DIFu: Depth-Guided Implicit Function for Clothed Human Reconstruction

Dae-Young Song†[1,2], HeeKyung Lee[1], Jeongil Seo[1], Donghyeon Cho*[2]
[1]Electronics and Telecommunications Research Institute, Daejeon, South Korea
[2]Chungnam National University, Daejeon, South Korea
{eadyoung,lhk95,seoji}@etri.re.kr, cdh12242@gmail.com

Abstract

Recently, implicit function (IF)-based methods for clothed human reconstruction using a single image have received a lot of attention. Most existing methods rely on a 3D embedding branch using volume such as the skinned multi-person linear (SMPL) model, to compensate for the lack of information in a single image. Beyond the SMPL, which provides skinned parametric human 3D information, in this paper, we propose a new IF-based method, DIFu, that utilizes a projected depth prior containing textured and non-parametric human 3D information. In particular, DIFu consists of a generator, an occupancy prediction network, and a texture prediction network. The generator takes an RGB image of the human front-side as input, and hallucinates the human back-side image. After that, depth maps for front/back images are estimated and projected into 3D volume space. Finally, the occupancy prediction network extracts a pixel-aligned feature and a voxel-aligned feature through a 2D encoder and a 3D encoder, respectively, and estimates occupancy using these features. Note that voxel-aligned features are obtained from the projected depth maps, thus it can contain detailed 3D information such as hair and cloths. Also, colors of each query point are also estimated with the texture inference branch. The effectiveness of DIFu is demonstrated by comparing to recent IF-based models quantitatively and qualitatively.

1. Introduction

In order to implement virtual reality and an immersive metaverse environment, a method of reconstructing a realistic human avatar is an important technology. In particular, if there are methods that can create a complete 3D model with only a single view image without specialized devices such as 3D scanning, it will be highly useful in various fields such as education, video conference, and entertainment. Recently, there have been approaches to clothed human reconstruction using a single-view image based on the implicit function (IF) [1,3,9,10,12,13,22,34,35,43,53]. While IF-based methods have shown promising results thus far, their performance is limited in unobservable parts. Also, IF-based methods often produce over-smoothed results, particularly in intricate areas such as clothing and hair. Without proper conditions for occluded parts, clothed human reconstruction is still an open and highly ill-posed problem.

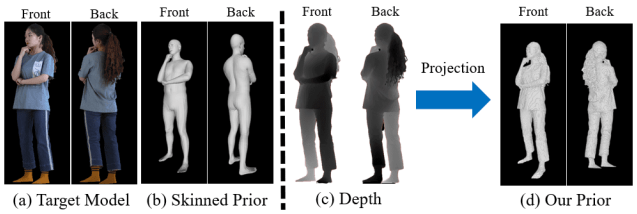

Figure 1. (a) Front/back color images. (b) Parametric model volume. (c) Depth maps. (d) Projected depth volume.

To overcome the aforementioned issue, there are several attempts using parametric models [17, 23, 30, 44] to provide geometric patterns of the human. Leveraging these benefits, Zheng *et al*. [53] proposed the parametric model-conditioned implicit representation (PaMIR). Using the skinned multi-person linear (SMPL) voxel from pretrained GCMR [19], PaMIR extracts 3D geometric features to overcome depth ambiguity. Also, Xiu *et al*. [43] proposed a method using the signed distance from the skinned model to the query points. Their approach helps approximate the distance from the skinned model to the target surface. While the skinned model can provide global and pose information to condition occluded parts, it may struggle to estimate surface that is far from the skin, such as long hair or skirts. As shown in Figure 1-(b), significant discrepancies exist between the detailed surface shape of the skinned model and the target model. Considering that the parametric model-based methods are trained by losses on the sampled

* Corresponding author.
† This work was done while the first author was pursuing his Master's degree at Chungnam National University.
Project page is at https://eadcat.github.io/DIFu

query points, the over-smoothing becomes more severe.

Therefore, in this paper, we propose a new IF-based method using projected depth maps. Specifically, our method uses a generator to make a back-side color image and front-/back-side depth maps from a front input image. Then, we project the depth maps into 3D volume space as shown in Figure 1-(d). All information, including RGB images, depth maps and projected depths are passed into the occupancy prediction network to predict the occupancy of each query point. The voxel-aligned features extracted from the 3D encoder in the occupancy prediction network are derived from projected depth maps rather than the SMPL model. As a result, they are more effective at conveying 3D information about the detailed surfaces of the target. To obtain the final 3D mesh, the marching cubes algorithm [24] is applied to these occupancies. Similar to the occupancy prediction network, we can estimate the colors of each query point via the texture inference network.

2. Related Works

2.1. Priors Knowledge for Novel Views

Generating prior knowledge for novel views, such as silhouettes [28], textures [20], and depths [37], can help overcome the lack of observations. Moreover, we further extend it by providing explicit guidance to the model through a projection method. Our experiments show the effectiveness of explicitly empowered implicit functions in clothed human reconstruction.

