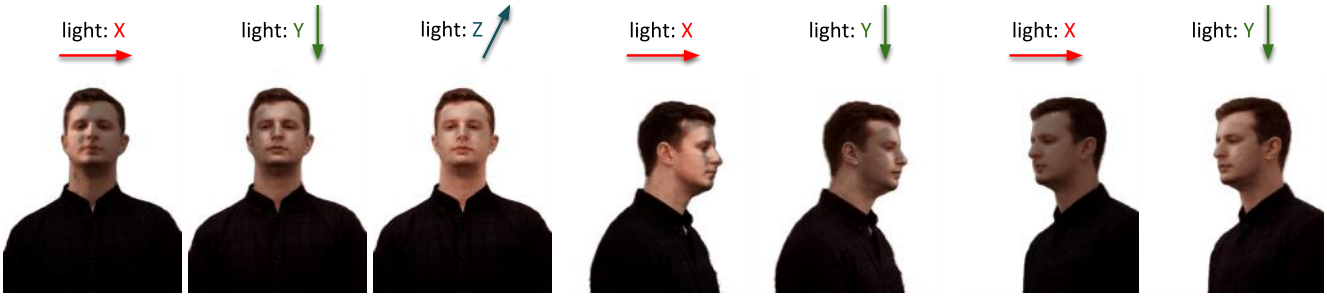

Figure 3: Qualitative comparison of the smartphone-captured real head portraits rendering under several novel viewpoints and novel (non-colocated) directional light. Light directions in the top row are defined in the **world space** coordinate axes associated with the frontal head view. The people are additionally lighted by constant ambient (indirectly emitted) light for better appearance. *Electronic zoom-in recommended.*

Figure 4: Adding the light to the original sequences. Compared to the previous figure (*relighting* case), here we investigate *additional lighting* – the case when a point light is introduced while the room lighting is left in place. The color of a point light is chosen randomly for each rendering among red, green, blue, purple, yellow, and brown. The light directions are the same as in Fig. 3. The results are for the viewpoints from validation. *Electronic zoom-in recommended.*

(VGG [14], LPIPS [46], FID [13]) calculated over images with the masked background (for the predictions of both methods and the ground truth). The results are averaged over three people, ten non-flashlighted validation frames for each of them, and three directions of 1^{st}-order SH (*left-to-right, up-to-down, forward-to-backward* in the world space associated with the frontalized head). The examples of the renderings by our method and DPR under these conditions are presented in Fig. 7.

Fig. 8 depicts the difference between the predicted albedo and normals and the ground truth ones. In addition, the rendered normals for the facial part obtained from PRNet [8] meshes are presented. Normals from PRNet are the only source of direct supervision for normals during the pipeline training. They are predicted by PRNet with a certain error (especially for the side pixels at each view), which partially explains the general difference between the predicted normals and the ground truth normals. This effect can also be attributed to the non-uniqueness of normals, required to fit the flashlighted training images, which might happen due to the limited supervision and co-located lighting used during training. Note, however, that the network is

