S³: Neural Shape, Skeleton, and Skinning Fields for 3D Human Modeling

Ze Yang[1,2]   Shenlong Wang[1,2]   Sivabalan Manivasagam[1,2]   Zeng Huang[3]
Wei-Chiu Ma[1,4]   Xinchen Yan[1]   Ersin Yumer[1]   Raquel Urtasun[1,2]

[1]Uber Advanced Technologies Group   [2]University of Toronto,
[3]University of Southern California   [4]Massachusetts Institute of Techonology

{zeyang, slwang, manivasagam, urtasun}@cs.toronto.edu,
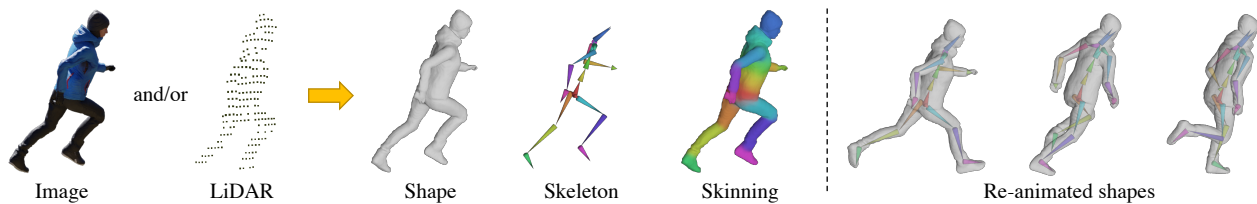zenghuan@usc.edu, weichium@mit.edu, {xcyan, yumer}@uber.com

Figure 1: Given a single image and/or a single LiDAR sweep as input, our model infers shape, skeleton and skinning jointly, which can then be used to generate animated 3D characters in novel poses.

Abstract

Constructing and animating humans is an important component for building virtual worlds in a wide variety of applications such as virtual reality or robotics testing in simulation. As there are exponentially many variations of humans with different shape, pose and clothing, it is critical to develop methods that can automatically reconstruct and animate humans at scale from real world data. Towards this goal, we represent the pedestrian's shape, pose and skinning weights as neural implicit functions that are directly learned from data. This representation enables us to handle a wide variety of different pedestrian shapes and poses without explicitly fitting a human parametric body model, allowing us to handle a wider range of human geometries and topologies. We demonstrate the effectiveness of our approach on various datasets and show that our reconstructions outperform existing state-of-the-art methods. Furthermore, our re-animation experiments show that we can generate 3D human animations at scale from a single RGB image (and/or an optional LiDAR sweep) as input.

1. Introduction

Realistic, articulated human simulation is an important task with a wide range applications. It helps to bring characters to life in video games and movies [61], provides realistic AR/VR experiences in the context of sports [71, 67]

and social media [12], and is starting to play an important role in realistic simulations for testing robotic systems in both indoor [51, 58, 39] and outdoor environments [20, 64]. Traditionally, human reconstruction and animation is a time-consuming manual process. An artistic designer needs to create a scale-appropriate *"joint estimation"* that specifies the human's degrees of freedom, *"reconstruct"* the human 3D mesh geometry, and *"skin"* the mesh by describing how the vertex positions deform as a function of the skeleton joints pose. Finally, the artist must specify the pose sequence that enacts an animation. This manual approach is neither cost-effective nor efficient if we want to reconstruct and animate 3D humans at scale, for example by digitizing millions of different pedestrians observed in urban city scenes.

Automating the human reconstruction and animation is very challenging as there are large variations in pedestrian shape, pose, clothing, and accoutrement. Most existing animatable human modeling methods are pipelined systems consisting of a set of modules that run separately [59, 53, 61]. Typically, the first module performs joint estimation using marker [17] or markerless [11] motion capture. From either dense 3D scans or images, the system then reconstructs shape [14, 6, 34, 30, 66, 5, 25, 4, 21, 49] and texture [56, 55, 41]. The estimated pose skeleton is further associated with the mesh typically via closed-form skinning models [27, 28] with hand-designed or automatic weight painting [8, 19]. Unfortunately, for existing approaches to work successfully, they typically rely on an expensive 3D

scanner in a controlled environment, a multi-camera cage [59, 53], or alternatively require relatively controlled viewpoints, typical frontal views with small variations [61]. Few works so far have performed fully automatic end-to-end reconstruction and animation of clothed humans [22, 62, 9], but they all require fitting to a parametric body model, making it difficult to animate humans that deviate signficantly in geometry (i.e. non-tight clothing such as skirts and dresses).

In this paper, we propose a scalable solution by reconstructing 3D animatable humans in the wild that takes advantage of sensory data captured around our cities. However, this setting brings new challenges: in-the-wild data lacks ground-truth 3D shape and pose, making supervision for deep-learning based models challenging. Furthermore, the captured sensory data can be noisy, of low-resolution, under non-canonical views and poses, and with various lighting conditions. These conditions are especially challenging for skinning, which requires accurate correspondence of each surface location to its corresponding body part. Inaccurate assignments will cause large aberrations during animation. To address these challenges, we propose a novel approach that takes sensor data captured at a single viewpoint (i.e. image and/or LiDAR sweep) of a pedestrian and jointly predicts 3D mesh, skeleton joints, and skinning weights, all with a single network (see Figure 1). The resulting animatable pedestrian can be directly deformed to novel poses and placed into new scenarios using either motion capture data or artist-created animations. Inspired by the recent success of implicit modeling [42, 7, 15] and neural radiance fields [38], we represent a 3D human as a continuous multi-dimensional neural field, which outputs the occupancy, human joint probability as well as skinning weights given each input 3D location in continuous space. This representation is very flexible and can capture fine details of clothed humans, handle different surface topologies (i.e. skirts), and adapt well to unseen human shapes as it is not constrained by a parametric model with a fixed (mesh) topology. In addition, our end-to-end architecture overcomes the challenge of error propagation in conventional pipelines.

