Refining 3D Human Texture Estimation From a Single Image

Said Fahri Altindis, Adil Meric, Yusuf Dalva, *Graduate Student Member, IEEE*, Uğur Güdükbay, *Senior Member, IEEE*, and Aysegul Dundar

***Abstract*—Estimating 3D human texture from a single image is essential in graphics and vision. It requires learning a mapping function from input images of humans with diverse poses into the parametric (*uv*) space and reasonably hallucinating invisible parts. To achieve a high-quality 3D human texture estimation, we propose a framework that adaptively samples the input by a deformable convolution where offsets are learned via a deep neural network. Additionally, we describe a novel cycle consistency loss that improves view generalization. We further propose to train our framework with an uncertainty-based pixel-level image reconstruction loss, which enhances color fidelity. We compare our method against the state-of-the-art approaches and show significant qualitative and quantitative improvements.**

***Index Terms*—Texture estimation, deformable convolution, uncertainty estimation.**

I. Introduction

ESTIMATING 3D human texture from a single image is fundamental in many areas, such as virtual reality (VR), augmented reality (AR), gaming, robotics, and cloth try-ons. This problem is very challenging given the requirement for predicting the textures of invisible human body parts and the diversity of the pose and appearance of human bodies.

Predicting a three-dimensional (3D) human textured model from a single image receives increasing attention from the research community. Deep learning models are trained for this task thanks to the differentiable renderers, sometimes called neural renderers [31]. These renderers enable end-to-end training by approximating rasterization gradients and allow the backpropagation of image-based reconstruction losses. However, many of the proposed methods require labor-intensive, expensive data for training, such as 3D scanning [24], [35], [43], [49] or dense human pose estimation [35], [44]. In this work, we aim to learn texture reconstruction from a single image without the expensive 3D labels by relying on only multi-view images [28], [33], [62], [63], [65]. Among existing approaches, Zhao et al. [65] propose to use cross-view consistency to enforce the rendered image to match the image from a different view. Wang et al. [56] incorporate the re-identification loss to train the 3D human texture estimation model. Xu and Loy [63] set an attention-based architecture to allow information processing globally.

Fig. 1. Texture estimation is not aligned, unlike many tasks with input-output alignment (e.g., part segmentation).

Even though significant progress has been achieved in this domain, previous works still have limitations that hinder the quality of 3D human texture estimation. First, texture estimation from a single image has a set-up in which input and output images are not spatially aligned and, therefore, unsuitable for solving with Convolutional Neural Networks (CNNs) with local receptive fields. For example, hands can appear anywhere in input images, but they have a fixed corresponding location in the parametric *uv* space (cf. Fig. 1). Previous works propose an attention-based architecture to remedy this problem by effectively distributing input features into suitable locations in the parametric *uv* space. Our work shows that we can further improve texture estimation via a deformable convolution-based module, which we refer to as the *refinement* module. The learnable offsets of the deformable convolution come from a deep attention-based architecture; therefore, the refinement module can adaptively sample input images to output high-fidelity texture predictions for visible and invisible pixels. Second, previous approaches avoid using pixel-level reconstruction loss between rendered and ground-truth images since it performs poorly in generating details. The inaccurate human body pose and shape estimations result in misalignments between the rendered and input images, causing this performance degradation. We propose using a confidence-based pixel-level reconstruction loss to handle the misalignments, significantly improving results. Finally, we enhance our texture estimation with a novel cycle consistency loss. We apply cycle consistency by estimating texture from a

Received 12 March 2023; revised 10 May 2024; accepted 4 September 2024. Date of publication 10 September 2024; date of current version 5 November 2024. The work of Aysegul Dundar was supported in part by Marie Skłodowska-Curie Individual Fellowship. Recommended for acceptance by V. Lempitsky. *(Corresponding author: Aysegul Dundar.)*

Said Fahri Altindis, Uğur Güdükbay, and Aysegul Dundar are with the Department of Computer Science, Bilkent University, 06800 Ankara, Turkey (e-mail: adundar@cs.bilkent.edu.tr).

Adil Meric is with the School of Computation, Information and Technology, Technical University of Munich, 80333 München, Germany.

Yusuf Dalva is with the Department of Computer Science, Virginia Tech, Blacksburg, VA 24061 USA.

Code and additional results: https://github.com/saidaltindis/RefineTex.

Digital Object Identifier 10.1109/TPAMI.2024.3456817

single image, rendering it from a novel view, and encoding the texture again, making the model generalizable to different views.