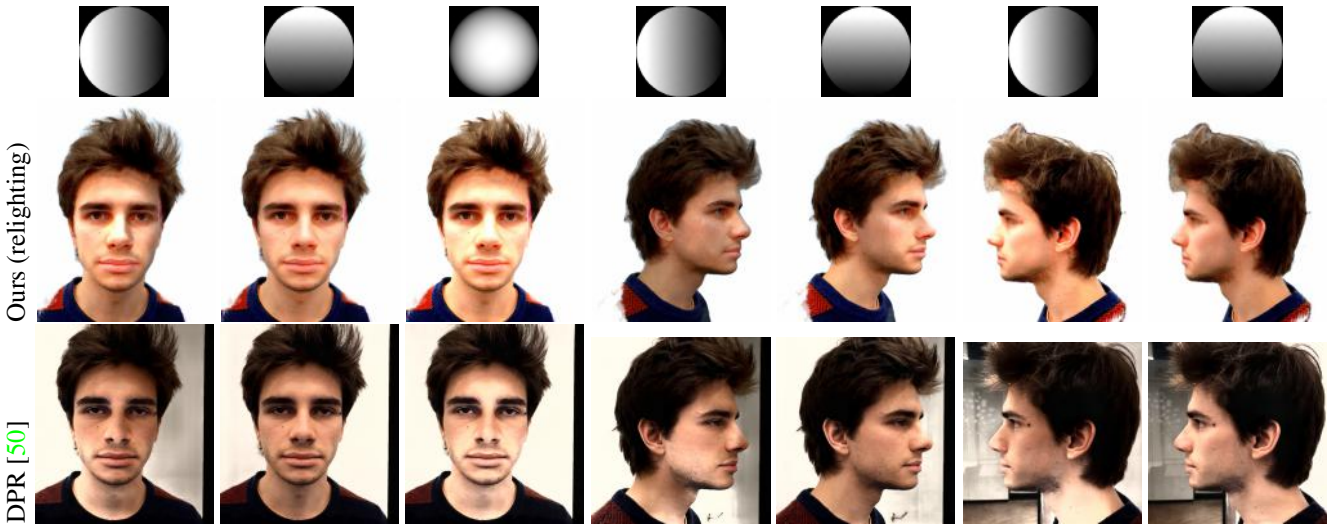

Figure 5: Qualitative comparison of the sample smartphone-captured real head portrait under several viewpoints from validation (corresponding to non-flashlighted images) and Spherical Harmonics (SH) lighting, corresponding to ambient and 1^{st}-order SH light. Under such lighting, our system can be compared with others, such as the DPR [50] method. The top row depicts a sphere, lighted by the selected SH lighting and rendered from the respective viewpoint. While DPR is able to realistically reshade an image just from a single photo, we observe it is sensitive to cropping and can create inconsistent lighting at the side views.

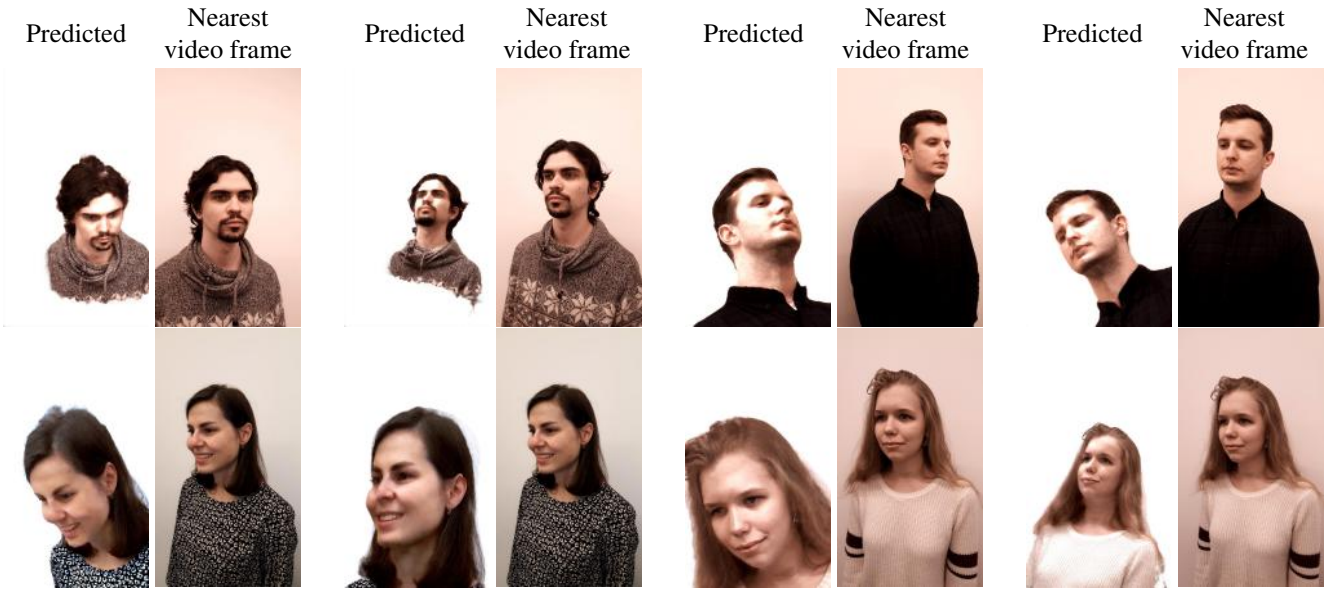

Figure 6: Our models rendered from viewpoints selected far from the trajectory of capture. In each case, the nearest frame from the captured video is shown alongside. The nearest frame is selected by taking 5% of train viewpoints with the most similar view angle and then selecting the view with the closest viewpoint among those. The models are rendered with additional randomly-directed light. The segmentation artefacts near the top of the head can be attributed to the sparseness of point clouds in that region resulting from our capture protocol. *Electronic zoom-in recommended.*

able to propagate the normals learned for the face region to the head and upper body (to a certain extent).

