

# Photorealistic Monocular 3D Reconstruction of Humans Wearing Clothing

Thiemo Alldieck

Mihai Zanfir

Cristian Sminchisescu

Google Research

{alldieck, mihai, sminchisescu}@google.com

## Abstract

We present PHORHUM, a novel, end-to-end trainable, deep neural network methodology for photorealistic 3D human reconstruction given just a monocular RGB image. Our pixel-aligned method estimates detailed 3D geometry and, for the first time, the unshaded surface color together with the scene illumination. Observing that 3D supervision alone is not sufficient for high fidelity color reconstruction, we introduce patch-based rendering losses that enable reliable color reconstruction on visible parts of the human, and detailed and plausible color estimation for the non-visible parts. Moreover, our method specifically addresses methodological and practical limitations of prior work in terms of representing geometry, albedo, and illumination effects, in an end-to-end model where factors can be effectively disentangled. In extensive experiments, we demonstrate the versatility and robustness of our approach. Our state-of-the-art results validate the method qualitatively and for different metrics, for both geometric and color reconstruction.

## 1. Introduction

We present PHORHUM, a method to photorealistically reconstruct the 3D geometry and appearance of a dressed person as photographed in a single RGB image. The produced 3D scan of the subject not only accurately resembles the visible body parts but also includes plausible geometry and appearance of the non-visible parts, see fig. 1. 3D scans of people wearing clothing have many use cases and demand is currently rising. Applications like immersive AR and VR, games, telepresence, virtual try-on, free-viewpoint photo-realistic visualization, or creative image editing would all benefit from accurate 3D people models. The classical way to obtain models of people is to automatically scan using multi-camera set-ups, manual creation by an artist, or a combination of both as often artists are employed to ‘clean up’ scanning artifacts. Such approaches are difficult to scale, hence we aim for alternative, automatic solutions that would be cheaper and easier to deploy.

Prior to us, many researchers have focused on the problem of human digitization from a single image [6, 16, 17, 19,



Figure 1. Given a single image, we reconstruct the full 3D geometry – including self-occluded (or unseen) regions – of the photographed person, together with albedo and shaded surface color. Our end-to-end trainable pipeline requires no image matting and reconstructs all outputs in a single step.

36, 37, 42]. While these methods sometimes produce astonishingly good results, they have several shortcomings. First, the techniques often produce appearance estimates where shading effects are baked-in, and some methods do not produce color information at all. This limits the usefulness of the resulting scans as they cannot be realistically placed into a virtual scene. Moreover, many methods rely on multi-step pipelines that first compute some intermediate representation, or perceptually refine the geometry using estimated normal maps. While the former is at the same time impractical (since compute and memory requirements grow), and potentially sub-optimal (as often the entire system cannot be trained end-to-end to remove bias), the latter may not be useful for certain applications where the true geometry is needed, as in the case of body measurements for virtual try-on or fitness assessment, among others. In most existing methods color is exclusively estimated as a secondary step. However, from a methodological point of view, we argue that geometry and surface color should be computed simultaneously, since shading is a strong cue for surface geometry [18] and cannot be disentangled.

Our PHORHUM model specifically aims to address the above-mentioned state of the art shortcomings, as summarised in table 1. In contrast to prior work, we present an end-to-end solution that predicts geometry and appear-

end-to-end trainable	signed distances	returns color	returns albedo	returns shading	true surface normals	no mask needed	
$\times$	$\checkmark$	$\times$	$\times$	$\checkmark$	$\times$		PIFu [36]
$\times$	$\times$	$\times$	$\times$	$\times$	$\checkmark$	$\checkmark$	PIFuHD [37]
$\times$	$\times$	$\times$	$\times$	$\times$	$\checkmark$	$\times$	Geo-PIFu [16]
$\times$	$\times$	$\checkmark$	$\times$	$\times$	$\times$	$\times$	Arch [19]
$\times$	$\times$	$\checkmark$	$\times$	$\times$	$\times$	$\times$	Arch++ [17]
$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	<b>PHORHUM (ours)</b>

Table 1. Overview of the properties of single image 3D human reconstruction methods. Our method is the only one predicting albedo surface color and shading. Further, our method has the most practical training set-up, does not require image matting at test-time, and returns signed distances rather than binary occupancy – a more informative representation.

ance as a result of processing in a single composite network, with inter-dependent parameters, which are jointly estimated during a deep learning process. The appearance is modeled as albedo surface color without scene specific illumination effects. Furthermore, our system also estimates the scene illumination which makes it possible, in principle, to disentangle shading and surface color. The predicted scene illumination can be used to re-shade the estimated scans, to realistically place another person in an existing scene, or to realistically composite them into a photograph. Finally, we found that supervising the reconstruction using only sparse 3D information leads to perceptually unsatisfactory results. To this end, we introduce rendering losses that increase the perceptual quality of the predicted appearance. Our contributions can be summarised as follows:

- We present an end-to-end trainable system for high quality human digitization
- Our method computes, for the first time, albedo and shading information
- Our rendering losses significantly improve the visual fidelity of the results
- Our results are more accurate and feature more detail than current state-of-the-art

## 2. Related Work

Reconstructing the 3D shape of a human from a single image or a monocular video is a wide field of research. Often 3D shape is a byproduct of 3D human pose reconstruction and is represented through parameters of a statistical human body model [26, 44]. In this review, we focus on methods that go beyond and reconstruct the 3D human shape as well as garments or hairstyle. Early pioneering work is optimization-based. Those methods use videos of moving subjects and integrate information over time in order to reconstruct the complete 3D shape [5, 9]. The advent of deep