2.2. Statistical Mesh Reconstruction

Unlike implicit representations, parametric approaches based on statistical templates [17, 23, 30, 44] reconstruct skinned human 3D shapes and poses by regressing predefined parameters, as introduced in [5, 7, 18, 19, 32, 38–40, 45, 50]. Although parametric models are relatively fast and multi-human inferences are readily available, the skinned models do not take information such as hair and garments into account. Therefore, there is a gap between the skinned 3D model and the real 3D model.

2.3. Implicit Function-Based Reconstruction

Implicit representations [4, 21, 26, 29, 31, 46] were proposed for 3D reconstruction that can learn continuous spatial representation while having less complexity compared to voxel or point cloud-based methods. For human 3D reconstruction, the IF-based methods have successfully reconstructed a clothed human, as in [1, 9, 13, 34]. However, under a single image input setting, IF-based approaches suffer from depth ambiguity, occlusion, and over-smoothing. Therefore, there were several attempts to solve these problems using the parametric model [10, 53], surface normals [22, 35, 43], and additional modules [3]. Zheng *et al*. [53] proposed a method that utilizes the SMPL model to extract 3D voxel features. When provided with a single input image, the parametric model can serve as a robust indicator to infer the shape of a human, given its global shapes and poses. However, the parametric model has only 3D skinned human information, thus it has a disadvantage in detailed target surface inference. To mitigate this drawback, Xiu *et al*. [43] devised a branch that estimates the clothed surface normal from the skinned one. For each query point, the corresponding normal vector is used directly as one of the features. Further from those techniques, we propose a depth-guided implicit function (DIFu) method based on the projected depth prior. To this end, we project estimated depth maps into 3D volume space, then extract 3D features using the 3D encoder. Since the 3D encoder is based on 3D convolutional neural networks exploiting inductive bias [2, 48, 51], 3D features in DIFu convey useful information in the local region more than features of the clothed normal vector. As far as we know, DIFu is the first to successfully apply the depth projection for the IF-based clothed human reconstruction.

3. Method

In this section, we review a baseline method for clothed human reconstruction using an IF. Then, we describe the pipeline of the DIFu, including the generator, depth projection, and occupancy/texture prediction networks. Lastly, the training mechanisms are explained.

3.1. Baseline

For a continuous 3D query point $x \in \mathbb{R}^3$, an implicit function f estimates a probability p of occupancy under condition c as:

$$p = f(x, c), p \in [0, 1]. \tag{1}$$

In PIFu [34], the IF for human reconstruction is defined as:

$$p = f(X(F_i, \pi(x)), z(x)), \tag{2}$$

where F_i denotes the feature map from a 2D encoder, $\pi(x)$ the 2D projection of x on the F_i, $z(x)$ the depth value of x in the weak-perspective camera space, and X the bilinear sampling interpolation to sample F_i at $\pi(x)$. Furthermore, Zheng *et al*. [53] add a 3D branch to condition the f instead of $z(x)$ as:

$$p = f(X(F_i, \pi(x)), X(F_v, x)), \tag{3}$$

where F_v means volume feature map encoded by a 3D encoder using SMPL volume $\mathbb{V}_O$. To extract F_v from a single view RGB input, PaMIR [53] utilizes pre-trained GCMR [19]. In DIFu, we adopt the architecture of PaMIR as our baseline for the IF but $\mathbb{V}_O$ is substituted with the projected depth to obtain 3D voxel-aligned features. Compared

