This CVPR 2020 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

ARCH: Animatable Reconstruction of Clothed Humans

Zeng Huang[1,2]*, Yuanlu Xu[1], Christoph Lassner[1], Hao Li[2], Tony Tung[1]
[1]Facebook Reality Labs, Sausalito, USA [2]University of Southern California, USA
zenghuan@usc.edu, merayxu@gmail.com, classner@fb.com, hao@hao-li.com, tony.tung@fb.com

Abstract

In this paper, we propose ARCH (Animatable Reconstruction of Clothed Humans), a novel end-to-end framework for accurate reconstruction of animation-ready 3D clothed humans from a monocular image. Existing approaches to digitize 3D humans struggle to handle pose variations and recover details. Also, they do not produce models that are animation ready. In contrast, ARCH is a learned pose-aware model that produces detailed 3D rigged full-body human avatars from a single unconstrained RGB image. A Semantic Space and a Semantic Deformation Field are created using a parametric 3D body estimator. They allow the transformation of 2D/3D clothed humans into a canonical space, reducing ambiguities in geometry caused by pose variations and occlusions in training data. Detailed surface geometry and appearance are learned using an implicit function representation with spatial local features. Furthermore, we propose additional per-pixel supervision on the 3D reconstruction using opacity-aware differentiable rendering. Our experiments indicate that ARCH increases the fidelity of the reconstructed humans. We obtain more than 50% lower reconstruction errors for standard metrics compared to state-of-the-art methods on public datasets. We also show numerous qualitative examples of animated, high-quality reconstructed avatars unseen in the literature so far.

1. Introduction

3D human reconstruction has been explored for several decades in the field of computer vision and computer graphics. Accurate methods based on stereo or fusion have been proposed using various types of sensors [12, 42, 31, 33, 38, 49, 50], and several applications have become popular in sports, medicine and entertainment (*e.g.*, movies, games, AR/VR experiences). However, these setups require tightly controlled environments. To date, full 3D human reconstruction with detailed geometry and appearance from in-the-wild pictures is still challenging (*i.e.*, taken in natural conditions as opposed to laboratory environments).

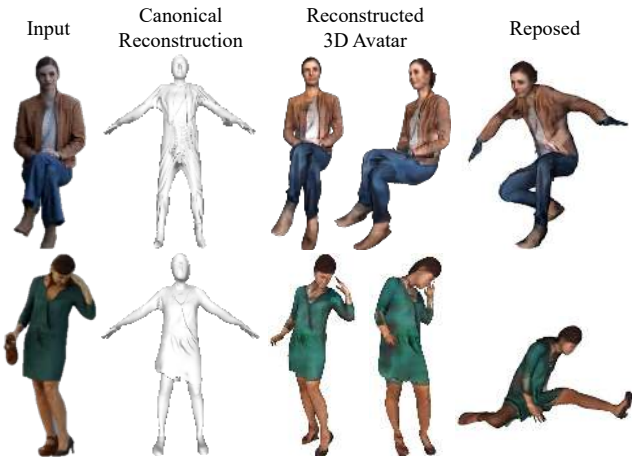


Figure 1. Given an image of a subject in arbitrary pose (left), ARCH creates an accurate and animatable avatar with detailed clothing (center). As rigging and albedo are estimated, the avatar can be reposed and relit in new environments (right).

Moreover, the lack of automatic rigging prevents animation-based applications.

Recent computer vision models have enabled the recovery of 2D and 3D human pose and shape estimation from a single image. However, they usually rely on representations that have limitations: (1) skeletons [11] are kinematic structures that are accurate to represent 3D poses, but do not carry body shape information. (2) surface meshes [18, 35, 51] can represent body shape geometry, but have topology constraints; (3) voxels [44] are topology-free, but memory costly with limited resolution, and need to be rigged for animation. In this paper, we propose the ARCH (Animatable Reconstruction of Clothed Humans) framework that possesses all benefits of current representations. In particular, we introduce a learned model that has human body structure knowledge (i.e., body part semantics), and is trained with humans in arbitrary poses.

First, 3D body pose and shape estimation can be inferred from a single image of a human in arbitrary pose by a prediction model [51]. This initialization step is used for normalized-pose reconstruction of clothed human shape within a canonical space. This allows us to define a Semantic Space (SemS) and a Semantic Deformation Field (SemDF) by densely sampling 3D points around the clothed

*Work performed at Facebook Reality Labs.

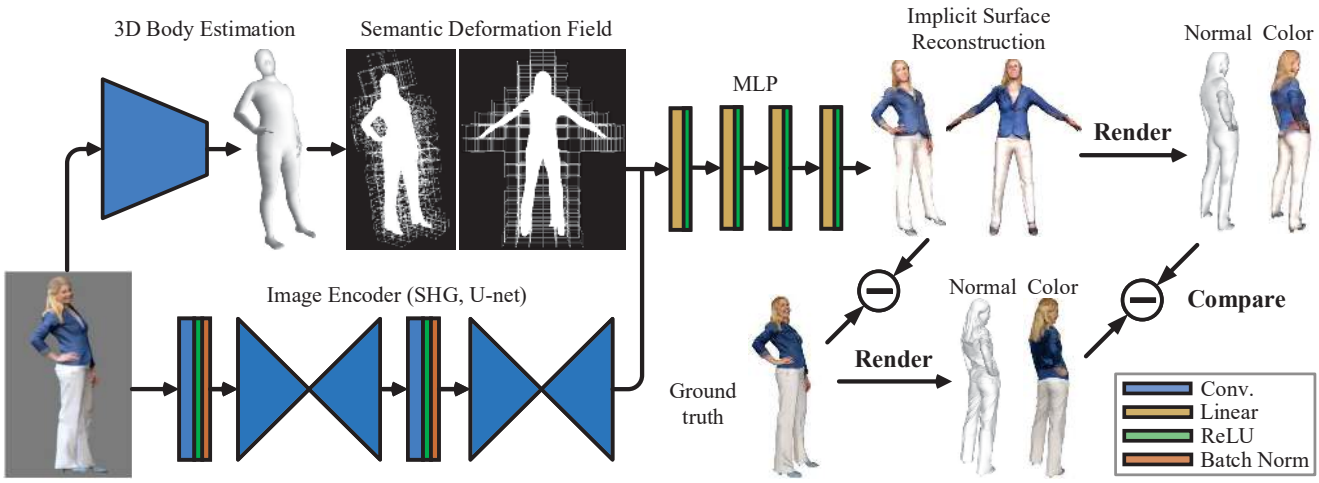


Figure 2. *ARCH overview.* The framework contains three components: i) estimation of correspondences between an input image space and the canonical space, ii) implicit surface reconstruction in the canonical space from surface occupancy, normal and color estimation, iii) refinement of normal and color through differentiable rendering.