learning questioned the need for video. First, hybrid reconstruction methods based on a small number of images have been presented [4, 8]. Shortly after, approaches emerged to predict 3D human geometry from a single image. Those methods can be categorized by the used shape representation: voxel-based techniques [20, 42, 50] predict whether a given segment in space is occupied by the 3D shape. A common limitation is the high memory requirement resulting in shape estimates of limited spatial resolution. To this end, researchers quickly adopted alternative representations including visual hulls [31], moulded front and back depth maps [14, 39], or augmented template meshes [6]. Another class of popular representations consists of implicit function networks (IFNs). IFNs are functions over points in space and return either whether a point is inside or outside the predicted shape [11, 27] or return its distance to the closest surface [32]. Recently IFNs have been used for various 3D human reconstruction tasks [12, 13, 15, 30] and to build implicit statistical human body models [7, 28]. Neural radiance fields [29] are a related class of representations specialized for image synthesis that have also been used to model humans [25, 33, 43]. Saito *et al.* were the first to use IFNs for monocular 3D human reconstruction. They proposed an implicit function conditioned on pixel-aligned features [36, 37]. Other researchers quickly adopted this methodology for various use-cases [16, 24, 45, 49]. ARCH [19] and ARCH++ [17] also use pixel-aligned features but transform information into a canonical space of a statistical body model. This process results in animatable reconstructions, which comes, however, at the cost of artifacts that we will show. In this work, we also employ pixel-aligned features but go beyond the mentioned methods in terms of reconstructed surface properties (albedo and shading) and in terms of the quality of the 3D geometry. Also related is H3D-Net [35], a method for 3D head reconstruction, which uses similar rendering losses as we do, but requires three images and test-time optimization. In contrast, we work with a monocular image, purely feed-forward.

## 3. Method

Our goal is to estimate the 3D geometry  $\mathcal{S}$  of a subject as observed in a single image  $\mathbf{I}$ . Further, we estimate the unshaded albedo surface color and a per-image lighting model.  $\mathcal{S}$  is defined as the zero-level-set of the signed distance function (SDF)  $f$  represented using a neural network,

$$\mathcal{S}_{\theta}(\mathbf{I}) = \left\{ \mathbf{x} \in \mathbb{R}^3 \mid f(g(\mathbf{I}, \mathbf{x}; \theta), \gamma(\mathbf{x}); \theta) = (0, a) \right\} \quad (1)$$

where  $\theta$  is the superset of all learnable parameters. The surface  $\mathcal{S}$  is parameterized by pixel aligned features  $\mathbf{z}$  (*cf.* [36]) computed from the input image  $\mathbf{I}$  using the feature extractor network  $G$

$$g(\mathbf{I}, \mathbf{x}; \theta) = b(G(\mathbf{I}; \theta), \pi(\mathbf{x})) = \mathbf{z}_{\mathbf{x}}, \quad (2)$$

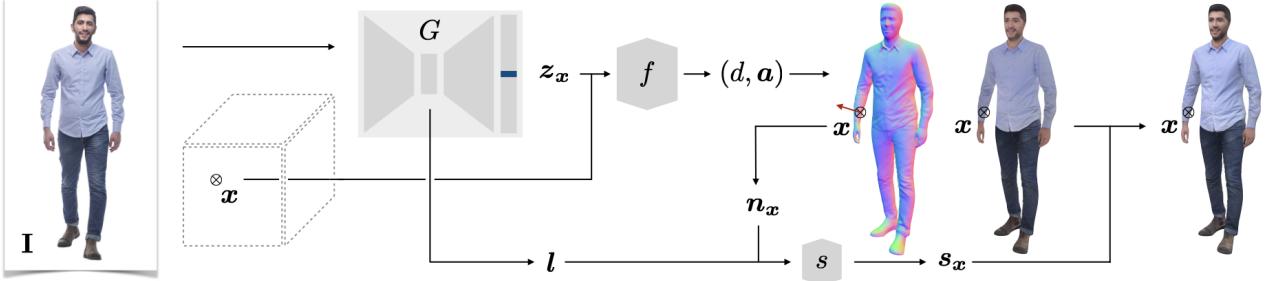


Figure 2. Overview of our method. The feature extractor network  $G$  produces pixel-aligned features  $z_x$  from an input image  $\mathbf{I}$  for all points in space  $\mathbf{x}$ . The implicit signed distance function network  $f$  computes the distance  $d$  to the closest surface given a point and its feature. Additionally  $f$  returns albedo colors  $\mathbf{a}$  defined for surface points. The shading network  $s$  predicts the shading for surface points given its surface normal  $\mathbf{n}_x$ , as well as illumination  $\mathbf{l}$ . On the right we show the reconstruction of geometry and albedo colors, and the shaded 3D geometry.

where  $b$  defines pixel access with bilinear interpolation and  $\pi(\mathbf{x})$  defines the pixel location of the point  $\mathbf{x}$  projected using camera  $\pi$ .  $f$  returns the signed distance  $d$  of the point  $\mathbf{x}$  w.r.t.  $\mathcal{S}$  and additionally its albedo color  $\mathbf{a}$

$$f(z_x, \gamma(\mathbf{x}); \theta) = (d, \mathbf{a}), \quad (3)$$

where  $\gamma$  denotes basic positional encoding as defined in [40]. In the sequel, we will use  $d_x$  for the estimated distance at  $\mathbf{x}$  and  $a_x$  for the color component, respectively.

To teach the model to decouple shading and surface color, we additionally estimate the surface shading using a per-point surface shading network

$$s(\mathbf{n}_x, \mathbf{l}; \theta) = s_x, \quad (4)$$

where  $\mathbf{n}_x = \nabla_{\mathbf{x}} d_x$  is the estimated surface normal defined by the gradient of the estimated distance w.r.t.  $\mathbf{x}$ .  $\mathbf{l}(\mathbf{I}; \theta) = \mathbf{l}$  is the illumination model estimated from the image. In practice, we use the bottleneck of  $G$  for  $\mathbf{l}$  and further reduce its dimensionality. The final shaded color is then  $\mathbf{c} = \mathbf{s} \circ \mathbf{a}$  with  $\circ$  denoting element-wise multiplication. We now define the losses we use to train  $f$ ,  $G$ , and  $s$ .

### 3.1. Losses

We create training examples by rendering scans of humans and drawing samples from the raw meshes – please see §3.2 for details. We define losses based on sparse 3D supervision and losses informed by ray-traced image patches.