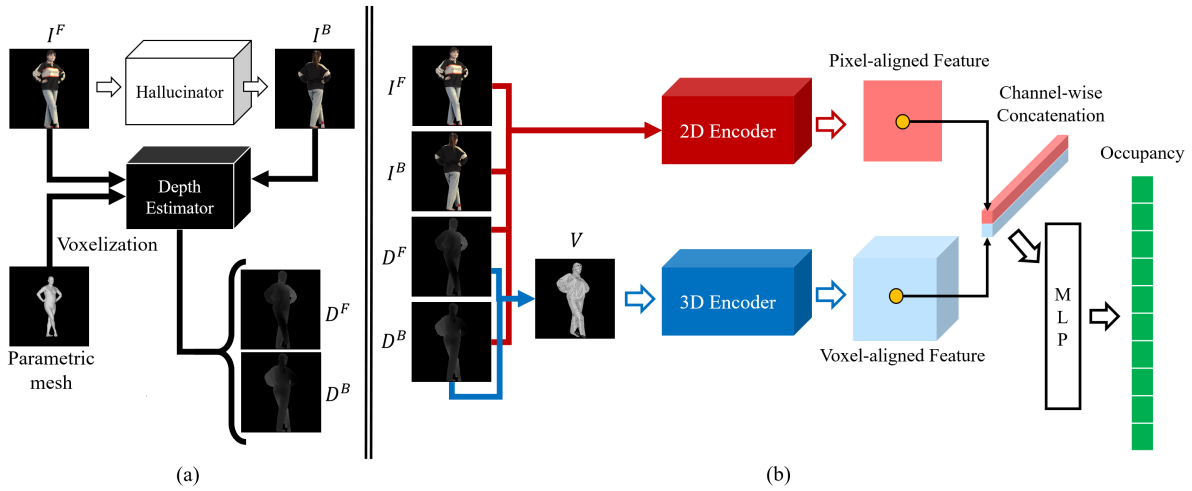

Figure 2. An Overview of our method. (a) First, the hallucinator generates a back-side image I^B using I^F. The depth estimator receives I^F and I^B and estimates a front depth map D^F and a back depth map D^B simultaneously. (b) With I^F, I^B, D^F, and D^B, the 2D encoder extracts a feature map to be transformed into the pixel-aligned feature. Meanwhile, D^F and D^B are projected to form a depth volume V. A 3D feature map is extracted by the 3D encoder from V, and aligned to the voxel-aligned feature. Both aligned features are concatenated in the channel axis and used by MLPs to estimate the final occupancy vector for given query points.

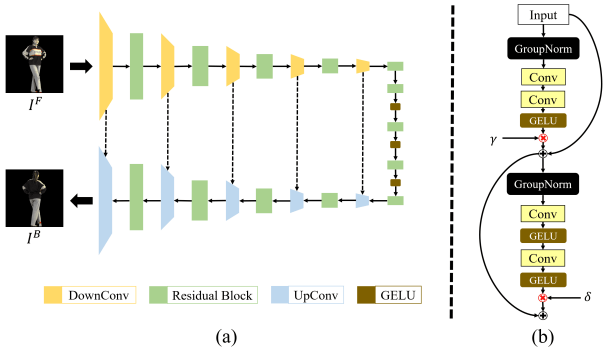

Figure 3. (a) The architecture of our hallucinator. (b) The structure of the residual block in (a). γ and δ are learnable parameters.

to $\mathbb{V}_O$, the projected depth provides detailed information about the surface. Therefore, we discuss how to estimate depth maps, and to create a 3D volume by projecting them into 3D space in Section 3.2 and Section 3.3.

3.2. Generator

Our generator consists of a hallucinator and a depth estimator. As shown in Figure 2-(a), the goal of the generator is to make a back-side image as well as depth maps for both sides. Then, the outputs of the generator are used as inputs to the following networks as shown in Figure 2-(b).

Hallucinator. Inspired by previous studies that show effects of the hallucination on human reconstruction [22, 35, 43], we design a hallucinator f_H that generates a back-side image I^B using a front-side image I^F as the follows:

$$I^B = f_H(I^F). \tag{4}$$

As shown in Figure 3-(a), f_H is based on U-Net [33]. Unlike pix2pix [15], we use GELU [11] instead of ReLU [27] for the faster learning and stochastic representation as illustrated in Figure 3-(b). Also, we adopt group normalizations [42] rather than batch normalizations [14] for robust synthesis.

Depth Estimator. The proposed depth estimator consists of an SMPL voxel encoder, an U-Net estimator, and a scale regressor. The SMPL voxel encoder f_S takes the estimated SMPL volume $\mathbb{V}_O$ and produces SMPL voxel features as:

$$F_s = f_S(\mathbb{V}_O), \tag{5}$$

where f_S is composed of 2D depthwise convolutional layers [6]. In other words, 2D convolution is applied to each depth layer of the 3D volume. Meanwhile, the U-Net consists of an encoder $f_{D_{enc}}$ and a decoder $f_{D_{dec}}$. The $f_{D_{enc}}$ takes I^F and I^B as input and extracts features as follows:

$$F_d = f_{D_{enc}}(I^F, I^B). \tag{6}$$

Then, $f_{D_{dec}}$ takes F_d and F_s for the depth prediction as:

$$\bar{D}^F, \bar{D}^B = f_{D_{dec}}(F_d, F_s), \tag{7}$$

where $\bar{D}^F$ and $\bar{D}^B$ are depths of the front and back human images, respectively. Since predicted depth maps suffer from scale ambiguity, we design a scale regressor f_λ as:

$$\lambda = f_\lambda(F_d^4, F_s^{fin}), \tag{8}$$

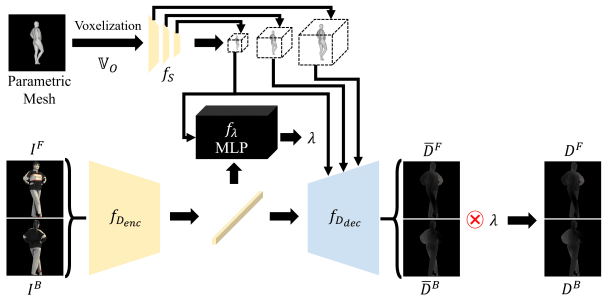

Figure 4. An illustration of our depth estimator. Our depth estimator consists of an U-Net f_D, a voxel encoder f_S, and a scale regressor f_λ.