4.3. Ablation study

Fig. 9 reflects the comparison with several ablations conducted on a real head portrait, created by: removing sin-

	Person 1 (Carla)		
	VGG ↓	FID ↓	LPIPS ↓
Ours (Final result)	**174.59**	**61.664**	**0.0815**
DPR [50] (Final result)	305.96	69.263	0.129

	Person 2 (Claudia)		
	VGG ↓	FID ↓	LPIPS ↓
Ours (Final result)	**286.94**	**58.713**	**0.1380**
DPR [50] (Final result)	331.88	59.791	0.1441

	Person 3 (Eric)		
	VGG ↓	FID ↓	LPIPS ↓
Ours (Final result)	**259.14**	51.158	**0.1259**
DPR [50] (Final result)	335.07	**31.467**	0.1418

Table 1: Quantitative comparison of the relighted images for 3 synthetic people from RenderPeople dataset. For each of the people, the final result is the image relighted by a combination of ambient light and 1^{st}-order Spherical Harmonics (SH). In this setting, the comparison with the DPR method [50] is possible. The ground truth was constructed from albedo and normals rendered from mesh in Blender. To report the results, we use three perceptual metrics evaluated on images for ten validation viewpoints corresponding to the non-flashlighted images from the validation. The result is averaged across all of these viewpoints and three ambient+spherical harmonics lightings (with first-order spherical harmonics simulating *left-to-right*, *up-to-down*, *forward-to-backward* directional light). The predictions under these lighting conditions are presented in Fig. 7.

gle loss term, excluding the point cloud filtering stage from the preprocessing pipeline, or leaving only VGG term in Δ mismatch. Note the difference in normals reconstruction without L_{normal} (in this case, the network predicts normals closer to average due to the lack of geometric information) and with only VGG term in Δ (results in worse color matching of normals). The albedo becomes more blurry without L_{TV} (zoom-in is recommended), while the shadows are captured less accurately without L_{symm}.

5. Discussion

We have presented a pipeline for creating virtual portraits of humans based on smartphone-captured videos. Starting from the 3D rendering approach from Neural Point-Based Graphics [3], we modify it by introducing the decomposition of a rendered image into albedo, normals, and room shadows. After training, the mappings can be rendered for an arbitrary viewpoint and with varying lighting. We demonstrate the relighting ability for simple models, such as directional light, and for more complex ones, such as spherical harmonics. Compared to most of the similar pipelines, ours does not require complex equipment for the data acquisition.

The method has certain limitations. First of all, the view interpolation ability is softly constrained by the trajectory of capturing, and the rendering quality can decrease when a viewpoint far from train is selected. Also, the approach models occlusions caused by other head parts (e.g. nose casting shadow on a cheek) only at train time, but not at test time. Finally, the method performs the best in the *additional lighting* setting, i.e. the case when new lighting is added, but the room lighting is left in place.