**Geometry and Color Losses.** Given a ground truth mesh  $\mathcal{M}$  describing the surface  $\mathcal{S}$  as observed in an image  $\mathbf{I}$  and weights  $\lambda_*$  we define losses as follows. The surface is supervised via samples  $\mathcal{O}$  taken from the mesh surface  $\mathcal{M}$  and enforcing their distance to return zero and the distance gradient to follow their corresponding ground truth surface normal  $\bar{\mathbf{n}}$

$$\mathcal{L}_g = \frac{1}{|\mathcal{O}|} \sum_{i \in \mathcal{O}} \lambda_{g1} |d_{\mathbf{x}_i}| + \lambda_{g2} \|\mathbf{n}_{\mathbf{x}_i} - \bar{\mathbf{n}}_i\|. \quad (5)$$

Moreover, we supervise the sign of additional samples  $\mathcal{F}$  taken around the surface

$$\mathcal{L}_l = \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} \text{BCE}(l_i, \phi(k d_{\mathbf{x}_i})), \quad (6)$$

where  $l$  are inside/outside labels,  $\phi$  is the sigmoid function, and BCE is the binary cross-entropy.  $k$  determines the sharpness of the decision boundary and is learnable. Following [15], we apply geometric regularization such that  $f$  approximates a SDF with gradient norm 1 everywhere

$$\mathcal{L}_e = \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} (\|\mathbf{n}_{\mathbf{x}_i}\| - 1)^2. \quad (7)$$

Finally, we supervise the albedo color with the ‘ground truth’ albedo  $\bar{\mathbf{a}}$  calculated from the mesh texture

$$\mathcal{L}_a = \lambda_{a1} \frac{1}{|\mathcal{O}|} \sum_{i \in \mathcal{O}} |\mathbf{a}_{\mathbf{x}_i} - \bar{\mathbf{a}}_i| + \lambda_{a2} \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} |\mathbf{a}_{\mathbf{x}_i} - \bar{\mathbf{a}}_i|. \quad (8)$$

Following [36], we apply  $\mathcal{L}_a$  not only on but also near the surface. Since albedo is only defined on the surface, we approximate the albedo for points near the surface with the albedo of their nearest neighbor on the surface.

**Rendering losses.** The defined losses are sufficient to train our networks. However, as we show in the sequel, 2D rendering losses help further constrain the problem and increase the visual fidelity of the results. To this end, during training, we render random image patches of the surface  $\mathcal{S}$  with random strides and fixed size using ray-tracing. First, we compute the rays  $\mathcal{R}$  corresponding to a patch as defined by  $\pi$ . We then trace the surface using two strategies. First, to determine if we can locate a surface along a ray, we query  $f$  in equal distances along every ray  $\mathbf{r}$  and compute the sign of the minimum distance value

$$\sigma_{\mathbf{r}} = \phi(k \min_{t \geq 0} d_{\mathbf{o} + t\mathbf{r}}), \quad (9)$$

where  $\mathbf{o}$  is the camera location. We then take the subset  $\mathcal{R}_{\mathcal{S}} \subset \mathcal{R}$  of the rays containing rays where  $\sigma \leq 0.5$  and

$l = 0$ , i.e. we select the rays which located a surface where a surface is expected. Hereby, the inside/outside labels  $l$  are computed from pixel values of the image segmentation mask  $\mathbf{M}$  corresponding to the rays. For the subset  $\mathcal{R}_S$ , we exactly locate the surface using sphere tracing. Following [46], we make the intersection point  $\hat{\mathbf{x}}$  at iteration  $t$  differentiable w.r.t. to the network parameters without having to store the gradients of sphere tracing

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^t - \frac{\mathbf{r}}{\mathbf{n}^t \cdot \mathbf{r}} d_{\hat{\mathbf{x}}^t}. \quad (10)$$

In practice, we trace the surface both from the camera into the scene and from infinity back to the camera. This means, we locate both the front surface and the back surface. We denote the intersection points  $\hat{\mathbf{x}}^f$  for the front side and  $\hat{\mathbf{x}}^b$  for the back side, respectively. Using the above defined ray set  $\mathcal{R}_S$  and intersection points  $\hat{\mathbf{x}}$ , we enforce correct surface colors through

$$\mathcal{L}_r = \frac{1}{|\mathcal{R}_S|} \sum_{i \in \mathcal{R}_S} |\mathbf{a}_{\hat{\mathbf{x}}_i^f} - \bar{\mathbf{a}}_i^f| + |\mathbf{a}_{\hat{\mathbf{x}}_i^b} - \bar{\mathbf{a}}_i^b|, \quad (11)$$

where ground truth albedo colors  $\bar{\mathbf{a}}$  are taken from synthesized unshaded images  $\mathbf{A}^f$  and  $\mathbf{A}^b$ . The back image  $\mathbf{A}^b$  depicts the backside of the subject and is created by inverting the Z-buffer during rendering. We explain this process in more detail in §3.2. Additionally, we also define a VGG-loss [10]  $\mathcal{L}_{\text{VGG}}$  over the rendered front and back surface patches, enforcing that structure is similar to the unshaded ground-truth images. Finally, we supervise the shading using

$$\mathcal{L}_c = \frac{1}{|\mathcal{R}_S|} \sum_{i \in \mathcal{R}_S} |\mathbf{a}_{\hat{\mathbf{x}}_i^f} \circ \mathbf{s}_{\hat{\mathbf{x}}_i} - \mathbf{p}_i|, \quad (12)$$

with  $\mathbf{p}$  being the pixel color in the image  $\mathbf{I}$  corresponding to the ray  $\mathbf{r}$ . We found it also useful to supervise the shading on all pixels of the image  $\mathcal{I} = \{\mathbf{p}_0, \dots, \mathbf{p}_N\}$  using ground truth normals  $\bar{\mathbf{n}}$  and albedo  $\bar{\mathbf{a}}$

$$\mathcal{L}_s = \frac{1}{N} \sum_{i \in \mathcal{I}} |\bar{\mathbf{a}}_i^f \circ \mathbf{s}(\bar{\mathbf{n}}_i, \mathbf{l}; \theta) - \mathbf{p}_i|. \quad (13)$$

The final loss is a weighted combination of all previously defined losses  $\mathcal{L}_*$ . In §4.3, we ablate the usage of the rendering losses and the shading estimation network.