body surface and assigning skinning weights. We then learn an implicit function representation of the 3D occupancy in the canonical space based on SemS and SemDF, which enables the reconstruction of high-frequency details of the surface (including clothing wrinkles, hair style, etc.) superior to the state of the art [32, 40, 44]. The surface representing a clothed human in a neutral pose is implicitly rigged in order to be used as an animatable avatar. Moreover, a differentiable renderer is used to refine normal and color information for each 3D point in space by Granular Render-and-Compare. Here, we regard them as a sphere and develop a new blending formulation based on the estimated occupancy. See Fig. 2 for an overview of the framework.

In our experiments, we evaluate ARCH on the task of 3D human reconstruction from a single image. Both quantitative and qualitative experimental results show ARCH outperforms state-of-the-art body reconstruction methods on public 3D scan benchmarks and in-the-wild 2D images. We also show that our reconstructed clothed humans can be animated by motion capture data, demonstrating the potential applications for human digitization for animation.

Contributions. The main contributions are threefold: 1) we introduce the Semantic Space (SemS) and Semantic Deformation Field (SemDF) to handle implicit function representation of clothed humans in arbitrary poses, 2) we propose opacity-aware differentiable rendering to refine our human representation via Granular Render-and-Compare, and 3) we demonstrate how reconstructed avatars can directly be rigged and skinned for animation. In addition, we learn per-pixel normals to obtain high-quality surface details, and surface albedo for relighting applications.

2. Related Work

3D clothed human reconstruction focuses on the task of reconstructing 3D humans with clothes. There are multiple attempts to solve this task with video inputs [2, 3, 37, 1, 52], RGB-D data [53, 56] and in multi-view settings [5, 13, 14, 45, 46, 47, 48, 6]. Though richer inputs clearly provide more information than single images, the developed pipelines yield more limitations on the hardware and additional time costs in deployment. Recently, some progress [7, 15, 18, 20, 21, 23, 41, 51, 54] has been made in estimating parametric human bodies from a single RGB image, yet boundaries are under-explored to what extent 3D clothing details can be reconstructed from such inputs. In recent work [22, 24, 4], the authors learn to generate surface geometry details and appearance using 2D UV maps. While details can be learned, the methods cannot reconstruct loose clothing (*e.g.*, dress) and recover complex shapes such as hair or fine structures (*e.g.*, shoe heels). Due to different types of clothing topology, volumetric reconstruction has great benefits in this scenario. For example, BodyNet [44] takes a person image as input and learns to reconstruct voxels of the person with additional supervision through body priors (*e.g.*, 2D pose, 3D pose, part mask); while PIFu [40] assumes no body prior and learns an implicit surface function based on aligned image features, leading more clothes details and less robustness against pose variations.

In this paper, we incorporate body prior knowledge to transform people in arbitrary poses to the canonical space, and then learn to reconstruct an implicit representation.

Differentiable rendering makes the rendering operation differentiable and uses it to optimize parameters of the scene representation. Existing approaches can be roughly divided into two categories: mesh rasterization based rendering [9, 19, 25, 29, 43] and volume based rendering [16, 26]. For example, OpenDR [29] and Neural Mesh Renderer [19] manually define approximated gradients of the rendering operation to move the faces. SoftRasterizer [25] and DIB-R [9], in contrast, redefine the rasterization as a continuous and differentiable function, allowing gradients to be computed automatically. For volume-based differen-

tiable rendering, [16] represents each 3D point as a multivariate Gaussian and performs occlusion reasoning with grid discretization and ray tracing. Such methods require an explicit volume to perform occlusion reasoning. [26] develops differentiable rendering for implicit surface representations with a focus on reconstructing rigid objects.

In contrast, we use a continuous rendering function as in [25], but revisit it to handle opacity, and we use geometric primitives at points of interest and optimize their properties.

3. Proposed Framework

ARCH contains three components, after 3D body estimation by [51] (see Fig. 2): pose-normalization using Semantic Space (SemS) and Semantic Deformation Field (SemDF), implicit surface reconstruction, and refinement using a differentiable renderer by Granular Render-and-Compare (see Sec. 3.4).

3.1. Semantic Space and Deformation Field

Our goal is to transform an arbitrary (deformable) object into a *canonical space* where the object is in a predefined *rest pose*. To do so, we introduce two concepts: the Semantic Space (SemS) and the Semantic Deformation Field (SemDF). SemS $S = \{(p, s_p) : p \in \mathbb{R}^3\}$ is a space consisting of 3D points where each point $p \in S$ is associated to semantic information s_p enabling the transformation operation. SemDF is a vector field represented by a vector-valued function $\mathcal{V}$ that accomplishes the transformation,

In computer vision and graphics, 3D human models have been widely represented by a kinematic structure mimicking the anatomy that serves to control the pose, and a surface mesh that represents the human shape and geometry. Skinning is the transformation that deforms the surface given the pose. It is parameterized by skinning weights that individually influence body part transformations [28]. In ARCH, we define SemS in a similar form, with skinning weights.

Assuming a skinned body template model T in a normalized A-pose (i.e., the *rest pose*), its associated skeleton in the canonical space, and skinning weights W, SemS is then

$$S = \{(p, \{w_{i,p}\}_{i=1}^{N_K}) : p \in \mathbb{R}^3\}, \tag{1}$$

where each point p is associated to a collection of skinning weights $\{w_{i,p}\}$ defined with respect to N_K body parts (e.g., skeleton bones). In this paper, we approximate $\{w_{i,p}\}$ by retrieving the closest point p' on the template surface to p and assigning the corresponding skinning weights from W. In practice, we set a distance threshold to cut off points that are too far away from T.