We demonstrate the effectiveness of our approach on both photorealistic synthetic human 3D data and a large-scale real-world self-driving dataset. Our approach achieves better quantitative and qualitative performance compared to state-of-the-art methods in terms of shape reconstruction quality. Importantly, we also show that we can reliably re-animate the reconstructed 3D human given novel poses.

2. Related work

Human Reconstruction: Early works such as SCAPE [6] and SMPL [34] proposed to represent human body shape and pose-depedent variations using a parametric model. Building upon them, there have been efforts in extending SMPL by modeling soft-tissue body motions [46], integrating hand ges-

tures [47] and facial expressions [24, 43]. Later works proposed to estimate 3D pose and shape from a single image by minimizing the joint re-projection error [10, 25], silhouettes re-projection error with a differentiable renderer [57, 45]. Recent works also explored the cross-frame consistency in monocular videos [26, 44], human-object interactions [68], and part-specific attention [16] to improve the reconstruction quality towards the next level. One limitation of SMPL is that it only captures the shape space of naked human body, which inevitability leads to reality gap in practice. Most recent works [70, 4, 36, 5, 23] introduced per-vertex deformation or layered garment to better reconstruct the clothed human body.

Continuous Neural Representation: Recent years, deep implicit functions (or more generally, continuous neural field) has enjoyed increasing popularity in 3D representation for static scenes or rigid objects [42, 37, 63] and deformable objects such as human body shapes [15, 49, 50, 21, 69]. Those approaches generally have more degrees of freedom to capture the variations in human shape and clothing. Continuous neural representation has also shown to achieve decent performance in novel view synthesis [38, 52] and object recognition [29, 65].

Animating Human: A common approach in computer graphics to animate a human character is through skeletal animation. It requires an articulated object represented with two components: a surface representation and a hierarchical structure called skeleton. Skinning binds each surface vertex to the bones with skinning weights, measuring how the mesh's vertex will change when a bone moves. Linear blend skinning (LBS) [32] linearly combines the transformation from each bone through skinning weights and applies the composed transformation to mesh vertices. SMPL [34] extends LBS with identity-driven shape variations and pose-dependent shape variations. Jointly reconstructing and animating humans from an image is an emerging domain in vision and graphics. Recent approaches [61, 25, 31] reconstruct and animate from a single photo by using the SMPL [34] model. The implicit function-based methods achieve better reconstruction quality in handling cloth and accessories. However, animation requires additional post-processing, such as generating a warp field guided by parametric model [22], or explicitly fitting the SMPL model as a post-processing step [9]. Unlike these work, we predict all the necessary components for animation: shape, skeleton, and skinning through a single unified network.

3. Reconstructing Animatable Pedestrians

We propose $\mathbf{S}^3$, an end-to-end neural **S**hape, **S**keleton, and **S**kinning field model to reconstruct realistic and animatable 3D humans from either a single camera image, a single view LiDAR sweep, or a paired image and LiDAR input.
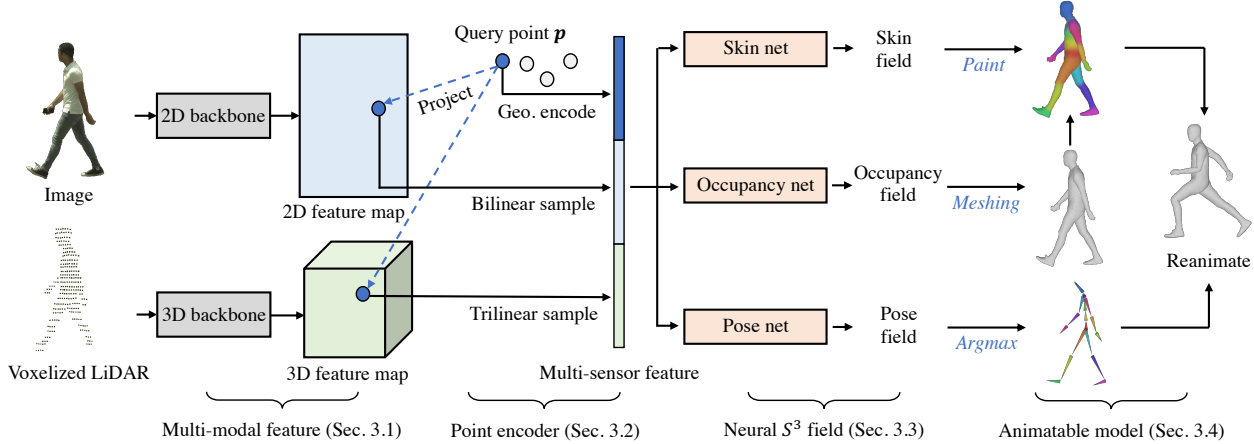
Figure 2: Overview of our proposed method. From left to right: We process input sensor data into spatial feature representations (Sec 3.1). We query points adaptively from 3D space and extract their point encoding (Sec. 3.2), which we use to query our neural implicit representations of shape, pose, and skinning (Sec. 3.3). We apply post-processing to construct the final explicit representation of an animatable person (Sec. 3.4).

Assuming backgrounds are masked out for both image and LiDAR, our network takes image $\mathcal{I}$ and/or LiDAR $\mathcal{C}$ as input to form $\mathbf{x} = (\mathcal{I}, \mathcal{C}) \in \mathcal{X}$. For a 3D query point in continuous space $\mathbf{p} \in \mathbb{R}^3$, the network outputs a continuous-valued multi-dimensional vector representing occupancy, joints positions as well as the skinning weights.

Fig. 2 depicts an overview of our approach. We first encode the input $\mathbf{x}$ into 2D and 3D feature representations via a backbone feature network (Sec. 3.1). An implicit feature vector is then obtained for a query 3D point based on the generated feature representations (Sec. 3.2). Given the feature vector at a 3D location, its occupancy probability, human joint probability and skinning weight are predicted using a fully-connected network (Sec. 3.3). Finally, shapes, skeletons and skinning weights are extracted based on the final predictions of the sampled points (Sec. 3.4).