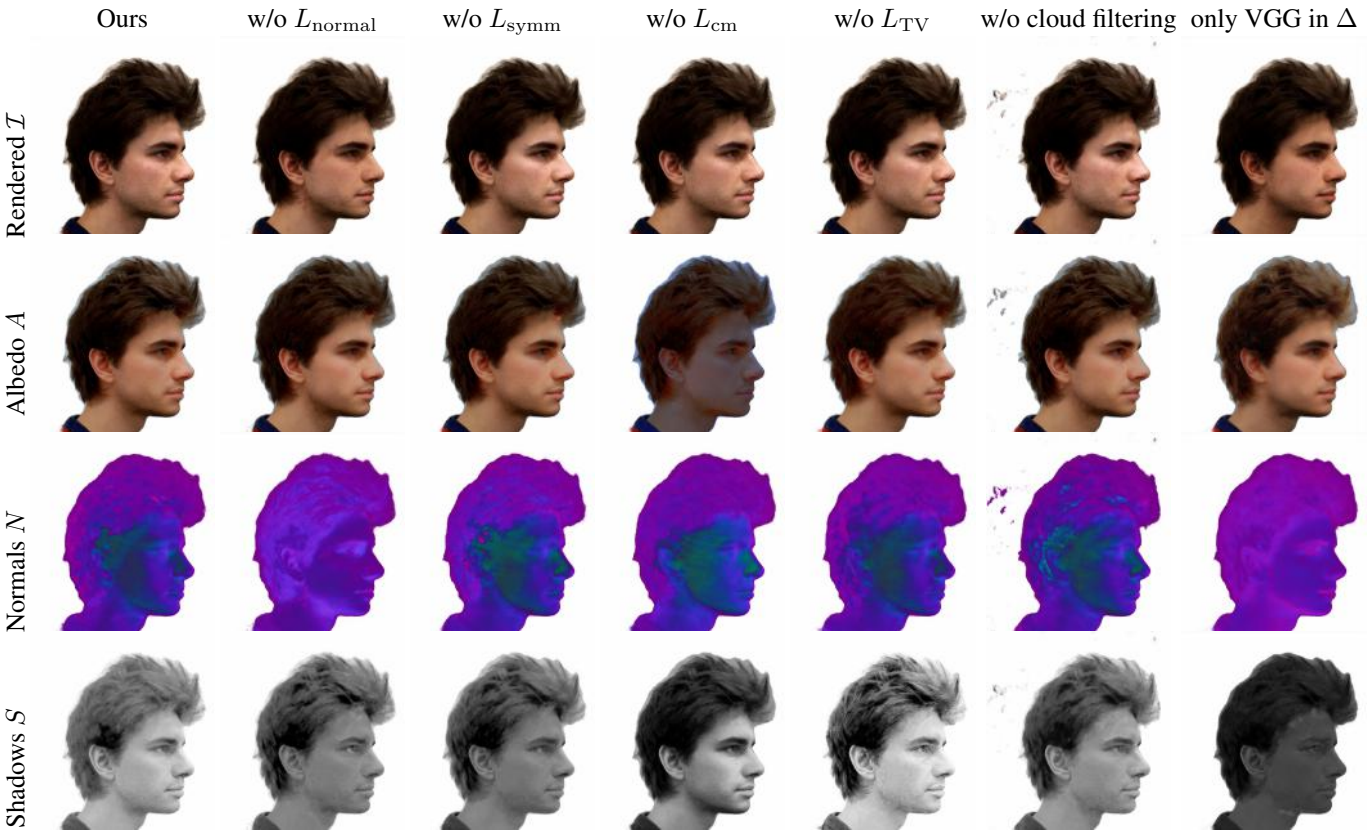

Figure 9: Visual comparison of ablations, created by either removing one of the loss terms, excluding the preprocessing stage of filtering the point cloud by learned segmentation masks, or leaving only VGG term in Δ mismatch, which is used to evaluate L_{final}, L_{normal}, L_{symm}. The viewpoint was not used for model fitting. *Electronic zoom-in recommended.*