In ARCH, SemDF actually performs an *inverse-skinning* transformation, putting a human in arbitrary pose to its normalized-pose in the canonical space. This extends standard skinning (e.g., Linear Blend Skinning or LBS [28]) applied to structured objects to arbitrary 3D space and enables transforming an entire space in arbitrary poses to the canonical space, as every point p' can be expressed as a linear combination of points p with skinning weights $\{w_{i,p}\}$.

Following LBS, the canonical space of human body is tied to a skeletal rig. The state of the rig is described by relative rotations $R = \{r_i\}_{i=1}^{N_K}$ of all skeleton joints $X = \{x_i\}_{i=1}^{N_K}$. Every rotation is relative to the orientation of the parent element in a kinematic tree. For a skeleton with N_K body parts, $R \in \mathbb{R}^{3\times N_K}, X \in \mathbb{R}^{3\times N_K}$. Given a body template model T in rest pose with N_V vertices, the LBS function $\mathcal{V}(v_i, X, R; W)$ takes as input the vertices $v_i \in T$, the joints X, a target pose R, and deforms every v_i to the posed position v_i' with skinning weights $W \in \mathbb{R}^{N_V \times N_K}$, namely,

$$\mathcal{V}(v_i, X, R; W) = \sum\nolimits_{k=1}^{N_K} w_{k,i}\, G_k(R, X)\, v_i, \tag{2}$$

where $G_k(R, X)$ is the rest-pose corrected affine transformation to apply to body part k.

3.2. Implicit Surface Reconstruction

We use the occupancy map O to implicitly represent the 3D clothed human, *i.e.*,

$$O = \{(p, o_p) : \ p \in \mathbb{R}^3, 0 \le o_p \le 1\}, \tag{3}$$

where o_p denotes the occupancy for a point p. To obtain a surface, we can simply threshold τ the occupancy map O to obtain the isosurface O'_τ.

In this paper, we incorporate a human body prior by always reconstructing a neutral-posed shape in the canonical space. Similar to [40], we develop a deep neural network that takes a canonical space point p, its correspondent 2D position q, and the 2D image I as inputs and estimates occupancy o_p, normal n_p, color c_p for p; that is,

$$\begin{aligned} o_p &= \mathcal{F}(f_p^s, I; \theta_o), \\ n_p &= \mathcal{F}(f_p^s, I, f_p^o; \theta_n), \\ c_p &= \mathcal{F}(f_p^s, I, f_p^o, f_p^n; \theta_c), \\ f_p^s \in \mathbb{R}^{171},\ f_p^o &\in \mathbb{R}^{256},\ f_p^n \in \mathbb{R}^{64},\ f_p^c \in \mathbb{R}^{64}, \end{aligned} \tag{4}$$

where θ^o, θ^n and θ^c denote the occupancy, normal and color sub-network weights, f_p^s is the *spatial feature* extracted based on SemS. We use the estimated 57 canonical body landmarks from [51] and compute the Radial Basis Function (RBF) distance between p and the i-th landmark p'_i, that is

$$f_p^s(i) = \exp\{-\mathcal{D}(p, p'_i)\}, \tag{5}$$

where $\mathcal{D}(\cdot)$ is the Euclidean distance. We also evaluate the effects of different types of spatial features in Sec. 4.3. f_p^o and f_p^n the feature maps extracted from occupancy and normal sub-networks, respectively (see also Fig. 2). The three sub-networks are defined as follows:

The **Occupancy sub-network** uses a Stacked Hourglass (SHG) [34] as the image feature encoder and a Multi-Layer Perceptron (MLP) as the regressor. Given a 512×512 input image I, the SHG produces a feature map $f \in \mathbb{R}^{512\times512\times256}$ with the same grid size. For each 3D point p, we consider the feature located at the corresponding projected pixel q as its visual feature descriptor $f_p^o \in \mathbb{R}^{256}$. For points that do not align onto the grid, we apply bi-linear in-

terpolation on the feature map to obtain the feature at that pixel-aligned location. The MLP takes the spatial feature of the 3D point $p \in \mathbb{R}^3$ and the pixel-aligned image features $f_p^o \in \mathbb{R}^{256}$ as inputs and estimates the occupancy $o_p \in [0, 1]$ by classifying whether this point lies inside the clothed body or not.

The **Normal sub-network** uses a U-net [39] as the image feature encoder and a MLP which takes the spatial feature, and feature descriptors $f_p^n \in \mathbb{R}^{64}$ and $f_p^o \in \mathbb{R}^{256}$ from its own backbone and from the occupancy sub-network as inputs and estimates the normal vector n_p.

The **Color sub-network** also uses a U-net [39] as the image feature encoder and a MLP which takes the spatial feature, and feature descriptors $f_p^c \in \mathbb{R}^{64}$, $f_p^n \in \mathbb{R}^{64}$ and $f_p^o \in \mathbb{R}^{256}$ from its own backbone, as well as the normal and occupancy sub-networks as inputs and estimates the color c_p in RGB space.

For each sub-network, the MLP takes the pixel-aligned image features and the spatial features (as described in Sec. 3.1), where the numbers of hidden neurons are $(1024, 512, 256, 128)$. Similar to [40], each layer of MLP has skip connections from the input features. For the occupancy sub-network, the MLP estimates one-dimension occupancy $o_p \in [0, 1]$ using Sigmoid activation. For the normal sub-network, the MLP estimates three-dimension normal $n_p \in [0, 1]^3$, $\|n_p\|_2 = 1$ using L2 normalization. For the color sub-network, the MLP estimates three-dimension color $c_p \in [0, 1]^3$ using range clamping.

3.3. Training

During training, we optimize the parameters of all three sub-models, *i.e.*, the occupancy, normal and color models. We define the training in three separate loops to train each part with the appropriate losses and avoid computational bottlenecks. The total loss function is defined as

$$\mathcal{L} = \mathcal{L}_{3d}^o + \mathcal{L}_{3d}^n + \mathcal{L}_{3d}^c + \mathcal{L}_{2d}^n + \mathcal{L}_{2d}^c, \tag{6}$$

where $\mathcal{L}_{3d}^o$ is the 3D loss for occupancy network, $\mathcal{L}_{3d}^n$ and $\mathcal{L}_{2d}^n$ are the 3D and 2D losses for normal network, and $\mathcal{L}_{3d}^c$ and $\mathcal{L}_{2d}^c$ are the 3D and 2D losses for color network. For every training iteration, we perform the following three optimizations.