3.1. Multi-Modal Feature

To extract multi-modal feature representations for animatable pedestrian generation, we process a sparse LiDAR point cloud and a single camera image through two separate backbone networks to obtain the 3D volumentric features and the 2D image features, respectively.

Volumetric feature representation: The volumetric feature backbone takes as input the LiDAR data and outputs a dense 3D voxel tensor, which encodes the 3D shape information given from the sparse point cloud. Specifically, we first transform the input point cloud $\mathcal{C}$ into a canonical coordinate frame. In particular, the point cloud is rotated along the yaw axis and translated along horizontal plane, such that its yaw angle and horizontal translation is normalized to be zero-centered. We then voxelize the normalized point cloud to a voxel grid. A 3D convolution network g_{vox} is then ex-

ploited to produce a volumetric feature tensor $g_{\text{vox}}(\mathcal{C})$. Our 3D convolution network has a U-Net architecture [48] due to its expressiveness and efficiency. The encoder contains 8 convolution layers and the decoder contains 6 convolution layers with skip connections. The final output feature has the same resolution as the input voxel grid. Please see supplementary for more details.

Image feature representation: To take advantage of the rich semantic and shape cues from images, we use a 2D convolution network to compute an image feature map $g_{\text{im}}(\mathcal{I})$ for image $\mathcal{I}$. The input image is ROI-cropped and masked by a given instance mask so that the target human is centered at the image with normalized size and the background is clean. Our 2D convolution network is a four-stacked hourglass network [40] with the modifications proposed in [49]. The final output feature has $1/4$ spatial resolution compared to the input image.

3.2. Point Feature Encoding

To better predict a continuous multi-dimensional neural field representation of shape, pose, and skinning we exploit contextual information encoded from the multi-modal input and the inductive biases from its 3D positions w.r.t. the observer. This point feature is an aggregation of the three features, namely the voxel, image and viewpoint features.

$$\phi(\mathbf{p}, \mathbf{x}) = [\phi_{\text{vox}}(\mathbf{p}, \mathcal{C}), \phi_{\text{im}}(\mathbf{p}, \mathcal{I}), \phi_{\text{view}}(\mathbf{p})] \quad (1)$$

We now describe these features in more detail.

LiDAR and Image Encoding: Given the input voxel feature $g_{\text{vox}}(\mathcal{C})$ and image feature $g_{\text{im}}(\mathcal{I})$, we compute the feature vector at an arbitrary continuous point $\mathbf{p}$ through interpolation. Specifically, we use trilinear interpolation to obtain corresponding 3D feature, denoted as

$\phi_{\text{vox}}(\mathbf{p}, \mathcal{C}) = \texttt{TrilinearSample}(\mathbf{p}, g_{\text{vox}}(\mathcal{C}))$; We also project the point onto the camera plane and retrieve 2D image feature through bilinear interpolation, denoted as $\phi_{\text{im}}(\mathbf{p}, \mathcal{I}) = \texttt{BilinearSample}(\pi(\mathbf{p}), g_{\text{im}}(\mathcal{I}))$, where π is the perspective projection function. Here we assume the calibration is known a priori.

Viewpoint Encoding: To provide the global context, we compute an additional global positional feature $\phi_{\text{view}}(\mathbf{p})$ for a given query point $\mathbf{p}$, expressing how $\mathbf{p}$ is viewed wrt the sensor viewing angle. The viewpoint feature $\phi_{\text{view}}(\mathbf{p})$ is defined as the inner product between sample point $\mathbf{p}$ and the unit camera ray $\frac{\mathbf{r}}{\|\mathbf{r}\|_2}$ passing the origin $\mathbf{c}$:

$$\phi_{\text{view}}(\mathbf{p}) = (\mathbf{p} - \mathbf{c}) \cdot \frac{\mathbf{r}}{\|\mathbf{r}\|_2} \qquad (2)$$

where the origin $\mathbf{c}$ is defined as the pelvis joint of human by default or LiDAR points centre when LiDAR is available. This feature is especially useful in areas where the LiDAR is missing, the query points on the same outgoing camera ray will have same pixel feature $\phi_{\text{im}}(\mathbf{p}, \mathcal{I})$, this viewpoint encoding can help to distinguish those sample points.

3.3. Neural S³ Field

In this section we describe how we predict the final output $f(\phi(\mathbf{p}, \mathbf{x}))$ for each query point $\mathbf{p}$. The final output forms a high-dimensional vector field in continuous 3D space, encoding the occupancy, joint position and skinning weight for an arbitrary given point. The input to the network f is the point encoding $\phi(\mathbf{p}, \mathbf{x})$ described in Eqn. 1. Our network consists of three 5-layer multi-layer perceptrons (MLPs), namely the occupancy net f_{occ}, the pose net f_{pose}, and the skinning net f_{skin}. We use three separate headers to improve the prediction capacity for each task without sharing weights, whereas their shared input feature still promotes consistency across each task's output. We do not share the weights among the modules to allow for better expressiveness.

Occupancy Net: Inspired by [49, 37], our occupancy head outputs a probability value $\mathbf{o} \in [0, 1]$ representing the probability of the point $\mathbf{p}$ being inside the human.

$$f_{\text{occ}}(\phi(\mathbf{p}, \mathbf{x})) \rightarrow \mathbf{o} \in [0, 1]$$

Such continuous representation is easier to train than implicit shape representations, such as the signed distance function, while maintaining expressive power.