### 3.2. Dataset

We train our networks using pairs of meshes and rendered images. The meshes are scans of real people from commercial websites [3] and our own captured data. We employ high dynamic range images (HDRI) [2] for realistic image-based lighting and as backgrounds. Additionally to the shaded images, we also produce an alpha mask and unshaded albedo images. In the absence of the true surface albedo, we use the textures from the scans. Those are



Figure 3. A sample from our dataset. From left to right: rendered, shaded image on HDRI background; front and back albedo images; normal and an alpha map, and 3D mesh used for sampling.

uniformly lit but may contain small and local shading effects, e.g. from small wrinkles. As mentioned earlier, we produce not only a front side albedo image, but also one showing the back side. We obtain this image by inverting the Z-buffer during rendering. This means, not that the first visible point along each camera ray is visible, but the last passed surface point. See fig. 3 for an example of our training images. Furthermore, we produce normal maps used for evaluation and to supervise shading. Finally, we take samples by computing 3D points on and near the mesh surface and additionally sample uniformly in the bounding box of the whole dataset. For on-surface samples, we compute their corresponding albedo colors and surface normals, and for near and uniform samples we compute inside/outside labels by casting randomized rays and checking for parity.

We use 217 scans of people in different standing poses, wearing various outfits, and sometimes carrying bags or holding small objects. The scans sources allow for different augmentations: we augment the outfit colors for 100 scans and repose 38 scans. In total we produce a dataset containing  $\approx 190K$  images, where each image depicts a scan rendered with a randomly selected HDRI backdrop and with randomized scan placement. Across the 217 scans some share the same identity. We strictly split test and train identities and create a test-set containing 20 subjects, each rendered under 5 different light conditions.

### 3.3. Implementation Details

We now present our implementation and training procedure. Our networks are trained with images of  $512 \times 512$ px resolution. During training we render  $32 \times 32$ px patches with stride ranging from zero to three. We discard patches that only include background. Per training example we draw random samples for supervision from the surface and the space region around it. Concretely, we draw each 512 samples from the surface, near the surface and uniformly distributed over the surrounding space. The samples are projected onto the feature map using a projective camera with fixed focal length.

The feature extractor  $G$  is a U-Net with 13 encoder-decoder layers and skip connections. The first layer contains 64 filters and the filter size is doubled in the encoder

in each layer up to 512 at the maximum. The decoder halves the filter size at the 11th layer, which effectively means that  $G$  produces features in  $\mathbb{R}^{256}$ . We use Leaky ReLU activations and blur-pooling [48] for the encoder and bilinear resizing for the decoder, respectively. The geometry network  $f$  is a MLP with eight 512-dimensional fully-connected layers with Swish activation [34], an output layer with Sigmoid activation for the color component, and a skip connection to the middle layer. The shading network  $s$  is conditioned on a 16-dimensional illumination code and consists of three 256-dimensional fully-connected layers with Swish activation and an output layer with ReLU activation. Our total pipeline is relatively small and has only 48.8M trainable parameters. We train all network components jointly, end-to-end, for 500k iterations using the Adam optimizer [21], with learning-rate of  $1 \times 10^{-4}$ , linearly decaying with a factor of 0.9 over 50k steps. Please refer to our supplementary material for a list of our loss weights  $\lambda_*$ .

## 4. Experiments

We present quantitative evaluation results and ablation studies for geometric and color reconstruction on our own dataset. We also show qualitative results for real images.

**Inference.** At inference time, we take as input an RGB image of a person in a scene. Note that we do not require the foreground-background mask of the person. However, in practice we use a bounding box person detector to center the person and crop the image – a step that can also be performed manually. We use Marching Cubes [23] to generate our reconstructions by querying points in a 3D bounding box at a maximum resolution of  $512^3$ . We first approximate the bounding box of the surface by probing at coarse resolution and use Octree sampling to progressively increase the resolution as we get closer to the surface. This allows for very detailed reconstructions of the surface geometry with a small computational overhead, being made possible by the use of signed distance functions in our formulation.

**Camera Model.** Different from other methods in the literature, we deviate from the standard orthographic camera model and instead use perspective projection, due to its general validity. A model assuming an orthographic camera would in practice produce incorrect 3D geometry. In fig. 5 one can see the common types of errors for such models. The reconstructed heads are unnaturally large, as they extend in depth away from the camera. In contrast, our reconstructions are more natural, with correct proportions between the head and the rest of the body.

**Competing Methods.** We compare against other single-view 3D reconstructions methods that leverage pixel-aligned image features. PIFu [36] is the pioneering work and learns an occupancy field. PIFuHD [37], a very parameter-heavy model, builds upon PIFu with higher res-

Front side	Back side	Mean	
2.68	2.15	2.42	PIFu [36]
2.51	2.04	2.28	ARCH [19]
2.68	2.26	2.47	ARCH++ [17]
<b>2.88</b>	<b>2.43</b>	<b>2.65</b>	<b>PHORHUM (Ours)</b>

Table 2. Inception Score of renderings of the front and back side of the 3D reconstructions. Our method produces the most natural surface colors for both the front and the unseen back.

olution inputs and leverages a multi-level architecture for coarse and fine grained reconstruction. It also uses off-line estimated front and back normal maps as additional input. GeoPIFu [16] is also a multi-level architecture, but utilizes latent voxel features as a coarse human shape proxy. ARCH [19] and ARCH++ [17] transform information into the canonical space of a statistical body model. This sacrifices some of the reconstruction quality for the ability to produce animation-ready avatars. For PIFu, ARCH, ARCH++, an off-the-shelf detector [22] is used to segment the person in the image, whereas PHORHUM (us) and PIFuHD use the raw image. The results of ARCH and ARCH++ have been kindly provided by the authors.