Occupancy. We use the available ground truth to train the occupancy prediction model in a direct and supervised way. First, we sample $20\,480$ points in the canonical space. They are sampled around the template mesh according to a normal distribution with a standard deviation of $5\,\mathrm{cm}$. This turned out to cover the various body shapes and clothing well in our experiments, but can be selected according to the data distribution at hand. These points are then processed by the occupancy model, providing us with an estimated occupancy value for every sampled point. We use a *sigmoid* function on these values to normalize the network output to the interval $[0, 1]$, where we select 0.5 as the position of the isosurface. 0.5 is the position where the derivative of the *sigmoid* function is the highest and we expect to optimize the surface prediction best. The loss $\mathcal{L}_{3d}^o$ is defined as the Huber loss comparing the occupancy prediction and ground truth. Similar to [36], we found a less aggressive loss function than the squared error better suited for the optimization, but found the quadratic behavior of the Huber loss around zero to be beneficial.

Normals and colors for surface points. Colors and normals can be optimized directly from the ground truth mesh for points that lie on its surface. To use this strong supervision signal we introduce a dedicated training stage. In this stage, we sample points only from the mesh surface and push them through the color and normal models. In our setup, we use $51\,200$ point samples per model per training step. The loss terms $\mathcal{L}_{3d}^n$ and $\mathcal{L}_{3d}^c$ are defined as the $L1$ loss comparing the predicted normals and colors with the ground truth across all surface points. The occupancy predictions are kept unchanged.

Normals and colors for points not on the surface. For points not on the mesh surface, it is not clear how the ground truth information can be used in the best way to improve the prediction without an additional mapping. In a third step for the training, we sample another set of $51\,200$ points, and push them through the occupancy, color and normal models and use a differentiable renderer on the prediction. We render the image using the occupancy information as opacity, and by using the color channels to represent colors or normals and use the gradients to update the predicted values. $\mathcal{L}_{2d}^n$ and $\mathcal{L}_{2d}^c$ are defined as the per-pixel L1 loss between the rendered image and the ground truth. For details on this step, see Fig. 3 and the following Sec. 3.4.

3.4. Granular Render-and-Compare

The prediction from the model is an implicit function representation. By sampling points in a predefined volume and optimizing $\mathcal{L}_{3d}^o$, $\mathcal{L}_{3d}^n$ and $\mathcal{L}_{3d}^c$, we can optimize the occupancy, normal and color at these points directly given 3D ground truth. However, it is not clear what the gradients should be for points that are located not directly on the surface of the ground truth mesh. To address this problem, we propose to use a differentiable renderer.

We first create an explicit geometric representation of the scene at hand. For every sample point to optimize, we place a geometric primitive with a spatial extent at its position. To be independent of the viewpoint, we choose this to a sphere with $1\,\mathrm{cm}$ radius for every sampled point (for an overview of the differentiable rendering loss computation, see Fig. 3). During training, every scene to render contains $51\,200$ spheres.

We then define a differentiable rendering function [25] to project the spheres onto the image plane so that we can perform pixel-level comparisons with the projected ground truth. We use a linear combination with a weight w_j^i to associate the color contribution from point p_i to the pixel q_j. Having the color c_i and normal n_i for point p_i, the color and normal for pixel q_j are calculated as the weighed linear

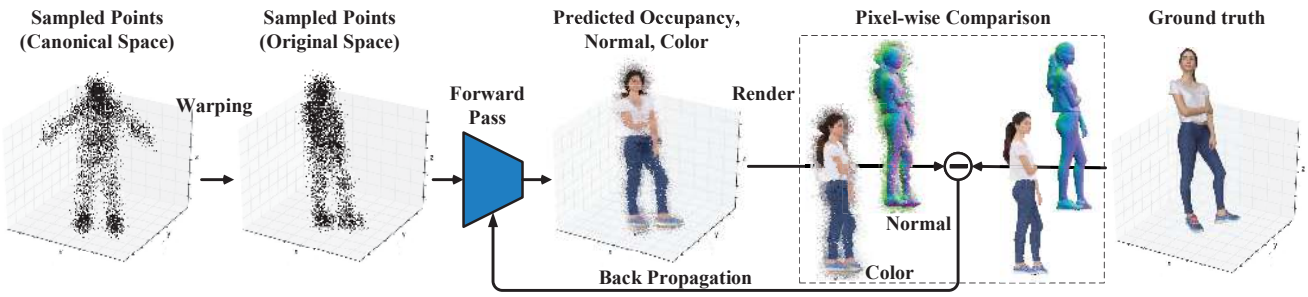


Figure 3. *Illustration of the loss computation through differentiable rendering*. From left to right: points are sampled according to a Gaussian distribution around our template mesh in the canonical space. They are transformed with the estimated Semantic Deformation Field and processed by the model. The model provides estimations of occupancy, normal and color for each 3D point. We use a differentiable renderer to project those points onto a new camera view and calculate pixel-wise differences to the rendered ground truth.

combination of point values $\sum_i w_j^i c_i$ and $\sum_i w_j^i n_i$.

We define w_j^i considering two factors: the depth of the sphere for point p_i at pixel q_j, z_j^i, and the proximity of the projected surface of the sphere for point p_i to pixel q_j, d_j^i. To make occlusion possible, the depth needs to have a strong effect on the resulting weight. Hence, [25] defines the weight as

$$w_j^i = \frac{d_j^i \exp(z_j^i/\gamma)}{\sum_k d_k^i \exp(z_k^i/\gamma) + \exp(\epsilon/\gamma)} \tag{7}$$

with ϵ being a small numerical constant. With this definition, the proximity has linear influence on the resulting weight while the depth has exponential influence. The impact ratio is controlled by the scaling factor γ, which we fix to 1×10^{-5} in our experiments.

In contrast to [25] we also need to use an opacity α_i per sphere for rendering. We tie this opacity value α_i directly to the predicted occupancy value through linear scaling and shifting. To stay with the formulation of the render function, we integrate α_i into the weight formulation in Eqn. 7.