Pose Net: The human skeleton is represented as a set of key-points and interconnected bones in 3D. To recover the underlying skeleton, the pose net maps each query point $\mathbf{p}$ into a M-dimensional probability vector $\mathbf{j}$ representing the likelihood that the query point belongs to each human joint

$$f_{\text{pose}}(\phi(\mathbf{p}, \mathbf{x})) \rightarrow \mathbf{j} \in [0, 1]^M$$
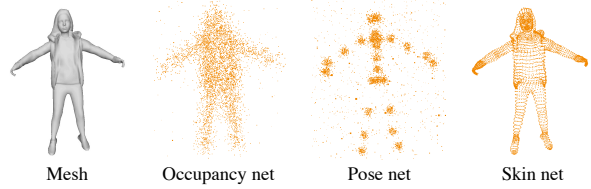
where M is the number of joints.

Figure 3: Sampled points to compute loss on each module.

Skinning Net: Skinning is the process that binds the vertices on the 3D mesh to each bone in the human skeleton. For instance, rotating the neck of a character should most likely influence the vertices on the head. The skinning weights describe the influence of each bone's rigid motion on a given vertex, and the values are normalized such that the sum of skinning weights on all bones equals to 1. To reconstruct the skinning weights, we learn a skinning function over the human surface, which maps each surface point on the human mesh into a skinning vector. Similar to the other two heads, our skinning net takes the point encoding $\phi(\mathbf{p}, \mathbf{x})$ as input and outputs a K-dimensional probability simplex,

$$f_{\text{skin}}(\phi(\mathbf{p}, \mathbf{x})) \rightarrow \mathbf{s} \in \Delta^K$$

with K the number of joints. In addition, we add a softmax layer after the fully-connected layers of skinning head to make the prediction sum to one.

3.4. Animatable Model Extraction

Suppose now we have learned the function $f(\phi(\mathbf{p}, \mathbf{x}))$. The next step is to extract an animatable human model from it. We first construct a dense grid of query points $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^N$ according to the desired resolution. For each query point $\mathbf{p}$, we evaluate the probaility of its volumetric occupancy $\mathbf{o}$ and body joint $\mathbf{j}$, creating a dense field of occupancy as well as joint heatmaps. Then we run the Marching Cubes algorithm [35] on the occupancy grid to generate the human mesh. We also locate all the human joints' positions by taking its most probable location over each joint's heatmap. Finally, for each vertex $\mathbf{v}$ of the reconstructed mesh, we compute the per-vertex skinning weight $\mathbf{s}$. Fig. 2 (right) depicts such animatable model extraction process.

3.5. Network Training

We learn our implicit functions parameterized by neural network weights through a dataset composed of pairs of sensory input and ground-truth animatable mesh. The networks parameters are trained by jointly minimizing the following loss function:

$$L = \lambda_{\text{occ}} L_{occ} + \lambda_{pose} L_{pose} + \lambda_{skin} L_{skin} \qquad (3)$$

Fig. 3 shows the sampled query points to train the three implicit functions. The parameters of all the backbone and head networks in $f(\phi(\mathbf{p}, \mathbf{x}))$ are learned jointly through

back-propagation over the sampled query points. We now describe the loss functions and query point sample strategy for each head in detail.

Occupancy Loss: This loss encodes the disagreement between the ground truth (GT) occupancy and the estimated occupancy probability. During training, we sample a batch of query points $\mathbf{p}_i \in \mathbb{R}^3, i \in 1, 2, \cdots N_{\text{occ}}$ from the continuous 3D space, and use the mean square error between the ground-truth occupancy o_i and predicted occupancy $f_{\text{occ}}(\phi(\mathbf{p}_i, \mathbf{x}))$ as occupancy loss function:

$$L_{\text{occ}} = \frac{1}{N_{\text{occ}}} \sum_{i=1}^{N_{\text{occ}}} \left(\mathbf{o}_i - f_{\text{occ}}(\phi(\mathbf{p}_i, \mathbf{x}))\right)^2 \qquad (4)$$

We follow [49] and sample near surface points and uniform points in the space as the query training points. For near surface points, we randomly sample points from the mesh surface, and perturb them with Gaussian noise. Such sampling strategy ensures training efficiency while still learning a high-definition surface boundary.

Pose Loss: We use the mean square error between the ground-truth key-point probability $\mathbf{j}_i$ and predicted joint probability $f_{\text{pose}}(\phi(\mathbf{p}_i, \mathbf{x}))$ as the pose loss:

$$L_{\text{pose}} = \frac{1}{N_{\text{pose}}} \sum_{i=1}^{N_{\text{pose}}} \left(\mathbf{j}_i - f_{\text{pose}}(\phi(\mathbf{p}_i, \mathbf{x}))\right)^2 \qquad (5)$$

where the GT key-point probability $\mathbf{j}_i$ is determined by computing the Gaussian function on the distance between query point $\mathbf{p}_i$ and GT key-point location. Unlike prior 2D and 3D pose estimation methods which learn a joint probability on a dense grid heatmap [40, 54], we sample the query points $\mathbf{p}_i$ from a Gaussian distributions centered at the GT key-point locations, and we augment them with uniformly sampled points in the space to ensure background points are also covered. Fig. 3 depicts an example of such sampling strategy.

Skinning Loss: For each queried point $\mathbf{p}_i$ during training, we also apply the mean square error between the ground-truth skinning weight $\mathbf{s}_i$ and predicted skinning weight $f_{\text{skin}}(\phi(\mathbf{p}_i, \mathbf{x}))$ as the loss function:

$$L_{\text{skin}} = \frac{1}{N_{\text{skin}}} \sum_{i=1}^{N_{\text{skin}}} \left(\mathbf{s}_i - f_{\text{skin}}(\phi(\mathbf{p}_i, \mathbf{x}))\right)^2 \qquad (6)$$

where N_{skin} is the number of query points.

4. Experimental Evaluation

In this section, we begin by introducing our experimental setting. We then compare our approach against several state-of-the-art methods in 3D shape reconstruction and present some ablation studies. Due to the page limits, we refer the readers to see the additional ablation studies in the supplementary material. We also showcase our method's ability to animate the reconstructed human shape to target poses.