Due to the lack of a standard dataset and the non-availability of training scripts of most methods, all methods have been trained with similar but different datasets. All datasets are sufficiently large to enable generalization across various outfits, body shapes, and poses. Please note that our dataset is by far the smallest with only 217 scans. All other methods use  $> 400$  scans.

### 4.1. Reconstruction Accuracy

To evaluate the geometric reconstruction quality, we report several metrics, namely: bi-directional Chamfer distance (Ch.  $\downarrow$ ), Normal Consistency (NC  $\uparrow$ ), and Volumetric Intersection over Union (IoU  $\uparrow$ ). To account for the inherent ambiguity of monocular reconstruction w.r.t. scale, we first use Iterative Closest Point to align the reconstructions with the ground truth shapes. Additionally, we evaluate how well the visible part of the person is reconstructed. This also mitigates effects caused by camera model assumptions. We render the reconstruction under the assumed camera model and compare with the original image, the unshaded albedo image, and the rendered normals. For image reconstruction metrics, we use peak signal-to-noise ratio (PSNR  $\uparrow$ ), structural similarity index (SSIM  $\uparrow$ ) and learned perceptual image patch similarity (LPIPS  $\downarrow$ ). Finally, we use the Inception Score (IS  $\uparrow$ ) [38] as a perceptual metric. This allows us to also evaluate non-visible parts where no ground truth is available, as in the case of the shaded backside view of a person.

We report the introduced metrics in tables 2 and 3. Our model produces the most natural surface colors for both the visible front side and the non-visible back side. Further-

3D Metrics			Rendered Normals			Shaded Rendering			Albedo Rendering			
Ch. ↓	IoU ↑	NC ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	
3.21	0.61	0.77	0.71	0.17	17.69	0.83	0.16	24.57	—	—	—	PIFu [36]
4.54	0.62	0.78	0.78	0.10	20.15	—	—	—	—	—	—	PIFuHD [37]
4.98	0.54	0.72	0.68	0.18	17.25	—	—	—	—	—	—	Geo-PIFu [16]
3.58	0.57	0.75	0.68	0.20	15.51	0.72	0.23	19.28	—	—	—	ARCH [19]
3.48	0.59	0.77	0.70	0.17	16.24	0.83	0.17	22.69	—	—	—	ARCH++ [17]
1.14	0.73	0.84	0.77	0.12	19.25	—	—	—	0.82	0.16	20.51	Ours w/o rendering
1.29	0.73	0.85	0.78	0.11	19.67	—	—	—	0.85	0.13	22.02	Ours w/o shading
1.29	0.73	0.85	0.78	0.11	19.60	0.85	0.13	24.01	0.85	0.12	22.23	PHORHUM (Ours)

Table 3. Numerical comparisons with other single-view 3D reconstructions methods and ablations of our method. We mark the **best** and second best results. All Chamfer metrics are  $\times 10^{-3}$ .

more, our method produces the most accurate 3D reconstructions and is the only one that computes the surface albedo. Our results are on-par with those of PIFuHD in terms of surface normal reconstruction. In contrast to our method, PIFuHD specifically targets surface normals with a dedicated image-translation network. ARCH and ARCH++ also specifically handle surface normals, but in contrast to all other methods, only compute a normal map and do not refine the true geometry. Note that we use normal mapping (not true surface normals) for ARCH and ARCH++ in the comparison and in all the following figures. For shaded rendering of the front side, the original PIFu is numerically on par with our method. However, the results are blurry, which is evident in the lower Inception Score and LPIPS. PIFu and all other competing methods do not decompose albedo and shading, which means that they can simply project the original image onto the reconstruction. Although our method performs a harder task, our results are among the best, or the best, across all metrics.

## 4.2. Qualitative Results

Quantitative evaluations do not always correlate well with human perception. To this end, we show qualitative results of our method and results of PIFu, ARCH, and ARCH++ on real images in fig. 4, and a side-by-side comparison with PIFuHD in fig. 5.

In fig. 4, we show the 3D reconstructions with color-mapped normals, and the colored reconstructions, both front and back. For our method we render the albedo and additionally show the shaded reconstruction in the last column. Our method reliably reconstructs facial detail, hair, and clothing wrinkles. The albedo features small color patterns visible in the input image and, at the same time, does not contain strong shading effects. The reconstructed non-visible back side is sharp, detailed, and matches our expectations well. The clothing items are well separated and small details like hair curls are present. ARCH and ARCH++ encounter problems reconstructing the red dress in line two, sometimes produce artifacts, and fail entirely for the subject in line five. The observed problems are common for methods that reconstruct relative to, or in the canonical

space, of a body model. In contrast, our method produces complete, smooth, and detailed reconstructions.

PIFuHD does not compute surface color, thus we only compare the geometry in fig. 5. We show our shaded results only for completeness. Consistent with the numerical results, our results are on par in terms of level of detail. However, our reconstructions are smoother and contain less noise – a property of signed distance functions. Our model is capable of producing these results by using a rather small network capacity. In contrast PIFuHD is an extremely large model that is specifically tailored for surface normal estimation.

As mentioned before, our method is the only one that jointly estimates both albedo and shading. Albedo is a useful property in practice as it allows the usage of our reconstructions in virtual environments with their own light composition. Additionally, as a byproduct of our shading estimation, we can do image compositing [41, 47], one of the most common photo editing tasks. One example is given in fig. 7. We first computed the illumination  $l$  from a given target image. We then reconstruct two subjects from studio photographs and use  $l$  to re-shade them. This allows us to compose a synthesized group picture with matching illumination for all people in the scene.

## 4.3. Ablations

We now ablate two main design choices of our method: first, the rendering losses, and second, shading estimation. In tab. 3, we report metrics for our method trained without rendering losses (w/o rendering) and without shading estimation (w/o shading). Furthermore, in fig. 6 we show visual examples of results produced by our model variant trained without rendering losses.