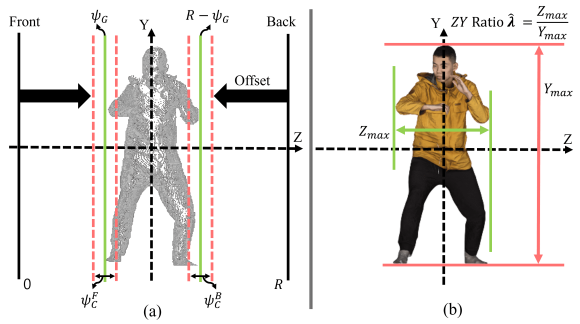

Figure 5. Side view descriptions of depth map projection. (a) Visualization of the projection with offsets. (b) Operation of $\hat{\lambda}$ at the rendering stage.

where λ is the scale factor, while F_d^4 and F_s^{fin} refer to the intermediate feature of the fourth layer in $f_{D_{enc}}$ and the feature from the last layer in f_S, respectively. With λ, the scale of $\bar{D}^F$ and $\bar{D}^B$ are compensated as:

$$D^F = \lambda \bar{D}^F, D^B = \lambda \bar{D}^B. \tag{9}$$

The illustration of the depth estimator is in Figure 4 and the effect of λ is shown in the supplementary materials.

3.3. Depth Projection

The estimated D^F and D^B need to be projected into a 3D space before being passed to the 3D encoder of the occupancy prediction network. Let us denote V^F and V^B as 3D volumes constructed by D^F and D^B. Note that D^F and D^B range from 0 to 1. The height, width, and depth of V^F and V^B are H, W, and R, respectively. Based on D^F and D^B, we can make V^F and V^B as follows:

$$V^F_{i,j,k} = \begin{cases} 1 & \text{if } k = R(D^F(i,j) + \psi^F), \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

$$V^B_{i,j,k} = \begin{cases} 1 & \text{if } k = R(1 - (D^B(i,j) + \psi^B)), \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where ψ^F and ψ^B are offsets for the projections of D^F and D^B, respectively. As shown in Figure 5-(a), V^F and V^B are properly gathered in the 3D space by using these offset values. Specifically, ψ^F and ψ^B are computed as:

$$\psi^F = \psi_G - \psi_C^F, \quad \psi^B = \psi_G - \psi_C^B, \tag{12}$$

where ψ_G is a global offset to put query points into the center of 3D volumes while ψ_C^F and ψ_C^B are compensation offsets for front and back volumes, respectively. The values of ψ_G, ψ_C^F, and ψ_C^B are computed as follows. Let D^F_{max} and D^B_{max} be the maximum depth values of D^F and D^B. Then, we can obtain ψ_G as follows:

$$\psi_G = \frac{1}{2}(1 - \max(D^F_{max}, D^B_{max})). \tag{13}$$

Since ψ_G is computed based on the higher maximum value of either front or back depth maps, the depth map with lower maximum values requires an compensation offset as:

$$\psi_C^F = \begin{cases} \psi_N & \text{if } D^F_{max} \geq D^B_{max}, \\ \frac{1}{2}|D^F_{max} - D^B_{max}| & \text{otherwise,} \end{cases} \tag{14}$$

$$\psi_C^B = \begin{cases} \psi_N & \text{if } D^F_{max} < D^B_{max}, \\ \frac{1}{2}|D^F_{max} - D^B_{max}| & \text{otherwise,} \end{cases} \tag{15}$$

where ψ_N is the random noise offset for data augmentation during the depth projection process. Note that ψ_N is changed randomly in the training stage, and fixed to zero in the test stage. Finally, based on (10) and (11), the volume of projected depth using D^F and D^B can be computed as:

$$V = V^F \cup V^B, \tag{16}$$

where $\cup$ is a union operator.

3.4. Occupancy Prediction Network

As shown in Figure 2, the occupancy prediction network consists of a 2D encoder, a 3D encoder, and MLPs. The 2D encoder f_{2d} takes I^F, I^B, D^F, and D^B as inputs and produces 2D feature maps as follows:

$$F_{2d} = f_{2d}(I^F, I^B, D^F, D^B). \tag{17}$$

The 3D encoder f_{3d} utilizes the projected depth volume in (16) and extracts 3D feature maps as:

$$F_{3d} = f_{3d}(V). \tag{18}$$

After that, F_{2d} and F_{3d} are passed into the geometry predictor f_G based on MLP layers to estimate the occupancy of each 3D query point as follows:

$$p = f_G(X(F_{2d}, \pi(x)), X(F_{3d}, x)). \tag{19}$$

Note that (19) is the same with (3) except for the input features of the implicit function. Using the marching cubes algorithm [24], the output occupancy p is converted to mesh.