4.1. Datasets

We train our algorithm on the RenderPeople [3] dataset, where ground-truth shapes are available. We evaluate our model on held out RenderPeople characters and animations as well as on a large-scale self-driving dataset containing pedestrians with large variations in pose, shape, clothing, under diverse environmental illuminations.

RenderPeople Dataset: We purchased 793 RenderPeople rigged characters [3] and then animate them with 39 different animations from Mixamo [2]. Our character set contains a wide array of pedestrians with different body shapes and clothing. We extend the diverse set of 32 animations from [33] with 7 additional animations that cover pedestrian behaviors, such as walking and running. We randomly select 3 frames per animation for each model, resulting in $777 \times 39 \times 3 = 90,909$ meshes for training, and $16 \times 39 \times 3 = 1,872$ meshes for evaluation. In addition, we use an extra 10 novel animations from Mixamo (result in $16 \times 10 \times 3 = 480$ meshes) to test the generalization ability of our model to unseen animations. To create realistic images, we use Blender Cycles engine [18] to render images. Characters and placed at a depth of 10 meters and view angle is uniformly distributed around yaw angle. We use ≈ 400 outdoor HDRI image from HDRI Haven [1] for environment lighting. In addition, we use Intel Embree ray tracer [60] to simulate LiDAR point cloud. Random ray dropping and disturbance are injected to push realism. We generate training and testing examples every 20 degrees and 120 degrees around the yaw-axis. Please see more details in supplementary.

Self-driving Dataset: We also conduct experiments on a real-world self-driving dataset, which contains diverse scenes across multiple metropolitan cities in North America. Our self-driving sensor platform has a global-shutter camera with a 112° horizontal FOV and a 64-beam spinning LiDAR. To create our dataset, we label 1.3k pedestrians that are within 6-25m of the ego-car with 3D box annotations and image-based instance annotations, resulting in over 1k evaluation examples.

4.2. Experimental Details

Implementation Details: We train our model on 512×512 RenderPeople images using 16 GPUs with a batch size of 64 for 14 epochs. We voxelize the LiDAR sweep to a voxel grid with shape $64 \times 64 \times 64$. We use RMSProp optimizer with initial learning rate of 1×10^{-3}, and then decay it by 10 at 10th epochs and 12th epochs. We sample

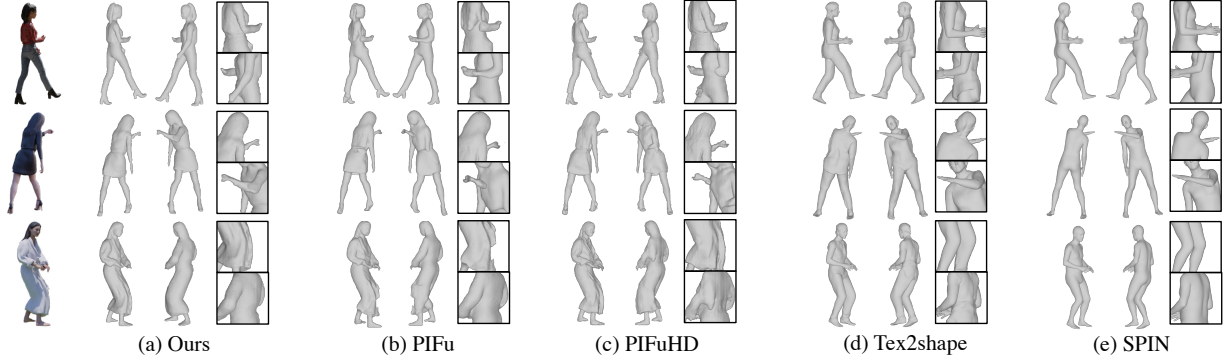(a) Ours   (b) PIFu   (c) PIFuHD   (d) Tex2shape   (e) SPIN

Figure 4: Reconstruction on RenderPeople dataset. In each cell, we show the shape in front, invisible, and zoomed-in views. "Ours" is with image + LiDAR as input.

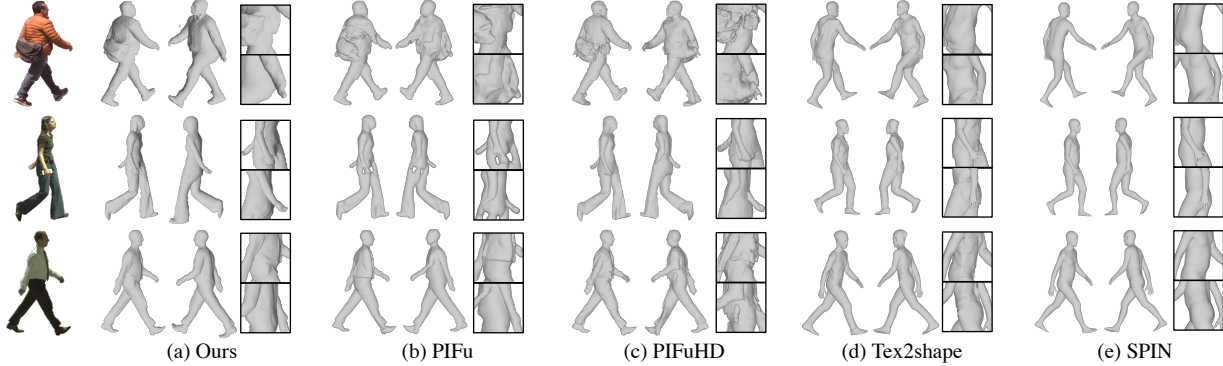(a) Ours   (b) PIFu   (c) PIFuHD   (d) Tex2shape   (e) SPIN

Figure 5: Reconstruction on real-world data. In each cell, we show the shape in front, invisible, and zoomed-in views. "Ours" is with image + LiDAR as input.