While only using 3D sparse supervision produces accurate geometry, the albedo estimation quality is, however, significantly decreased. As evident in fig. 6 and also numerically in tab. 3, the estimated albedo contains unnatural color gradient effects. We hypothesize that due to the sparse supervision, where individual points are projected into the feature map, the feature extractor network does not learn to understand structural scene semantics. Here our patch-



Figure 4. Qualitative comparisons on real images with state-of-the-art methods that produce color. From left to right: Input image, PIFu, ARCH, ARCH++, PHORHUM (ours), our shaded reconstruction. For each method we show the 3D geometry and the reconstructed color. Our method produces by far the highest level of detail and the most realistic color estimate for the unseen back side.

based rendering losses help, as they provide gradients for neighboring pixels. Moreover, our rendering losses could better connect the zero-level-set of the signed distance function with the color field, as they supervise the color at the current zero-level-set and not at the expected surface location. We plan to structurally investigate these observations, and leave these for future work.

Estimating the shading jointly with the 3D surface and albedo does not impair the reconstruction accuracy. On the contrary, as evident in tab. 3, this helps improve albedo reconstruction. This is in line with our hypothesis that shad-

ing estimation helps the networks to better decouple shading effects from albedo. Finally, shading estimating makes our method a holistic reconstruction pipeline.

## 5. Discussion and Conclusions

**Limitations.** The limitations of our method are sometimes apparent when the clothing or pose of the person in the input image deviates too much from our dataset distribution, see fig. 8. Loose, oversized, and non-Western clothing items are not well covered by our training set. The backside



Figure 5. Qualitative comparisons on real images with the state-of-the-art method PIFuHD. We show front and back geometry produced by PIFuHD (left) and our results (right). Our reconstructions feature a similar level of detail but contain less noise and body poses are reconstructed more reliably. Additionally, our method is able to produce albedo and shaded surface color – we show our shaded reconstructions for reference.



Figure 6. Loss ablation: The usage of our rendering losses (right) significantly improves albedo estimation. Note the unnatural color gradients when using sparse 3D supervision only (left).



Figure 7. We can apply the estimated illumination from one image to another, which allows us to create the group picture (right) by inserting the reconstructions of the subjects (left) with matching shaded surface.

of the person sometimes does not semantically match the front side. A larger, more geographic and culturally diverse dataset would alleviate these problems, as our method does not make any assumptions about clothing style or pose.

**Application Use Cases and Model Diversity.** The construction of our model is motivated by the breadth of transformative, immersive 3D applications, that would become possible, including clothing virtual apparel try-on, immersive visualisation of photographs, personal AR and VR for improved communication, special effects, human-computer



Figure 8. Failure cases. Wide clothing is under-represented in our dataset and this can be addressed with more diverse training. Complex poses can lead to missing body parts. The back-side sometimes mismatches the front (subject is wearing a hood).

interaction or gaming, among others. Our models are trained with a diverse and fair distribution, and as the size of this set increases, we expect good practical performance.

**Conclusions.** We have presented a method to reconstruct the three-dimensional (3D) geometry of a human wearing clothing given a single photograph of that person. Our method is the first one to compute the 3D geometry, surface albedo, and shading, from a single image, jointly, as prediction of a model trained end-to-end. Our method works well for a wide variation of outfits and for diverse body shapes and skin tones, and reconstructions capture most of the detail present in the input image. We have shown that while sparse 3D supervision works well for constraining the geometry, rendering losses are essential in order to reconstruct perceptually accurate surface color. In the future, we would like to further explore weakly supervised differentiable rendering techniques, as they would support, long-term, the construction of larger and more inclusive models, based on diverse image datasets of people, where accurate 3D surface ground truth is unlikely to be available.

## Supplementary Material

In this supplementary material, we detail our implementation by listing the values of all hyper-parameters. Further, we report inference times, demonstrate how we can repose our reconstructions, conduct further comparisons, and show additional results.

### A. Implementation Details

In this section, we detail our used hyper-parameters and provide timings for mesh reconstruction via Marching Cubes [23].

#### A.1. Hyper-parameters

When training the network, we minimize a weighted combination of all defined losses:

$$\begin{aligned} \mathcal{L} = & \mathcal{L}_g + \lambda_e \mathcal{L}_e + \lambda_l \mathcal{L}_l + \mathcal{L}_a + \lambda_r \mathcal{L}_r \\ & + \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s + \lambda_{\text{VGG}} \mathcal{L}_{\text{VGG}}. \end{aligned} \quad (14)$$

Further, we have defined the weights  $\lambda_{g_1}$ ,  $\lambda_{g_2}$ ,  $\lambda_{a_1}$ , and  $\lambda_{a_2}$  inside the definitions of  $\mathcal{L}_g$  and  $\mathcal{L}_a$ . During all experiments, we have used the following empirically determined configuration:

$$\lambda_e = 0.1, \lambda_l = 0.2, \lambda_r = 1.0, \lambda_c = 1.0, \lambda_s = 50.0, \lambda_{\text{VGG}} = 1.0, \lambda_{g_2} = 1.0, \lambda_{a_1} = 0.5, \lambda_{a_2} = 0.3$$

Additionally we found it beneficial to linearly increase the surface loss weight  $\lambda_{g_1}$  from 1.0 to 15.0 over the duration of 100k interactions.

#### A.2. Inference timings

To create a mesh we run Marching Cubes over the distance field defined by  $f$ . We first approximate the bounding box of the surface by probing at coarse resolution and use Octree sampling to progressively increase the resolution as we get closer to the surface. This allows us to extract meshes with high resolution without large computational overhead. We query  $f$  in batches of  $64^3$  samples up to the desired resolution. The reconstruction of a mesh in a  $256^3$  grid takes on average 1.21s using a single NVIDIA Tesla V100. Reconstructing a very dense mesh in a  $512^3$  grid takes on average 5.72s. Hereby, a single batch of  $64^3$  samples takes 142.1ms. In both cases, we query the features once which takes 243ms. In practise, we also query  $f$  a second time for color at the computed vertex positions which takes 56.5ms for meshes in  $256^3$  and 223.3ms for  $512^3$ , respectively. Meshes computed in  $256^3$  and  $512^3$  grids contain about 100k and 400k vertices, respectively. Note that we can create meshes in arbitrary resolutions and our reconstructions can be rendered through sphere tracing without the need to generate an explicit mesh.