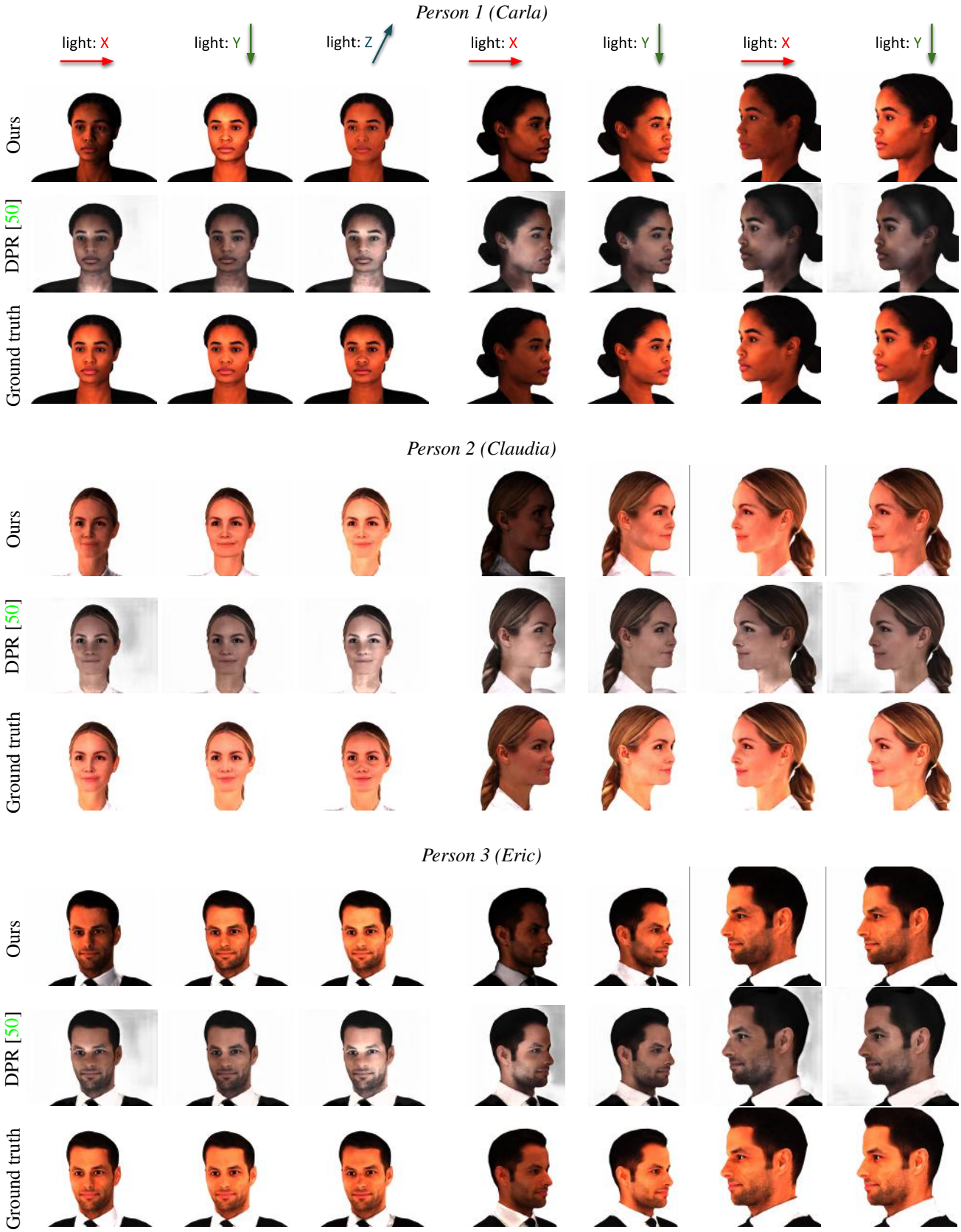

Figure 7: Qualitative comparison of the rendering synthetic models from RenderPeople under several viewpoints from validation and ambient+spherical harmonics lighting (see caption of Table 1 for more detail). The light directions are given in the **world space** and are aligned with the coordinate axes associated with the frontalized head.