	Seen Poses			Unseen Poses	
	CD ↓	P2S↓	Norm↑	CD ↓	P2S↓
SPIN* [31]	3.25	2.97	0.76	3.26	2.97
PIFu*,† [49]	2.78	2.63	0.78	2.74	2.58
PIFuHD*,† [50]	2.47	2.44	0.79	2.41	2.38
PIFu [49]	1.03	1.05	0.88	1.61	1.70
Ours (img)	0.92	0.93	0.89	1.58	1.63
Ours (img+lidar)	**0.66**	**0.65**	**0.91**	**0.76**	**0.75**

Table 1: Quantitative comparisons on RenderPeople dataset. *: model pre-trained from other RenderPeople dataset; †: model using orthographic images. We report Chamfer and P2S error in cm for both seen and unseen poses, and the lower is better. For Normal consistency, the higher is better.

5,000 points for each module to compute the loss function during training. During inference, we evaluate the neural implicit fields to construct a 256^3 spatial resolution volume and perform Marching Cubes with iso-surface threshold at 0.5 to extract the output meshes.

Baselines: We compare our model with several state-of-the-art shape reconstruction approaches, namely PIFu [49], PIFuHD [50], Tex2Shape [5] and SPIN [31]. Due to the

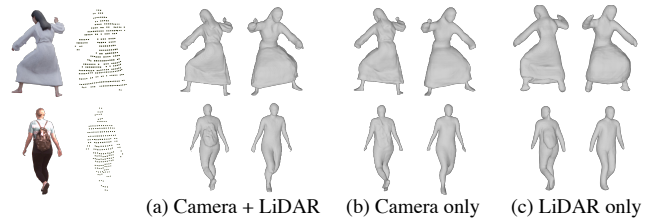(a) Camera + LiDAR   (b) Camera only   (c) LiDAR only

Figure 6: Reconstruction with different sensory inputs.

lack of training scripts or configurations, we directly use the officially released models that are trained on the authors' private RenderPeople datasets, which might differ from ours in terms of poses, characters, lighting and rendering parameters. Note that the pretrained model provided by PIFu [49] and PIFuHD [50] are trained on orthographic camera images. We found they do not transfer to perspective camera well. To ensure a fair comparison during testing, we render orthographic camera images as input for these models while keeping the remaining rendering settings constant, including character, action, camera pose, lighting, and rendering engine. There is also a global coordinate/pose shift between the shape predicted by the pre-trained models and our GT. We thus perform point-to-plane ICP [13] between the predicted and ground-truth shape before evaluation. We also

Sensor	Chamfer↓	P2S↓	Normal↑
Image	0.92	0.93	0.89
LiDAR	1.16	1.30	0.89
Image+LiDAR	**0.66**	**0.65**	**0.91**

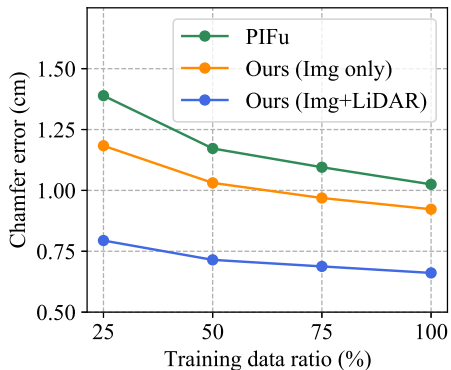Table 2: Ablation on multi-sensor feature.

Figure 7: Chamfer error as a function of training set scale.

scale SPIN [31]'s predicted meshes such that they are under the same scale as our GT meshes. In addition, we also train PIFU model from scratch on the same training set as ours.

Metrics: We evaluate the reconstruction performance of our model on three metrics: (1) an average Chamfer distance (cm) between our reconstructed mesh and the GT mesh, (2) an average point-to-surface (P2S) distance (cm) between the vertices on our reconstructed surface to the GT surface, and (3) a normal consistency measure, which is defined as the average of *normal accuracy* and *normal completeness*. We compute the normal accuracy by sampling 10,000 points from the predicted mesh surface, finding their closest points on the ground-truth mesh surface, and computing the cosine similarity between their normal directions. We compute the normal completeness by sampling 10,000 points from the GT mesh surface, finding closest points on the prediction surface, and computing the cosine similarity between their normal directions.

4.3. RenderPeople Dataset Results

State-of-the-art (SoTA) Comparison: We compare our model with SOTA on our test set (RenderPeople). We use * to denote pre-trained models on third-party datasets and † to represent the usage of orthographic camera images. As shown in Table 1, our method achieves the best performance in all metrics. Note that PIFu and PIFu*,† have a significant performance gap, as the RenderPeople datasets and data generation process may differ in terms of pose distribution, lighting, rendering engine and training dataset scale. The PIFu model is equivalent to our image-only baseline using depth feature as viewpoint encoding. Figure 4 shows a qualitative comparison. We show the result using our model

(image+LiDAR) by default in all figures. Our model captures better topological structures and fine details, especially on occluded parts.

Quantitative results on unseen animations: Prior works [33, 49, 4, 5] utilize a similar human pose distribution during training and testing. To test model generalization, we also evaluate reconstruction performance on models with 10 unseen animations. As shown in Table 1, both our image only method and multi-modal method outperform all the competing approaches in this setting as well. It is worth noting that testing on novel pose/actions in general will hurt the performance for the methods trained on our training set, especially for image-only approaches. This is because single-view 3D reconstruction is an ill-posed problem and there may be multiple solutions. Learning-based methods tend to pick solutions that are within the training set distribution.

Ablation on sensor type: As shown in Table 2, using all sensors achieves the best performance. Figure 6 shows qualitative comparison. While using sparse LiDAR alone captures global structure and human topology, the image provides additional information to refine the pose and shape details such as hair and clothing.

Ablation on dataset scale: We also ablate the human reconstruction performance as a function of the training set size. As shown in Figure 7, our approach performs well even when trained with only 25% of the data.

4.4. Self-driving Results

We qualitatively evaluate our method on self-driving data to reconstruct in-the-wild pedestrians in Figure 5. Our reconstructions have less artifacts and better capture global topology, while PIFu [49] and PIFuHD [50] tend to either miss regions or predict incorrect body parts. Our model is robust to light accessories although our training data does not include those accessories.