## B. Additional Results

In the sequel, we show additional results and comparisons. First, we demonstrate how we can automatically rig our reconstructions using a statistical body model. Then we conduct further comparisons on the PeopleSnapshot Dataset [1]. Finally, we show additional qualitative results.

### B.1. Animating Reconstructions

In fig. 9, we show examples of rigged and animated meshes created using our method. For rigging, we fit the statistical body model GHUM [44] to the meshes. To this end, we first triangulate joint detections produced by an off-the-shelf 2D human keypoint detector on renderings of the meshes. We then fit GHUM to the triangulated joints and the mesh surface using ICP. Finally, we transfer the joints and blend weights from GHUM to our meshes. We can now animate our reconstructions using Mocap data or by sampling GHUM’s latent pose space. By first reconstructing a static shape that we then rig in a secondary step, we avoid reconstruction errors of methods aiming for animation ready reconstruction in a single step [17, 19].

### B.2. Comparisons on the PeopleSnapshot Dataset

We use the public PeopleSnapshot dataset [1, 5] for further comparisons. The PeopleSnapshot dataset contains of people rotating in front of the camera while holding an A-pose. The dataset is openly available for research purposes. For this comparison we use only the first frame of each video. We compare once more with PIFuHD [37] and additionally compare with the model-based approach Tex2Shape [6]. Tex2Shape does not estimate the pose of the observed subject but only its shape. The shape is represented as displacements to the surface of the SMPL body model [26]. In fig. 10 we show the results of both methods side-by-side with our method. Also in this comparison our method produces the most realistic results and additionally also reconstructs the surface color.

### B.3. Qualitative Results

We show further qualitative results in fig. 11. Our methods performs well on a wide range of subjects, outfits, backgrounds, and illumination conditions. Further, despite never being trained on this type of data, our method performs extremely well on images of people with solid white background. In fig. 12 we show a number of examples. This essentially means, matting the image can be performed as a pre-processing step to boost the performance of our method in cases where the model has problems identifying foreground regions.



Figure 9. Examples of reconstructions rigged and animated in a post processing step. We show the input image (left) and re-posed reconstructions (right). The reconstructions are rendered under a novel illumination.

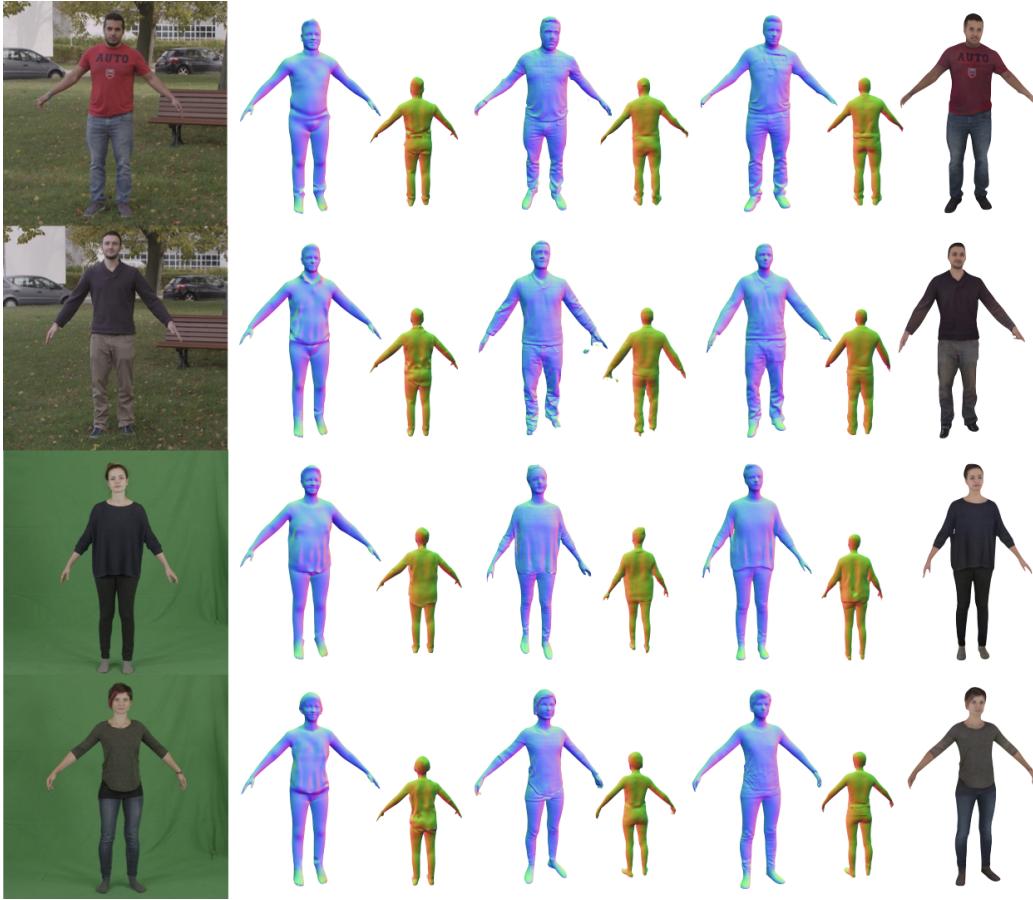


Figure 10. Qualitative comparison on the PeopleSnapshot dataset [1]. From left to right: Input image, geometry produced by Tex2Shape [6], PIFuHD [37], and PHORHUM (ours). We additionally show albedo reconstructions for our method.



Figure 11. Qualitative results on real images featuring various outfits, backgrounds, and illumination conditions. From left to right: Input image, 3D geometry (front and back), albedo reconstruction (front and back), and shaded surface.



Figure 12. Despite never being trained on matted images, our method performs extremely well on images with white background. From left to right: Input image, 3D geometry (front and back), albedo reconstruction (front and back), and shaded surface.