If the opacity is used as a linear factor in this equation, the *softmax* function will still render spheres with very low opacity over other spheres with a lower depth value. The problem is the exponential function that is applied to the scaled depth values. On the other hand, if an opacity factor is only incorporated into the exponential function, spheres will remain visible in front of the background (their weight factor is still larger than the background factor $\exp(\epsilon/\gamma)$). We found a solution by using the opacity value as both, linear scaling factor as well as exponential depth scaling factor. This solution turned out to be numerically stable and well-usable for optimization with all desired properties. This changes the weight function to the following:

$$w_j^i = \frac{\alpha^i d_j^i \exp(\alpha^i z_j^i/\gamma)}{\sum_k \alpha^i d_k^i \exp(\alpha^i z_k^i/\gamma) + \exp(\epsilon/\gamma)}. \tag{8}$$

Using this formulation, we optimize the color channel values c_i and normal values n_i per point. A per-pixel L1 loss is computed between the rendering and a rendering of the ground truth data and back-propagated through the model. For our experiments with $\gamma = 1 \times 10^{-5}$ and the depth of the volume, we map the occupancy values that define the isosurface at the value 0.5 to the threshold where α shifts to transparency. We experimentally determined this value to be roughly 0.7.

3.5. Inference

For inference, we take as input a single RGB image representing a human in an arbitrary pose, and run the forward model as described in Sec. 3.2 and Fig. 2. The network outputs a densely sampled occupancy field over the canonical space from which we use the Marching Cube algorithm [30] to extract the isosurface at threshold 0.5. The isosurface represents the reconstructed clothed human in the canonical pose. Colors and normals for the whole surface are also inferred by the forward pass and are pixel-aligned to the input image (see Sec. 3.2). The human model can then be transformed to its original pose R by LBS using SemDF and per-point corresponding skinning weights W as defined in Sec. 3.1.

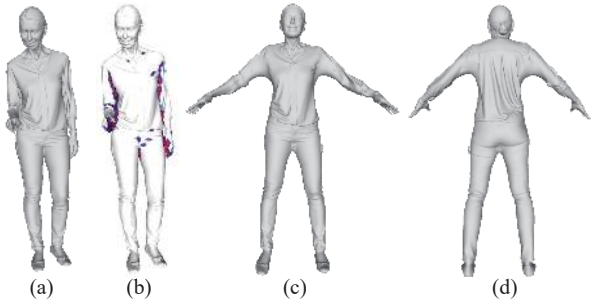
Furthermore, since the implicit function representation is equipped with skinning weights and skeleton rig, it can naturally be warped to arbitrary poses. The proposed end-to-end framework can then be used to create a detailed 3D avatar that can be animated with unseen sequences from a single unconstrained photo (see Fig. 5).

4. Experiments

We present details on ARCH implementation and datasets for training, with results and comparisons to the state of the art.

4.1. Implementation Details

ARCH is implemented in PyTorch. We train the neural network model using the RMSprop optimizer with a learning rate starting from 1e-3. The learning rate is updated using an exponential schedule every 3 epochs by multiplying with the factor 0.1. We are using 582 3D scans to train the model and use 360 views per epoch, resulting in $209\,520$ images for the training per epoch. Training the model on an NVIDIA DGX-1 system with one Tesla V100 GPU takes 90 h for 9 epochs.



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

Figure 4. *Illustration of reposing 3D scans to the canonical space.* **(a)** An original 3D scan from the RenderPeople dataset. **(b)** Automatically detected topology changes. Red marks points with self contacts, blue regions that are also removed before reposing to avoid problems with normals. **(c, d)** Reposed scan.

4.2. Datasets

Our training dataset is composed of 375 3D scans from the RenderPeople[1] dataset, and 207 3D scans from the AXYZ[2] dataset. The scans are watertight meshes which are mostly free of noise. They represent subjects wearing casual clothes, and potentially holding small objects (*e.g.*, mobile phones, books and purses). Our test dataset contains 64 scans from the RenderPeople dataset, 207 scans from the AXYZ dataset, 26 scans from the BUFF dataset [55], and 2D images from the DeepFashion [27] dataset, representing clothed people with a large variety of complex clothing. The subjects in the training dataset are mostly in standing pose, while the subjects in the test dataset are in arbitrary poses (standing, bending, sitting, ...). We create renders of the 3D scans using Blender. For each 3D scan, we produce 360 images by rotating a camera around the vertical axis with intervals of 1 degree. For the current experiments, we only considered the weak perspective projection (orthographic camera) but this can be easily adapted. We also used 38 environment maps to render each scan with different natural lighting conditions. The proposed model is trained to predict albedo (given by ground truth scan color). We also observed that increasing the number of images improves the fidelity of predicted colors (as in [40]).

In order to use a 3D scan for model training, we fit a rigged 3D body template to the scan mesh to estimate the 3D body pose (see Fig. 4). The estimated parametric 3D body can directly serve as ground truth input data during the model training step (see Sec. 3.3). This also allows us to obtain SemS and SemDF for the scan. However, since each 3D scan has its own topology, artifacts due to topology changes will occur when pose-normalization is naively applied to models containing self-contact (for example arms touching the body). This creates inaccurate deformations. Hence, we first detect regions of self-contact and topology changes and cut the mesh before pose-normalization (see Fig. 4 (c) and (d)). Holes are then filled up using Smooth Signed Distance Surface reconstruction [8] (see Fig. 4 (c) and (d)). For inference on 2D images from the DeepFashion dataset, we obtain 3D body poses using the pre-trained models from [51].

[1] http://renderpeople.com

[2] http://secure.axyz-design.com

Methods	RenderPeople			BUFF		
	Normal	P2S	Chamfer	Normal	P2S	Chamfer
BodyNet [44]	0.26	5.72	5.64	0.31	4.94	4.52
SiCloPe [32]	0.22	3.81	4.02	0.22	4.06	3.99
IM-GAN [10]	0.26	2.87	3.14	0.34	5.11	5.32
VRN [17]	0.12	1.42	1.6	0.13	2.33	2.48
PIFu [40]	0.08	1.52	1.50	0.09	1.15	1.14
ARCH, baseline	0.080	1.98	1.85	0.081	1.74	1.75
+ SemDF	0.042	**0.74**	**0.85**	0.045	**0.82**	**0.87**
+ GRaC	**0.038**	**0.74**	**0.85**	**0.040**	**0.82**	**0.87**

Table 1. *Quantitative comparisons of normal, P2S and Chamfer errors between posed reconstruction and ground truth on the RenderPeople and BUFF datasets.* Lower values are better.