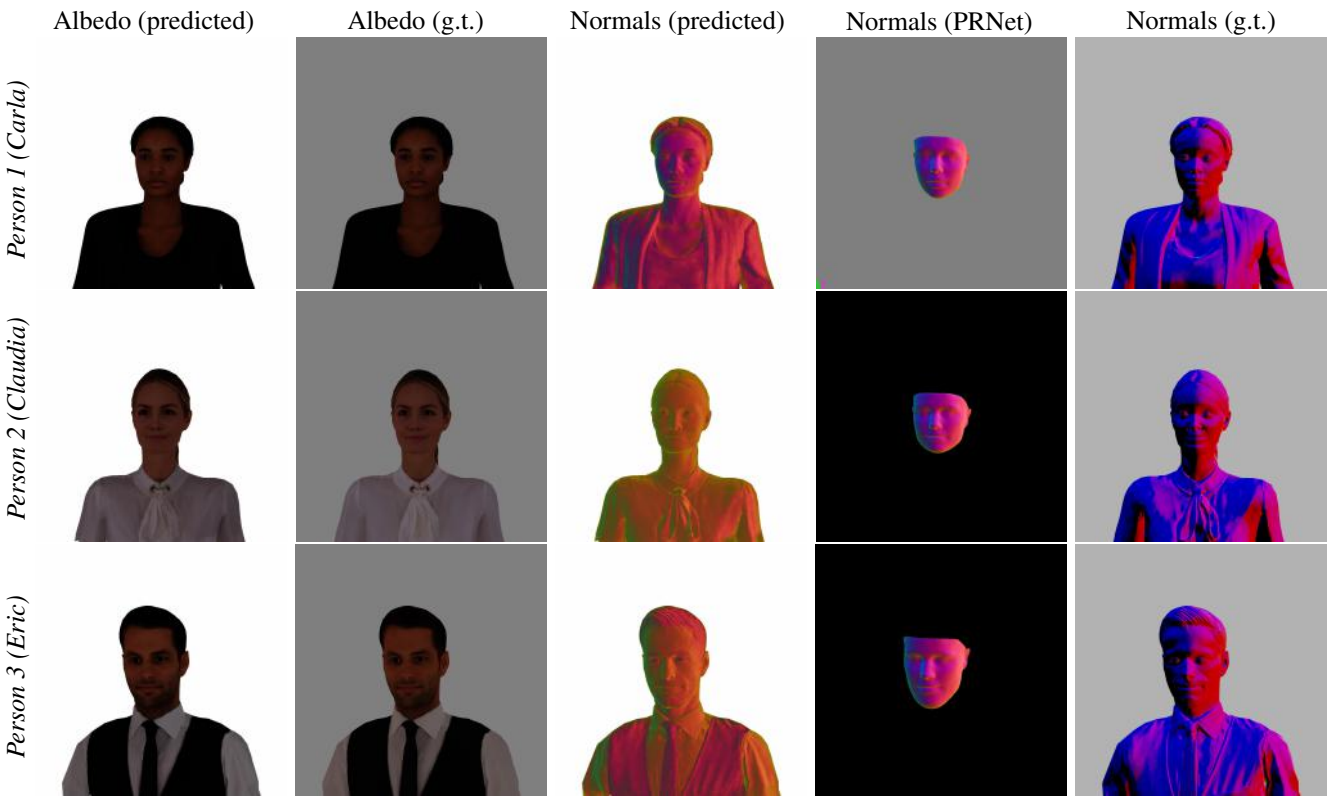

Figure 8: The comparison of predicted albedo rendering A and normal rendering N with the ground truth *(g.t.)* albedo and normals, respectively, rendered from the textured meshes in Blender. Additionally, we include the normals for the facial region obtained from face meshes predicted by PRNet [8]. Since normals from PRNet are the only source of direct normals supervision, their difference from the ground truth ones partially explains the systematic difference between the predicted and ground truth normals. The network is capable of extrapolating the learned normals from the face region to the head and upper body.

References

[1] Agisoft. *Metashape software*. 4, 6

[2] S. Agrawal, A. Pahuja, and S. Lucey. High accuracy face geometry capture using a smartphone video. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 81–90, 2020. 2

[3] K.-A. Aliev, A. Sevastopolsky, M. Kolos, D. Ulyanov, and V. Lempitsky. Neural point-based graphics. *European Conference on Computer Vision (ECCV)*, 2020. 2, 4, 10

[4] M. Boss, V. Jampani, K. Kim, H. Lensch, and J. Kautz. Two-shot spatially-varying brdf and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3982–3991, 2020. 3

[5] X. Cao, M. Waechter, B. Shi, Y. Gao, B. Zheng, and Y. Matsushita. Stereoscopic flash and no-flash photography for shape and albedo recovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3430–3439, 2020. 3

[6] Z. Chen, A. Chen, G. Zhang, C. Wang, Y. Ji, K. N. Kutulakos, and J. Yu. A neural rendering framework for free-viewpoint relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5599–5610, 2020. 2, 3

[7] Y. Dong, G. Chen, P. Peers, J. Zhang, and X. Tong. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics (TOG)*, 33(6):1–12, 2014. 3

[8] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 534–551, 2018. 5, 6, 7, 8, 13

[9] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *T-PAMI*, 32(8):1362–1376, 2009. 2

[10] V. Gkitsas, N. Zioulis, F. Alvarez, D. Zarpalas, and P. Daras. Deep lighting environment map estimation from spherical panoramas. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 640–641, 2020. 5

[11] K. Guo, P. Lincoln, P. Davidson, J. Busch, X. Yu, M. Whalen, G. Harvey, S. Orts-Escolano, R. Pandey, J. Dourgarian, et al. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Transactions on Graphics (TOG)*, 38(6):1–19, 2019. 2, 3