4.5. Animating the Reconstructed Characters

We now showcase how the reconstructed animatable 3D human model can be retargeted to new poses. Given an inferred 3D human mesh, skeleton and skinning weights, we use inverse kinematics (IK) to compute the transformation between our reconstructed skeleton and the target skeleton. We then re-animate the mesh using linear blend skinning (LBS) through the inferred skinning weights. Please see the supplementary for more details about our IK solver and the LBS model.

Quantitative evaluation: We introduce a new metric to measure the overall performance of our model, denoted as retarget error. Specifically, given a target human pose, we retarget our predicted human to generate the shape at this target pose, and we evaluate the chamfer distance (cm) between
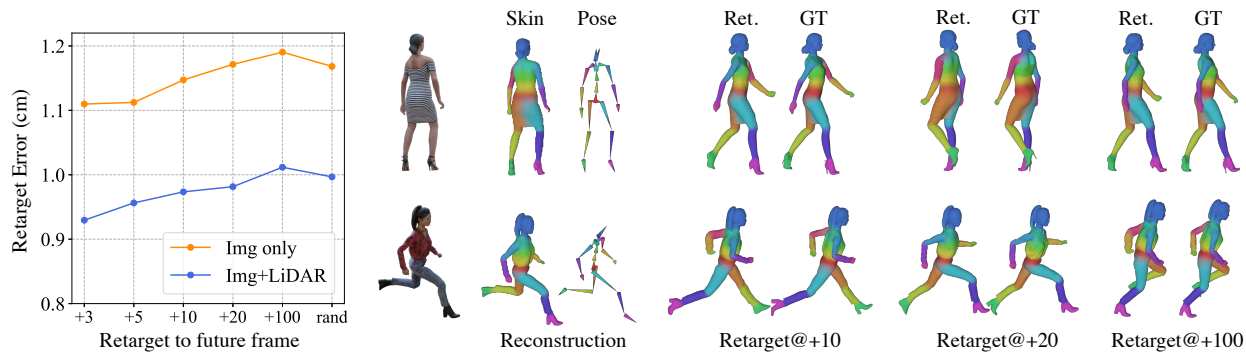
Figure 8: Reanimation results. Left: quantitative evaluation; Right: qualitative evaluation, from left to right: our predicted mesh, skeleton, mesh retarget to future (10, 20, 100) frames and GT mesh. In each cell, left is retarget shape, right is GT shape.
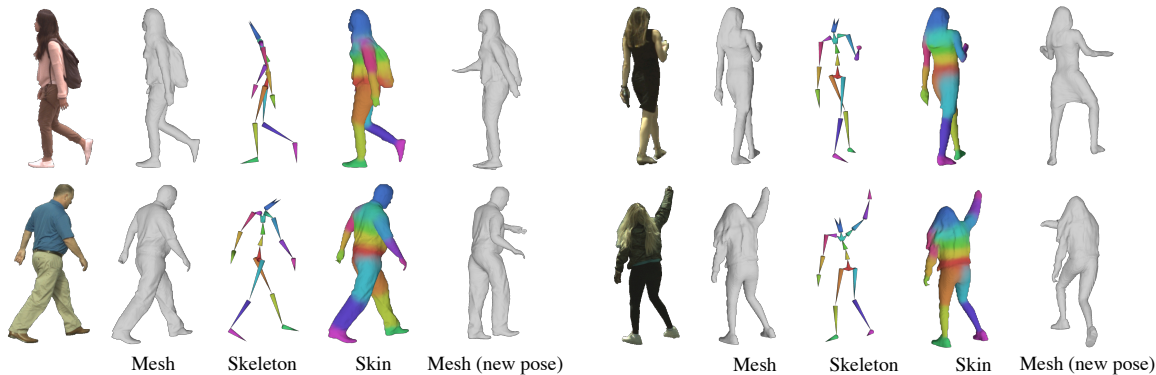

Figure 9: Our reconstructed mesh, skeleton and skinning weights from self driving data, which can be used to animate the mesh to novel poses.

our re-animated shape and GT shape at the target human pose. This retarget error measures the overall performance of surface reconstruction, pose estimation and re-animation of our model. We conduct the evaluation on our RenderPeople test set, and we report the retarget error on both future frame $(+3, +5, +10, +20, +100)$ and random frames. As shown in Figure 8 (left), the error increases when retargeting to further frames, and it saturates at about 100 future frames. To the best of our knowledge, we are the first to quantitative study the human re-animation performance. We hope this will result in further work in the field.

Qualitative evaluation: Figure 8 (right) shows a comparison of our retargeted mesh and GT mesh on RenderPeople dataset. We color the mesh with the skinning weights. We observe that the retargeting and skinning prediction are accurate, demonstrating the effectiveness of our model using a simple feed-forward network. Figure 9 shows the predicted mesh, skeleton, skinning weights visualizations, and the re-animated mesh in the target pose for examples in our self-driving dataset. The re-posed meshes look realistic and preserve local details.

Failure case analysis We also show the failure cases in Figure 10. Our shape reconstruction module will fail due

Figure 10: Visualizations of failure cases. Our shape model failed due to inaccurate segmentation, severe accessories. Our animation model failed due to self-contacts, *etc*.

to segmentation error; severe clothing and accessories, *etc*. This might because our training data does not include accessories. Our animation module will introduce artifacts in regions with self-contacts due to heavy clothing/accessories.

5. Conclusion

In this paper, we propose S^3, a novel deep animatble human shape reconstruction algorithm. Our network can take either a single image, a single LiDAR sweep, or both as input and predict a continuous multi-dimensional vector field in 3D space, representing occupancy, skeleton, and skinning weight. Our experimental results showcase that the proposed method achieves state-of-the-art performance in two challenging datasets. In addition, we demonstrate that we could generate 3D animated human character sequences using reconstructed 3D shapes from the proposed method.