3.5. Texture Prediction Network

Similar to previous IF-based methods such as PaMIR [53], the proposed DIFu can also predict textures using an additional texture branch. With F_{2d} in (17), the texture inference branch is defined as follows:

$$F_{a2d} = X(F_{2d}, \pi(x)) \oplus X(F_{c2d}, \pi(x)), \tag{20}$$

$$f_C(F_{a2d}, X(F_{c3d}, x)) \rightarrow (C(p)_t, \alpha, \beta), \tag{21}$$

where $\oplus$ is a concatenation operation, and F_{c2d} and F_{c3d} are 2D and 3D features from the texture branch, respectively. Note that F_{c3d} can be extracted by projecting color images pixel-wisely using depth maps. In addition, C_t is a predicted color value for the query point and α and β are two weight balancing factors. Since there are an input front and hallucinated back RGB images, we can obtain color values from inputs for the query point x as:

$$C(p)_\beta = \beta X(I^F, \pi(x)) + (1-\beta)X(I^B, \pi(x)) \tag{22}$$

Then, we compute the final color value as follows:

$$C(p) = \alpha C(p)_\beta + (1-\alpha)C(p)_t. \tag{23}$$

That is, our texture prediction network utilizes I^F and I^B as well as C_t, to make a finer estimate of color values. Detailed architectures are described in the supplementary materials.

3.6. Training Mechanisms

GT Data Preparation. GT occupancy label $\hat{p}$ is sampled from the GT scan, and utilized for supervising the occupancy prediction. Then, we render pairs of the front/back RGB images and the front/back depth maps from the GT scans and GT texture maps. Note that the front RGB image I^F is used as the input for our network while the back RGB image $\hat{I}^B$ is utilized as a GT for the hallucinator. Also, the rendered front/back depth maps are used for training the depth estimator. Because of scale ambiguity of the depth maps, we compensate them as follows. First, the range of depth values are normalized from 0 to 1. Then, we multiply a scale factor $\hat{\lambda}$ to obtain the GT depth maps $\hat{D}^F$ and $\hat{D}^B$. The $\hat{\lambda}$ is the ratio of depth to height for each scan as:

$$\hat{\lambda} = \frac{Z_{max}}{Y_{max}}, \tag{24}$$

where Z_{max} and Y_{max} are the human z-axis thickness and y-axis height, respectively, computed using a 3D clothed human scan as shown in Figure 5-(b). Typically, Z_{max} is smaller than Y_{max}, making $\hat{\lambda} \leq 1$. The final rendered depth maps include adjusted values that match the height-to-thickness ratio of the GT human scan.

Loss for the Hallucinator. For training the hallucinator, we utilize smoothed ℓ_1 loss [8] and perceptual loss [16] as:

$$\mathcal{L}_1 = \begin{cases} \mathbb{E}[0.5(\hat{I}^B - I^B)^2] & \text{if } |\hat{I}^B - I^B| < 1, \\ \mathbb{E}[|\hat{I}^B - I^B| - 0.5] & \text{otherwise,} \end{cases} \tag{25}$$

$$\mathcal{L}_p = \mathbb{E}[\sum_{j=1}^{n} |\phi_j(\hat{I}^B) - \phi_j(I^B)|], \tag{26}$$

where $\hat{I}^B$ and I^B denote the GT and hallucinated back-side images, respectively. Also, $\phi_j(\cdot)$ the activations of the jth layer of pre-trained VGG-19 [36]. For the generalization of unseen data, we also use the adversarial loss from LSGAN [25] with a patch discriminator f_{PD} as follows:

$$\begin{aligned} \min_{f_{PD}} \mathcal{L}_{adv}(f_{PD}) = &\frac{1}{2}\mathbb{E}[(f_{PD}(\hat{I}^B) - b)^2] \\ &+ \frac{1}{2}\mathbb{E}[(f_{PD}(f_H(I^F)) - a)^2], \end{aligned} \tag{27}$$

$$\min_{f_H} \mathcal{L}_{adv}(f_H) = \frac{1}{2}\mathbb{E}[(f_{PD}(f_H(I^F)) - c)^2], \tag{28}$$

where a and b are the labels for fake and real data, and c is the value that f_H wants f_{PD} to believe for fake data, respectively. Through experiments, we find that it is better to use the hallucinator trained without adversarial loss when the occupancy prediction network is trained. In contrast, the hallucinator trained with adversarial loss achieves better performance during the inference stage. Relevant discussion is covered in detail in Section 4.3.

Loss for the Depth Estimator. To supervise our depth estimator, we adopt the depth regression loss as follows:

$$\mathcal{L}_{reg} = \mathbb{E}[(\log\hat{D} - \log D)], \tag{29}$$

where $\hat{D} = \{\hat{D}^F, \hat{D}^B\}$ and $D = \{D^F, D^B\}$. Note that $\hat{D}$ and D are the GT and predicted depths, respectively. f_λ is also trained with regression loss as follows:

$$\mathcal{L}_{scale} = \mathbb{E}[(\hat{\lambda} - \lambda)^2]. \tag{30}$$

Therefore, total loss for $\mathcal{L}_{depth}$ is defined as follows:

$$\mathcal{L}_{depth} = \mathcal{L}_{reg} + \eta\mathcal{L}_{scale}, \tag{31}$$

where η is a weight which is set to 50 in our experiment.