[12] M. Harman and et al. Open Camera application. https://sourceforge.net/p/opencamera/code/ci/master/tree/, 2013–2020. 6

[13] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017. 8

[14] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 8

[15] J. T. Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986. 3

[16] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 7

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[18] N. Kong, P. V. Gehler, and M. J. Black. Intrinsic video. In *European Conference on Computer Vision*, pages 360–375. Springer, 2014. 2

[19] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000. 2

[20] C. Lassner. Fast differentiable raycasting for neural rendering using sphere-based representations. *arXiv preprint arXiv:2004.07484*, 2020. 2

[21] A. Lattas, S. Moschoglou, B. Gecer, S. Ploumpis, V. Triantafyllou, A. Ghosh, and S. Zafeiriou. Avatarme: Realistically renderable 3d facial reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 760–769, 2020. 3

[22] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on pattern analysis and machine intelligence*, 16(2):150–162, 1994. 4

[23] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2475–2484, 2020. 3

[24] A. Meka, C. Haene, R. Pandey, M. Zollhöfer, S. Fanello, G. Fyffe, A. Kowdle, X. Yu, J. Busch, J. Dourgarian, et al. Deep reflectance fields: high-quality facial reflectance field inference from color gradient illumination. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 3

[25] A. Meka, R. Pandey, C. Haene, S. Orts-Escolano, P. Barnum, P. Davidson, D. Erickson, Y. Zhang, J. Taylor, S. Bouaziz, C. Legendre, W.-C. Ma, R. Overbeck, T. Beeler, P. Debevec, S. Izadi, C. Theobalt, C. Rhemann, and S. Fanello. Deep relightable textures - volumetric performance capture with neural rendering. In *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia)*, volume 39, December 2020. 3

[26] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *arXiv preprint arXiv:2003.08934*, 2020. 2

[27] G. Palma, N. Desogus, P. Cignoni, and R. Scopigno. Surface light field from video acquired in uncontrolled settings. In *2013 Digital Heritage International Congress (DigitalHeritage)*, volume 1, pages 31–38. IEEE, 2013. 3

[28] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, 2020. 4

[29] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500, 2001. 5, 7

[30] RenderPeople. *RenderPeople: World's largest library of 3D people*. 7

[31] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[32] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[33] 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. 4

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 5

[35] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, 2004. 6

[36] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017. 5

[37] Supervisely. *Supervisely Person Dataset*. 4

[38] S. Taheri, A. C. Sankaranarayanan, and R. Chellappa. Joint albedo estimation and pose tracking from video. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1674–1689, 2012. 2

[39] J. Thies, M. Zollhöfer, and M. Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph.*, 38(4), July 2019. 2

[40] G. Vogiatzis, C. H. Esteban, P. H. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *T-PAMI*, 29(12):2241–2246, 2007. 2

[41] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020. 2

[42] S. Wu, C. Rupprecht, and A. Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2020. 6

[43] S. Xu, J. Yang, D. Chen, F. Wen, Y. Deng, Y. Jia, and X. Tong. Deep 3d portrait from a single image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[44] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019. 4

[45] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 6, 7

[46] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 8

[47] X. Zhang, S. Fanello, Y.-T. Tsai, T. Sun, T. Xue, R. Pandey, S. Orts-Escolano, P. Davidson, C. Rhemann, P. Debevec, et al. Neural light transport for relighting and view synthesis. *arXiv preprint arXiv:2008.03806*, 2020. 2, 3

[48] X. C. Zhang, Y.-T. Tsai, R. Pandey, X. Zhang, R. Ng, D. E. Jacobs, et al. Portrait shadow manipulation. *arXiv preprint arXiv:2005.08925*, 2020. 2, 3

[49] T. Zhi, C. Lassner, T. Tung, C. Stoll, S. G. Narasimhan, and M. Vo. Texmesh: Reconstructing detailed human texture and geometry from rgb-d video. In *European Conference on Computer Vision*, pages 492–509. Springer, 2020. 2

[50] H. Zhou, S. Hadap, K. Sunkavalli, and D. W. Jacobs. Deep single-image portrait relighting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7194–7202, 2019. 2, 3, 7, 9, 10, 12