4.3. Results and Comparisons

We evaluate the reconstruction accuracy of ARCH with three metrics similar to [40]. We reconstruct the results on the same test set and repose them back to the original poses of the input images and compare the reconstructions with the ground truth surfaces in the original poses. We report the average point-to-surface Euclidean distance (P2S) in centimeters, the Chamfer distance in centimeters, and the L2 normal re-projection error in Tab. 1.

Additionally to comparing with state-of-the-art methods [10, 17, 18, 32, 40, 44], we include scores of an ablative study with the proposed method. In particular, we evaluate three variants and validate the effectiveness of two main components: the Semantic Deformation Field and the Granular Render-and-Compare loss.

ARCH, baseline: a variant of [40] using our own network specifications, taking an image as input and directly estimating the implicit surface reconstruction.

Semantic Deformation Field (SemDF): we first estimate the human body configuration by [51] and then reconstruct the canonical shape using the implicit surface reconstruction, and finally repose the canonical shape to the original pose in the input image.

Granular Render-and-Compare (GRaC): based on the previous step, we further refine the reconstructed surface normal and color using differentiable render-and-compare.

ARCH baseline specification already achieves state-of-the-art performance in normal estimation, but has inferior performance w.r.t. P2S and Chamfer error compared to PIFu [40]. We use a different training dataset compared to PIFu that apparently does not represent the test set as well. Also, PIFu normalizes every scan at training and prediction time to have its geometric center at the coordinate origin, whereas we use origin placed scans with slight displacements. Lastly, PIFu performs a size normalization of the body using the initial 3D body configuration estimate. The image is rescaled so that the height of the person matches the canonical size. This makes person height estimation for PIFu impossible, whereas we properly reconstruct it—at the

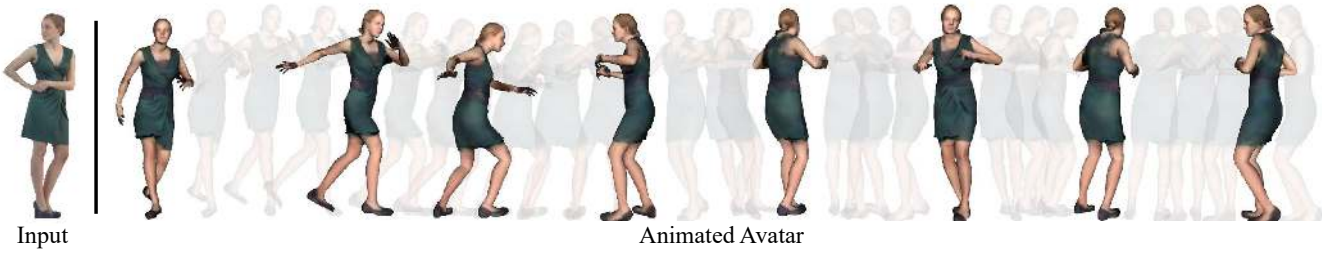


Figure 5. *An example for animating a predicted avatar.* We use a predicted, skinned avatar from our test set and drive it using off-the-shelf motion capture data. This avatar has been created using only a single, frontal view. Our model produces a plausible prediction for the unseen parts, for example the hair and the back of the dress.

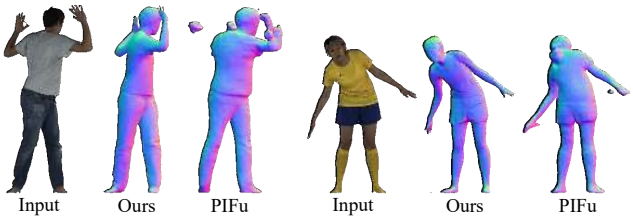


Figure 6. *Evaluation on BUFF.* Our method outperforms [40] for detailed reconstruction from arbitrary poses. We show results from different angles.

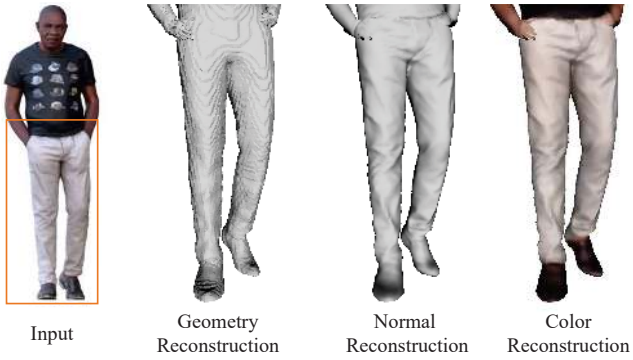


Figure 7. *Reconstruction quality of clothing details.* The geometry reconstruction from our method reproduces larger wrinkles and the seam of the pants and shoes while the predicted normals reproduce fine wrinkles. The normal and color predictions rendered together produce a plausible image.

cost of a more difficult task to solve. The benefit of this operation is not reflected in the scores because the metrics are calculated in the original image space.

When adding SemDF, we see a substantial gain in performance compared to our own baseline, but also to the so far best-performing PIFu metrics. We outperform PIFu on average with an improvement of over 50% on the RenderPeople dataset and an average improvement of over 60% on the BUFF dataset. When adding the Granular Render-and-Compare loss, these number improve again slightly, especially on the normal estimation. Additionally, the results gain a lot of visual fidelity and we manage to remove a lot of visual artifacts.

Fig. 7 shows the level of detail of geometry, normal and color predictions our model can achieve. Note that, for example, the zipper is not reproduced in the predicted normal

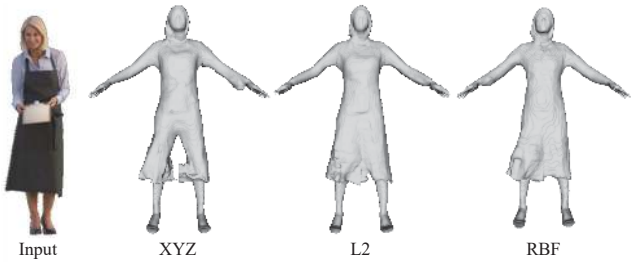


Figure 8. *Reconstruction example using different types of spatial features.* **XYZ**: absolute coordinates, **L2**: Euclidean distances to each joint, **RBF**: Radial basis function based distance to each joint. The proposed RBF preserves notably more details.