Training Occupancy Prediction Network. To train occupancy prediction, we sample n 3D points and compare the output in (19) to GT via the mean squared error as follows:

$$\mathcal{L}_{occ} = \frac{1}{n}\sum_{i=1}^{n}((\hat{p}_i - p_i)^2). \tag{32}$$

In addition, as mentioned in Section 3.3, we make ψ_N to be a random number between $-\psi_C$ and ψ_C during the training stage. With this augmentation, f_{3d} in (18) adaptively encodes V even the depths are somewhat inaccurate. ψ_N is clipped by R/v, where v is set to 15 in our experiment.

Training Texture Prediction Network. For the texture prediction network, we adopt schemes of PaMIR [53] as:

$$\mathcal{L}_{clr} = \frac{1}{n}\sum_{i=1}^{n}(|\hat{C}(p_i) - C(p_i)| + |\hat{C}(p_i) - C(p_i)_t|), \tag{33}$$

Figure 6. Qualitative Results for unseen data. (a) RGB inputs. (b) Ground-truth mesh. (c) PIFu. (d) PaMIR. (e) ICON. (f) Ours.

	THuman2.0					BUFF				
Models	P2S ↓	Chamfer ↓	Normal ↓	MSE ↓	LPIPS ↓	P2S ↓	Chamfer ↓	Normal ↓	MSE ↓	LPIPS ↓
PIFu [34]	4.629	4.220	0.164	0.102	0.145	5.349	4.642	0.186	0.077	0.162
PaMIR [53]	4.071	4.080	0.150	0.098	0.141	5.010	4.620	0.165	0.038	0.156
ICON [43]	4.595	4.537	0.170	0.102	0.156	3.621	3.634	0.151	**0.035**	0.149
Ours	**2.992**	**2.952**	**0.119**	**0.082**	**0.124**	**3.375**	**3.318**	**0.138**	**0.035**	**0.138**

Table 1. Quantitative result of unseen datasets from THuman2.0 and BUFF. The unit of 3D distance is cm. **bold**: best.

where $\hat{C}(p_i)$ is the GT color value while $C(p)_t$ and $C(p)$ are the predictions of the texture network and the final color value, respectively. To exploit colors from both I^F and I^B as much as possible, an alpha loss $\mathcal{L}_\alpha$ is included as follows:

$$\mathcal{L}_\alpha = -\mathbb{E}[\log(\alpha)]. \tag{34}$$

The total loss of the texture network is defined as:

$$\mathcal{L}_{tex} = \mathcal{L}_{clr} + \mathcal{L}_\alpha. \tag{35}$$

4. Experiments

4.1. Implementation Details

We evaluate DIFu with state-of-the-art methods: PIFu [34], PaMIR [53], and ICON [43]. For fair comparisons, we reproduce all methods under the same implemental environment. We adopt the THuman2.0 [47] dataset that includes 526 high-quality 3D clothed human scans and GT SMPL [23] parameters. We use 495 scans for training and the rest for evaluation. To verify the generality of DIFu, we perform further experiments on 143 human scans of the BUFF [49] dataset. All 3D scans are rendered by the OpenGL script at every degree. Also, front images for evaluation are rendered from yaw angles of [0, 90, 180, 270] degrees. For all models requiring SMPL, we utilize pre-trained GCMR [19] to predict the parameters.

4.2. Evaluation

As evaluation metrics for the reconstructed 3D mesh, we utilize point-to-surface (P2S) distance, Chamfer distance, and re-projection errors with GT mesh. In addition, we use mean squared error (MSE) and LPIPS [52] to check the performance of the texture inference. Since there is no texture branch in ICON, we implement it based on those of PaMIR.

As reported in Table 1, DIFu consistently outperforms other existing methods with large margins in THuman2.0. Even in the BUFF dataset, the performance gap with other models is relatively small, but DIFu still achieves the best performance. In addition, as shown in Figure 6, our model produces more plausible results. In particular, the shape of

		THuman2.0					BUFF				
Training	Inference	P2S ↓	Chamfer ↓	Normal ↓	MSE ↓	LPIPS ↓	P2S ↓	Chamfer ↓	Normal ↓	MSE ↓	LPIPS ↓
w/o L_{adv}	w/o L_{adv}	3.331	3.282	0.129	0.087	0.129	3.465	3.399	**0.137**	**0.034**	**0.138**
w/o L_{adv}	w L_{adv}	**2.992**	**2.952**	**0.119**	**0.082**	**0.124**	**3.374**	**3.318**	0.138	0.035	**0.138**
w L_{adv}	w/o L_{adv}	3.484	3.479	0.134	0.090	0.130	3.525	3.540	0.139	0.037	0.139
w L_{adv}	w L_{adv}	3.118	3.107	0.124	0.087	0.125	3.500	3.507	0.143	0.038	0.140

Table 2. An ablation result of L_{adv} and data augmentation. The unit of 3D distance is cm. **bold**: best.