References

[1] Hdri haven, https://hdrihaven.com, 2020. 5

[2] Mixamo, https://www.mixamo.com, 2020. 5

[3] Renderpeople, https://renderpeople.com/3d-rigged-people, 2020. 5

[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 *CVPR*, 2019. 1, 2, 7

[5] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *ICCV*, 2019. 1, 2, 6, 7

[6] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*. 2005. 1, 2

[7] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *CVPR*, 2020. 2

[8] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG)*, 2007. 1

[9] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. *arXiv*, 2020. 2

[10] 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 *ECCV*, 2016. 2

[11] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *PAMI*, 2019. 1

[12] Xin Chen, Chen Cao, Zehao Xue, and Wei Chu. Joint audio-video driven facial animation. In *ICASSP*, 2018. 1

[13] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 1992. 6

[14] KMG Cheung, Simon Baker, and Takeo Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *CVPR*, 2003. 1

[15] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *CVPR*, 2020. 2

[16] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Monocular expressive body regression through body-driven attention. In *ECCV*, 2020. 2

[17] CMU. Carnegie-mellon mocap database. 1

[18] Blender Online Community. *Blender - a 3D modelling and rendering package*, 2018. 5

[19] Olivier Dionne and Martin de Lasa. Geodesic voxel binding for production character meshes. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2013. 1

[20] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Conference on Robot Learning*, 2017. 1

[21] Tong He, John Collomosse, Hailin Jin, and Stefano Soatto. Geo-pifu: Geometry and pixel aligned implicit functions for single-view human reconstruction. *arXiv*, 2020. 1, 2

[22] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *CVPR*, 2020. 2

[23] Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. Bcnet: Learning body and cloth shape from a single image. In *ECCV*, 2020. 2

[24] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018. 2

[25] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 1, 2

[26] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *CVPR*, 2019. 2

[27] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. Skinning with dual quaternions. In *I3D*, 2007. 1

[28] Ladislav Kavan and Jiří Žára. Spherical blend skinning: a real-time deformation of articulated models. In *I3D*, 2005. 1

[29] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, 2020. 2

[30] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020. 1

[31] 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 *ICCV*, 2019. 2, 6, 7

[32] John P Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH*, 2000. 2

[33] Ruilong Li, Yuliang Xiu, Shunsuke Saito, Zeng Huang, Kyle Olszewski, and Hao Li. Monocular real-time volumetric performance capture. *arXiv*, 2020. 5, 7

[34] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 2015. 1, 2

[35] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 1987. 4

[36] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In *CVPR*, 2020. 2

[37] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 2, 4

[38] 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. *arXiv*, 2020. 2

[39] Aron Monszpart, Paul Guerrero, Duygu Ceylan, Ersin Yumer, and Niloy J Mitra. iMapper: interaction-guided scene map-

ping from monocular videos. *ACM Transactions on Graphics (TOG)*, 2019. 1

[40] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hour-glass networks for human pose estimation. In *ECCV*, 2016. 3, 5

[41] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *ICCV*, 2019. 1

[42] Jeong Joon Park, Peter Florence, Julian Straub, Richard New-combe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2

[43] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 2

[44] Georgios Pavlakos, Nikos Kolotouros, and Kostas Daniilidis. Texturepose: Supervising human mesh estimation with texture consistency. In *ICCV*, 2019. 2

[45] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *CVPR*, 2018. 2

[46] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J Black. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics (TOG)*, 2015. 2

[47] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 2017. 2

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 3

[49] 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 *ICCV*, 2019. 1, 2, 3, 4, 5, 6, 7

[50] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, 2020. 2, 6, 7

[51] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 2010. 1

[52] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NIPS*, 2019. 2

[53] Carsten Stoll, Juergen Gall, Edilson De Aguiar, Sebastian Thrun, and Christian Theobalt. Video-based reconstruction of animatable human characters. *ACM Transactions on Graphics (TOG)*, 2010. 1, 2

[54] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 5

[55] Yongbin Sun, Ziwei Liu, Yue Wang, and Sanjay E Sarma. Im2avatar: Colorful 3d reconstruction from a single image. *arXiv*, 2018. 1

[56] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017. 1

[57] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *NIPS*, 2017. 2

[58] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 1

[59] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 papers*. 2008. 1, 2

[60] Ingo Wald, Sven Woop, Carsten Benthin, Gregory S Johnson, and Manfred Ernst. Embree: a kernel framework for efficient cpu ray tracing. *ACM Transactions on Graphics (TOG)*, 2014. 5

[61] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. In *CVPR*, 2019. 1, 2

[62] Donglai Xiang, Fabian Prada, Chenglei Wu, and Jessica Hodgins. Monoclothcap: Towards temporally coherent clothing capture from monocular rgb video. *arXiv*, 2020. 2

[63] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NIPS*, 2019. 2

[64] Ze Yang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Recovering and simulating pedestrians in the wild. In *Conference on Robot Learning*, 2021. 1

[65] Ze Yang, Yinghao Xu, Han Xue, Zheng Zhang, Raquel Urtasun, Liwei Wang, Stephen Lin, and Han Hu. Dense reppoints: Representing visual objects with dense point sets. In *ECCV*, 2020. 2

[66] 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 *CVPR*, 2018. 1

[67] Haotian Zhang, Cristobal Sciutto, Maneesh Agrawala, and Kayvon Fatahalian. Vid2player: Controllable video sprites that behave and appear like professional tennis players. *arXiv*, 2020. 1

[68] Jason Y Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3d human-object spatial arrangements from a single image in the wild. In *ECCV*, 2020. 2

[69] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. *arXiv*, 2020. 2

[70] Hao Zhu, Xinxin Zuo, Sen Wang, Xun Cao, and Ruigang Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *CVPR*, 2019. 2

[71] Luyang Zhu, Konstantinos Rematas, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman. Reconstructing NBA players. In *ECCV*, 2020. 1