Spatial Feature Types	Normal	P2S	Chamfer
XYZ	0.045	0.75	0.91
L2	0.043	0.76	0.89
RBF	**0.042**	**0.74**	**0.85**

Table 2. *Ablation study on the effectiveness of spatial features.* The XYZ feature uses the plain location of body landmarks. The L2 and RBF features both improve the performance.

map. This is an indicator that the model does not simply reproduce differences in shading directly in the normal map, but is able to learn about geometric and shading properties of human appearance. In Fig. 6, we show qualitative results on challenging poses from the BUFF dataset. In Fig. 9, we provide a comparison of results of our method with a variety of state of the art models [44, 18, 40].

Ablative Studies. We evaluate the effectiveness of different types of spatial features in Tab. 2 and Fig. 8. We evaluate three different features: **XYZ** uses the absolute position of the sampled point, **L2** uses the Euclidean distance from the sampled point to each body landmark, and **RBF** denotes our proposed method in Sec. 3.1. It can be observed that RBF feature works best for this use case both qualitatively and quantitatively. RBF features strongly emphasize features that are close in distance to the currently analyzed point and puts less emphasis on points further away, facilitating optimization and preserving details.

Animating Reconstructed Avatars. With the predicted occupancy field we can reconstruct a mesh that is already rigged and can directly be animated. We show the animation of an avatar we reconstructed from the AXYZ dataset in Fig. 5, driven by an off-the-shelf retargetted Mixamo an-

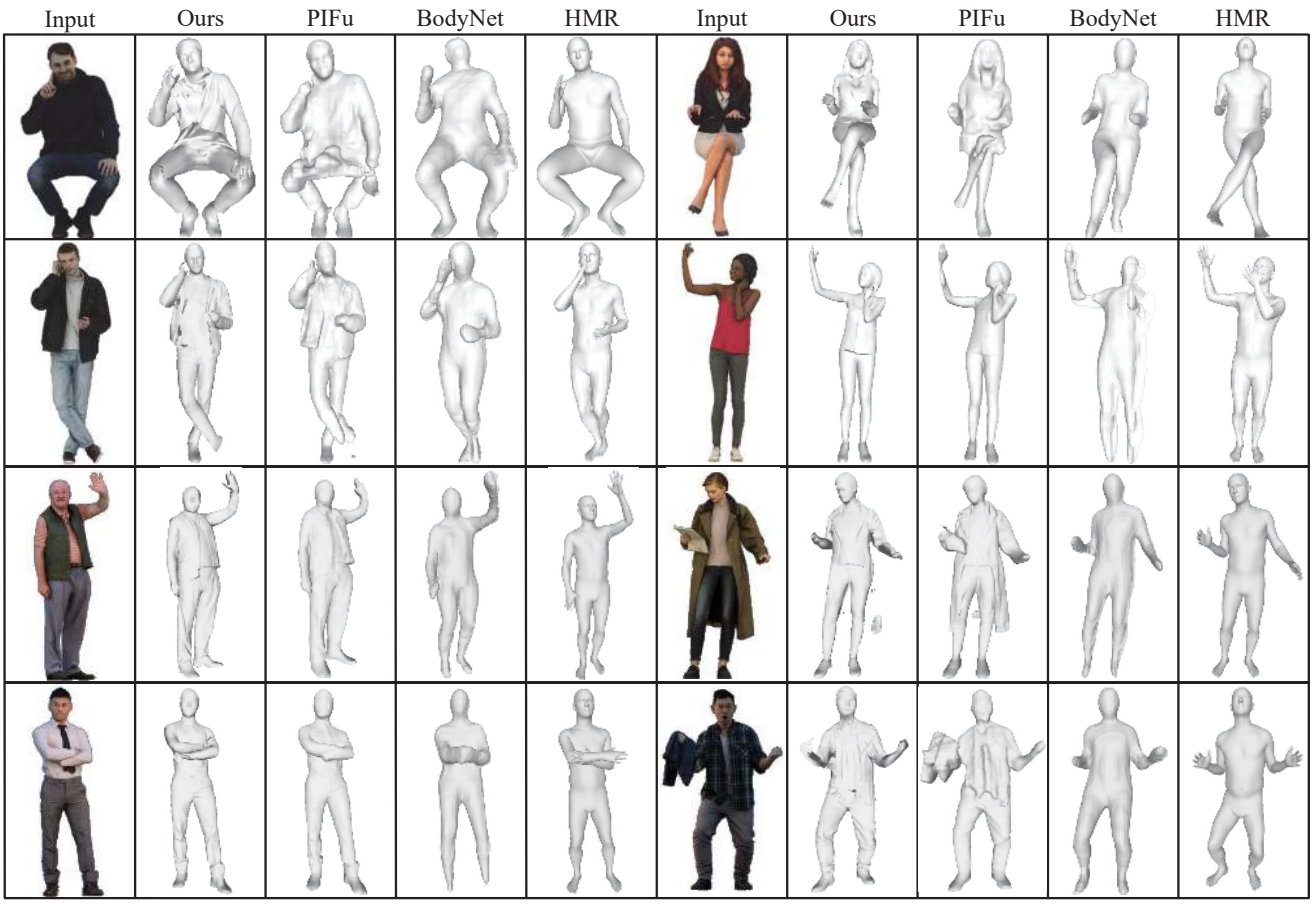


Figure 9. *Qualitative comparisons against state-of-the-art methods [18, 44, 40] on unseen images.* ARCH (Ours) handles arbitrary poses with self-contact and occlusions robustly, and reconstructs a higher level of details than existing methods. Images are from RenderPeople. Results on DeepFashion are of similar quality but are not shown due to copyright concerns. Please contact us for more information.

Figure 10. *Challenging cases.* Reconstruction of rare poses, and details in occluded areas could be further improved.

imation [51]. By working in the canonical space, the avatar is automatically rigged and can be directly animated. Given only a single view image, the avatar is reconstructed in 3D and looks plausible from all sides.

As shown in Fig 10, rare poses not sufficiently covered in the training dataset (*e.g.*, kneeing) return inaccurate body prior, and are then challenging to reconstruct. Also, details (i.e., normals) in occluded areas could be improved with specific treatment of occlusion-aware estimation.