Figure 7. Qualitative results for in-the-wild images. (a) Inputs. (b) PIFu. (c) PaMIR. (d) ICON. (e) Ours.

unobservable parts is generated plausibly. We guess that this is due to the effect of the hallucinator that creates the back-side image. Moreover, we test DIFu with images from the internet. As shown in Figure 7, our method preserves information well that is far from the skin such as hair and clothes. Also, the shape and color of the back-side look most realistic, and robust to geometric damage. Additional inference results are in the supplementary materials.

4.3. Ablation Studies

Hallucinator for Training. In terms of the performance of DIFu, we test the effect of L_{adv} on f_H. When L_{adv} is applied, we also utilize data augmentation for further generalization. Specifically, we use flip, gamma correction, brightness adjustment, and rotation up to 45 degrees. As reported in Table 2, DIFu performs best when we use f_H trained without L_{adv} in the training phase and adopt f_H trained with L_{adv} in the test phase. We guess that there are many unseen samples in the test stage, thus training f_H with L_{adv}, which is advantageous for generalization, has better performance. Conversely, in the training phase, f_H trained with L_{adv} seems to confuse the prediction networks resulting in over-smoothing.

Hallucinator Comparisons. To validate the effectiveness of the proposed hallucinator in Section 3.2, we conduct experiments using pix2pix [15] and pix2pixHD [41] as the hallucinator. Note that the pix2pixHD is a promising model that is employed in PIFuHD [35] and ICON [43] to translate front/back surface normals from a front image. As reported in Table 3, our architecture consistently shows better quantitative performance than others. In addition, our hallucinator trained with L_{adv} generates plausible hallucinations as shown in Figure 8-(g). Comparing Figure 8-(f) and Figure 8-(g), applying data augmentation achieves better performance on unseen data. Inspired by [43], we test a hallucinator additionally receiving front/back normal maps. However, it does not show particularly good qualitative results as shown in Figure 8-(h) and thus is not our final model. Note that the pix2pixHD in the experiment has 180M parameters while ours has only 55M.

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 8. Hallucinator comparisons for unseen data. (a) Front side RGB inputs. (b) Back side ground truth images. (c) Pix2Pix†. (d) Pix2PixHD†. (e) Ours w/o L_{adv}. (f) Ours. (g) Ours†. (h) Ours w SMPL F/B normal†. † denotes the data augmentation is applied.

Metrics	L1 (↓)	LPIPS (↓)	SSIM (↑)
Models	THuman2.0		
pix2pix†	0.0157	0.0663	0.9175
pix2pixHD†	0.0156	0.0615	0.9272
Ours w/o L_{adv}	**0.0117**	0.0609	**0.9410**
Ours	0.0127	0.0627	0.9252
Ours†	0.0128	**0.0588**	0.9295
Ours w SMPL F/B Normal†	0.0132	0.0613	0.9283
Models	BUFF		
pix2pix†	0.0123	0.0621	0.9376
pix2pixHD†	0.0119	0.0604	0.9421
Ours w/o L_{adv}	**0.0101**	0.0590	**0.9512**
Ours	0.0115	0.0616	0.9369
Ours†	0.0108	0.0568	0.9448
Ours w SMPL F/B Normal†	0.0106	**0.0567**	0.9448

Table 3. Comparisons for the hallucinator on THuman2.0 and BUFF. For pix2pix and pix2pixHD, L_{adv} is applied. † denotes the augmentation is applied. **bold**: best. underline: second best.

Metrics	P2S	Chamfer	Normal	MSE	LPIPS
w/o f_S	3.214	3.146	0.124	0.084	0.126
Ours	**2.992**	**2.952**	**0.119**	**0.082**	**0.124**

Table 4. A comparison with/without SMPL voxel embedding of f_S. The unit of 3D distance is cm. **bold**: best.

Depthwise Embedding. To check the effects of f_S, we perform ablation studies depending on whether f_S is included. As reported in Table 4, f_S helps generalize the depth estimator for unseen data.

5. Conclusion

In this paper, we have proposed a DIFu for clothed human 3D reconstruction, which consists of a generator, an occupancy prediction network, and a texture inference network. Specifically, the generator network is composed of the hallucinator and the depth estimator. The hallucinator takes a front-side RGB image as an input, then creates a back-side RGB image, and the depth estimator predicts depth maps for both front-/back-side images. Then, predicted depth maps are projected into 3D volume space and passed to the occupancy prediction network for estimating the occupancy of each query point. In addition, the texture inference network can estimate the color. Finally, with the occupancy values for all query points, we can generate 3D human mesh through the marching cubes algorithm.

Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2018-0-00207, Immersive Media Research Laboratory), and Korea Creative Content Agency(KOCCA) grant funded by the Korea government(MCST) (No. R2021040122, Development of Artificial Intelligence-based Photo/Painting Harmonization and Content Expansion Technology)

References

[1] Thiemo Alldieck, Mihai Zanfir, and Cristian Sminchisescu. Photorealistic monocular 3d reconstruction of humans wearing clothing. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 1506–1515, 2022. 1, 2

[2] Peter W Battaglia et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 2

[3] Yukang Cao, Guanying Chen, Kai Han, Wenqi Yang, and Kwan-Yee K Wong. Jiff: Jointly-aligned implicit face function for high quality single view clothed human reconstruction. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 2729–2739, 2022. 1, 2

[4] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 5939–5948, 2019. 2

[5] Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 1964–1973, 2021. 2

[6] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 1251–1258, 2017. 3

[7] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Monocular expressive body regression through body-driven attention. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 20–40. Springer, 2020. 2

[8] Ross Girshick. Fast r-cnn. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 1440–1448, 2015. 5

[9] Tong He, John Collomosse, Hailin Jin, and Stefano Soatto. Geo-pifu: Geometry and pixel aligned implicit functions for single-view human reconstruction. *Proc. of Neural Information Processing Systems (NeurIPS)*, 33:9276–9287, 2020. 1, 2

[10] Tong He, Yuanlu Xu, Shunsuke Saito, Stefano Soatto, and Tony Tung. Arch++: Animation-ready clothed human reconstruction revisited. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 11046–11056, 2021. 1, 2

[11] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3

[12] Yang Hong, Juyong Zhang, Boyi Jiang, Yudong Guo, Ligang Liu, and Hujun Bao. Stereopifu: Depth aware clothed human digitization via stereo vision. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 535–545, 2021. 1

[13] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 3093–3102, 2020. 1, 2

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of Int'l Conf. on Machine Learning (ICML)*, pages 448–456. PMLR, 2015. 3

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017. 3, 7

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 694–711. Springer, 2016. 5

[17] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 8320–8329, 2018. 1, 2

[18] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 7122–7131, 2018. 2

[19] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 4501–4510, 2019. 1, 2, 6

[20] Verica Lazova, Eldar Insafutdinov, and Gerard Pons-Moll. 360-degree textures of people in clothing from a single image. In *2019 International Conference on 3D Vision (3DV)*, pages 643–653. IEEE, 2019. 2

[21] Tianyang Li, Xin Wen, Yu-Shen Liu, Hua Su, and Zhizhong Han. Learning deep implicit functions for 3d shapes with dynamic code clouds. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 12840–12850, 2022. 2

[22] Kennard Yanting Chan; Guosheng Lin; Haiyu Zhao; Weisi Lin. Integratedpifu: Integrated pixel aligned implicit function for single-view human reconstruction. In *Proc. of European Conf. on Computer Vision (ECCV)*. Springer, 2022. 1, 2, 3

[23] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Trans. on Graph. (ToG)*, 34(6):1–16, 2015. 1, 2, 6

[24] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Proc. of ACM SIGGRAPH*, 21(4):163–169, 1987. 2, 4

[25] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 2794–2802, 2017. 5

[26] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 4460–4470, 2019. 2

[27] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. of Int'l Conf. on Machine Learning (ICML)*, 2010. 3

[28] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 4480–4490, 2019. 2

[29] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 2

[30] 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 *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 1, 2

[31] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 523–540. Springer, 2020. 2

[32] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. *arXiv preprint arXiv:2008.08324*, 2020. 2

[33] 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*, pages 234–241. Springer, 2015. 3

[34] 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 *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 2304–2314, 2019. 1, 2, 6

[35] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 84–93, 2020. 1, 2, 3, 7

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[37] David Smith, Matthew Loper, Xiaochen Hu, Paris Mavroidis, and Javier Romero. Facsimile: Fast and accurate scans from an image in less than a second. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 5330–5339, 2019. 2

[38] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei. Monocular, one-stage, regression of multiple 3d people. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 11179–11188, 2021. 2

[39] Yu Sun, Wu Liu, Qian Bao, Yili Fu, Tao Mei, and Michael J Black. Putting people in their place: Monocular regression of 3d people in depth. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 13243–13252, 2022. 2

[40] Ziniu Wan, Zhengjia Li, Maoqing Tian, Jianbo Liu, Shuai Yi, and Hongsheng Li. Encoder-decoder with multi-level attention for 3d human shape and pose estimation. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 13033–13042, 2021. 2

[41] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 8798–8807, 2018. 7

[42] Yuxin Wu and Kaiming He. Group normalization. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 3–19, 2018. 3

[43] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J Black. Icon: Implicit clothed humans obtained from normals. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 13286–13296. IEEE, 2022. 1, 2, 3, 6, 7

[44] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 6184–6193, 2020. 1, 2

[45] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 7760–7770, 2019. 2

[46] Jianglong Ye, Yuntao Chen, Naiyan Wang, and Xiaolong Wang. Gifs: Neural implicit function for general shape representation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 12829–12839, 2022. 2

[47] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 5746–5756, 2021. 6

[48] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 818–833. Springer, 2014. 2

[49] Chao Zhang, Sergi Pujades, Michael J Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3d scan sequences. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 4191–4200, 2017. 6

[50] Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 11446–11456, 2021. 2

[51] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 8827–8836, 2018. 2

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 6

[53] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. *IEEE Trans. on Pattern Anal. Mach. Intell. (TPAMI)*, 44(6):3170–3184, 2021. 1, 2, 5, 6