## References

- [1] <https://graphics.tu-bs.de/people-snapshot>. 9, 10
- [2] <https://polyhaven.com/>. 4
- [3] <https://renderpeople.com/>. 4
- [4] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single RGB camera. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1175–1186. IEEE, 2019. 2
- [5] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3D people models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8387–8397. IEEE, 2018. 2, 9
- [6] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2Shape: Detailed full human body geometry from a single image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2293–2303. IEEE, 2019. 1, 2, 9, 10
- [7] Thiemo Alldieck, Hongyi Xu, and Cristian Sminchisescu. imGHUM: Implicit generative models of 3D human shape and articulated pose. In *Int. Conf. Comput. Vis.*, 2021. 2
- [8] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *Int. Conf. Comput. Vis.* IEEE, oct 2019. 2
- [9] Federica Bogo, Michael J. Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *Int. Conf. Comput. Vis.*, pages 2300–2308, 2015. 2
- [10] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017. 4
- [11] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5939–5948, 2019. 2
- [12] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, jun 2020. 2
- [13] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Neural articulated shape approximation. In *Eur. Conf. Comput. Vis.* Springer, August 2020. 2
- [14] Valentin Gabeur, Jean-Sébastien Franco, Xavier Martin, Cordelia Schmid, and Gregory Rogez. Moulding humans: Non-parametric 3d human shape estimation from single images. In *Int. Conf. Comput. Vis.*, pages 2232–2241, 2019. 2
- [15] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Int. Conf. on Mach. Learn.*, pages 3569–3579. 2020. 2, 3
- [16] Tong He, John Collomosse, Hailin Jin, and Stefano Soatto. Geo-PIFu: Geometry and pixel aligned implicit functions for single-view human reconstruction. In *Adv. Neural Inform. Process. Syst.*, 2020. 1, 2, 5, 6
- [17] Tong He, Yuanlu Xu, Shunsuke Saito, Stefano Soatto, and Tony Tung. Arch++: Animation-ready clothed human reconstruction revisited. In *Int. Conf. Comput. Vis.*, pages 11046–11056, 2021. 1, 2, 5, 6, 9
- [18] Berthold KP Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. 1970. 1
- [19] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3093–3102, 2020. 1, 2, 5, 6, 9
- [20] Aaron S Jackson, Chris Manafas, and Georgios Tzimiropoulos. 3d human body reconstruction from a single image via volumetric regression. In *ECCV Workshops*, 2018. 2
- [21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *Int. Conf. Learn. Represent.*, 2015. 5
- [22] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9799–9808, 2020. 5
- [23] Thomas Lewiner, Hélio Lopes, Antônio Wilson Vieira, and Geovan Tavares. Efficient implementation of marching cubes’ cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003. 5, 9
- [24] Ruilong Li, Yuliang Xiu, Shunsuke Saito, Zeng Huang, Kyle Olszewski, and Hao Li. Monocular real-time volumetric performance capture. In *Eur. Conf. Comput. Vis.*, pages 49–67. Springer, 2020. 2
- [25] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Trans. Graph.*, 2021. 2
- [26] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015. 2, 9
- [27] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4460–4470, 2019. 2
- [28] Marko Mihajlovic, Yan Zhang, Michael J. Black, and Siyu Tang. LEAP: Learning articulated occupancy of people. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2
- [29] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Eur. Conf. Comput. Vis.*, 2020. 2
- [30] Armin Mustafa, Akin Caliskan, Lourdes Agapito, and Adrian Hilton. Multi-person implicit reconstruction from a single image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14474–14483, 2021. 2
- [31] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4480–4490, 2019. 2

- [32] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 165–174, 2019. 2
- [33] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 2
- [34] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 5
- [35] Eduard Ramon, Gil Triginer, Janna Escur, Albert Pumarola, Jaime Garcia, Xavier Giro-i Nieto, and Francesc Moreno-Noguer. H3D-Net: Few-shot high-fidelity 3d head reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5620–5629, 2021. 2
- [36] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morigima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Int. Conf. Comput. Vis.*, pages 2304–2314, 2019. 1, 2, 3, 5, 6
- [37] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1, 2, 5, 6, 9, 10
- [38] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29:2234–2242, 2016. 5
- [39] David Smith, Matthew Loper, Xiaochen Hu, Paris Mavroidis, and Javier Romero. Facsimile: Fast and accurate scans from an image in less than a second. In *Int. Conf. Comput. Vis.*, pages 5330–5339, 2019. 2
- [40] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Adv. Neural Inform. Process. Syst.*, volume 33, pages 7537–7547, 2020. 3
- [41] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3789–3797, 2017. 6
- [42] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *Eur. Conf. Comput. Vis.*, pages 20–36, 2018. 1, 2
- [43] Hongyi Xu, Thieno Alldieck, and Cristian Sminchisescu. H-NeRF: Neural radiance fields for rendering and temporal reconstruction of humans in motion. In *Adv. Neural Inform. Process. Syst.*, 2021. 2
- [44] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. GHUM & GHUML: Generative 3d human shape and articulated pose models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6184–6193, 2020. 2, 9
- [45] Ze Yang, Shenlong Wang, Sivabalan Manivasagam, Zeng Huang, Wei-Chiu Ma, Xinchen Yan, Ersin Yumer, and Raquel Urtasun. S3: Neural shape, skeleton, and skinning fields for 3d human modeling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13284–13293, 2021. 2
- [46] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33, 2020. 4
- [47] Mihai Zanfir, Elisabeta Oneata, Alin-Ionut Popa, Andrei Zanfir, and Cristian Sminchisescu. Human synthesis and scene compositing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12749–12756, 2020. 6
- [48] Richard Zhang. Making convolutional networks shift-invariant again. In *Int. Conf. on Mach. Learn.*, pages 7324–7334. PMLR, 2019. 5
- [49] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021. 2
- [50] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. DeepHuman: 3d human reconstruction from a single image. In *Int. Conf. Comput. Vis.*, October 2019. 2