5. Conclusion

In this paper, we propose ARCH, an end-to-end framework to reconstruct clothed humans from unconstrained photos. By introducing the Semantic Space and Semantic Deformation Field, we are able to handle reconstruction from arbitrary pose. We also propose a Granular Render-and-Compare loss for our implicit function representation to further constrain visual similarity under randomized camera views. ARCH shows higher fidelity in clothing details including pixel-aligned colors and normals with a wider range of human body configurations. The resulting models are animation-ready and can be driven by arbitrary motion sequences. We will explore handling heavy occlusion cases with in-the-wild images in the future.

Acknowledgements. We would like to thank Junbang Liang and Yinghao Huang (Interns at FRL) for their work on dataset creation.

References

[1] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2019. 2

[2] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *International Conference on 3D Vision*, 2018. 2

[3] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[4] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019. 2

[5] Alexandru O. Balan, Leonid Sigal, Michael J. Black, James E. Davis, and Horst W. Haussecker. Detailed human shape and pose from images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 2

[6] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019. 2

[7] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, 2016. 2

[8] Fatih Calakli and Gabriel Taubin. SSD: smooth signed distance surface reconstruction. *Comput. Graph. Forum*, 30(7):1993–2002, 2011. 6

[9] Wenzheng Chen, Jun Gao, Huan Ling, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *Annual Conference on Neural Information Processing Systems*, 2019. 2

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 6

[11] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *AAAI Conference on Artificial Intelligence*, 2018. 1

[12] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 1

[13] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362—1376, 2010. 2

[14] Juergen Gall, Carsten Stoll, Edilson de Aguiar, Christian Theobalt, Bodo Rosenhahn, and Hans-Peter Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 2

[15] Riza Alp Guler and Iasonas Kokkinos. Holopose: Holistic 3d human reconstruction in-the-wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[16] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Annual Conference on Neural Information Processing Systems*, 2018. 2, 3

[17] Aaron S. Jackson, Chris Manafas, and Georgios Tzimiropoulos. 3d human body reconstruction from a single image via volumetric regression. *European Conference of Computer Vision Workshops*, 2018. 6

[18] Angjoo Kanazawa, Michael J. Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 6, 7, 8

[19] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[20] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *IEEE International Conference on Computer Vision*, 2019. 2

[21] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[22] Zorah Laehner, Daniel Cremers, and Tony Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *European Conference on Computer Vision*, 2018. 2

[23] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[24] Verica Lazova, Eldar Insafutdinov, and Gerard Pons-Moll. 360-degree textures of people in clothing from a single image. In *International Conference on 3D Vision*, 2019. 2

[25] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *IEEE International Conference on Computer Vision*, 2019. 2, 3, 4, 5

[26] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. *Annual Conference on Neural Information Processing Systems*, 2019. 2, 3

[27] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6

[28] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):248, 2015. 3

[29] Matthew M. Loper and Michael J. Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, 2014. 2

[30] William E. Lorensen and Harvey E. Cline. Differentiable monte carlo ray tracing through edge sampling. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987. 5

[31] Takashi Matsuyama, Shohei Nobuhara, Takeshi Takai, and Tony Tung. *3D Video and Its Applications*. Springer, 2012. 1

[32] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 6

[33] Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1

[34] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, 2016. 3

[35] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. In *International Conference on 3D Vision*, 2018. 1

[36] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4

[37] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. Clothcap: seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics*, 36(4):73:1—73:15, 2017. 2

[38] Hang Qi, Yuanlu Xu, Tao Yuan, Tianfu Wu, and Song-Chun Zhu. Scene-centric joint parsing of cross-view videos. In *AAAI Conference on Artificial Intelligence*, 2018. 1

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, 2015. 4

[40] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *IEEE International Conference on Computer Vision*, 2019. 2, 3, 4, 6, 7, 8

[41] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *Annual Conference on Neural Information Processing Systems*, 2017. 2

[42] Tony Tung, Shohei Nobuhara, and Takashi Matsuyama. Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo. In *IEEE 12th International Conference on Computer Vision ICCV*, 2009. 1

[43] Fredo Durand Tzu-Mao Li, Miika Aittala and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics*, 37(6):222:1–222:11, 2018. 2

[44] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. BodyNet: Volumetric inference of 3D human body shapes. In *European Conference on Computer Vision*, 2018. 1, 2, 6, 7, 8

[45] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popovic. Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics*, 27(3):97:1—-97:9, 2008. 2

[46] Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popovic, Szymon Rusinkiewicz, and Wojciech Matusik. Dynamic shape capture using multi-view photometric stereo. In *ACM SIGGRAPH*, 2009. 2

[47] Ramesh Raskar Leonard McMillan Wojciech Matusik, Chris Buehler and Steven Gortler. Image-based visual hulls. In *ACM SIGGRAPH*, 2000. 2

[48] Chenglei Wu, Kiran Varanasi, and Christian Theobalt. Full body performance capture under uncontrolled and varying illumination: A shading-based approach. In *European Conference on Computer Vision*, 2012. 2

[49] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[50] Yuanlu Xu, Xiaobai Liu, Lei Qin, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *AAAI Conference on Artificial Intelligence*, 2017. 1

[51] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. DenseRaC: Joint 3D pose and shape estimation by dense render-and-compare. In *IEEE International Conference on Computer Vision*, 2019. 1, 2, 3, 6, 8

[52] Jinlong Yang, Jean-Sébastien Franco, Franck Hétroy-Wheeler, and Stefanie Wuhrer. Estimation of human body shape in motion with wide clothing. In *European Conference on Computer Vision*, 2016. 2

[53] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[54] Yixuan Wei Qionghai Dai Yebin Liu Zerong Zheng, Tao Yu. Deephuman: 3d human reconstruction from a single image. In *IEEE International Conference on Computer Vision*, 2019. 2

[55] Chao Zhang, Sergi Pujades, Michael Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 6

[56] Zerong Zheng, Tao Yu, Hao Li, Kaiwen Guo, Qionghai Dai, Lu Fang, and Yebin Liu. Hybridfusion: Real-time performance capture using a single depth sensor and sparse imus. In *European Conference on Computer Vision*, 2018. 2