We evaluate our method against the current state-of-the-art methods by using metrics, such as Structural Similarity Index Measure (SSIM) [59], Learned Perceptual Image Patch Similarity (LPIPS) [64], and Cosine Similarity (CosSim) [52], by comparing the input and rendered images from the same viewpoint, which provides a comparison for the original view. However, when used solely, this evaluation misses the essence of 3D models. A good model must generate appearance from *novel* viewpoints by predicting invisible regions successfully. Our work also evaluates the methods from novel views and aims to improve the results for the same and other novel viewpoints.

Our contributions are as follows:

1) We introduce a deformable convolution-based framework to handle the challenges of mapping unaligned spatially diverse input images into fixed parametric *uv* maps.
2) We adapt the confidence-based pixel-level reconstruction loss to handle the misalignments in the ground truth. We enable training with pixel-level reconstruction loss and facilitate a closer color appearance to the input image.
3) We introduce the cycle consistency loss to the task of texture estimation that improves the texture estimation quality.

We perform extensive evaluations with an array of quantitative metrics that show the effectiveness of our scheme compared to the several state-of-the-art approaches.

II. Related Work

3D texture estimation from images can enable various VR/AR applications and attracts much interest from the research community [6], [8], [11], [12], [16], [29], [36], especially the 3D human texture estimation [1], [2], [3], [5], [24], [35], [42], [49], [56], [63], [65], [68]. Many methods have been proposed for 3D human reconstruction that take multi-view images and optimize them [27], [47], [51]. Especially videos have been explored with implicit representations for modeling scenes [40], [41] as well as for human activity and performance reconstruction such as the recently proposed approaches, Vid2Avatar [21] and [45]. While achieving impressive results, these models are overfitting to a sequence and cannot be used for single-image inference.

In this work, we are also interested in 3D human reconstruction but inferring them from single-images [4], [19], [63], [65], [67], [68] since it is more applicable to real-world use cases. Even though many works aim to infer texture from single-view images, they may require significantly expensive labor-intensive data during training. For example, most of the methods require 3D scanning [24], [35], [43], [49], [53] such as the recently popularized implicit function-based methods [7], [22], [49], [50] and few others require dense human pose estimation [35], [44] or depth data [45]. In our work, we aim at learning texture prediction without expensive 3D labels but by relying on image datasets [56], [63], [65].

For 3D human texture estimation task, previous methods propose to train networks on image collections in a self-supervised manner to reconstruct the input image with a differentiable renderer [28], [31], [33], [62]. 3D body and pose models are predicted with state-of-the-art 3D human mesh reconstruction methods, and a human texture estimation network is trained to map images into the UV space. Mapping human images into UV-space is also used for dense human pose estimation [20], mesh recovery [58], garment prediction [26] as well as for texture estimations [56], [63], [65]. For texture estimation, training data is augmented with part segmentation models for superior performance [63], [65]. While promising progress has been achieved, there are still issues that need to be handled by previous works. One crucial challenge of texture prediction is strict input and output alignments. The prediction of textures needs to be sampled from a different location for each example. Some methods [29], [65] learn texture flows to sample pixels from the input image, while some others directly learn the texture maps in RGB values [56], and others use a combination [63]. Learning texture flows can transfer fine details directly from input to texture, whereas predicting RGB texture can more pleasantly visually synthesize invisible regions. The method of Xu and Loy [63] combines the benefits of these two by learning a fusion strategy. Our work combines the two approaches with a deformable convolution [10]. In this way, the prediction of the texture maps becomes easier for the network, which can operate on adaptive offsets. Deformable convolution is successfully integrated into many computer vision tasks such as object detection, instance segmentation [10], [70], and texture synthesis [39]. This task is also suitable for deformable convolution, and significant improvements can be achieved with a unique deformable-based design.

Another challenge of 3D texture estimation is to improve the texture predictions for invisible regions. For this problem, to output a complete texture, generative models are proposed that are referred to as neural rendering methods [14], [18], [34]. The most related to our work is StylePeople [18], which proposes to train a generative model of full-body human avatars. Via a StyleGAN [30] architecture, the model can sample a random code and synthesize novel humans. An encoder is trained to estimate the latent codes of StyleGAN to convert it to a reconstruction framework that outputs neural textures for a given input image. Because it is difficult to invert an image with an encoder alone, the framework also requires a computationally costly optimization step. Other reconstruction-based works utilize multi-view images for cross-view consistency learning to improve texture estimation for the invisible regions [65]. We further improve the results with a novel cycle consistency loss. Last but not least, previous works avoid using reconstruction losses between the input and output due to its degradation in performance [56], [63]. We propose to use a confidence-based reconstruction loss to improve the results further.

III. Method

A. Architecture

We work on a set-up where a 3D human texture estimation network is expected to map spatially variant input into the predefined *uv* space coordinates. For example, hands can be anywhere in input images but are registered to a fixed coordinate in

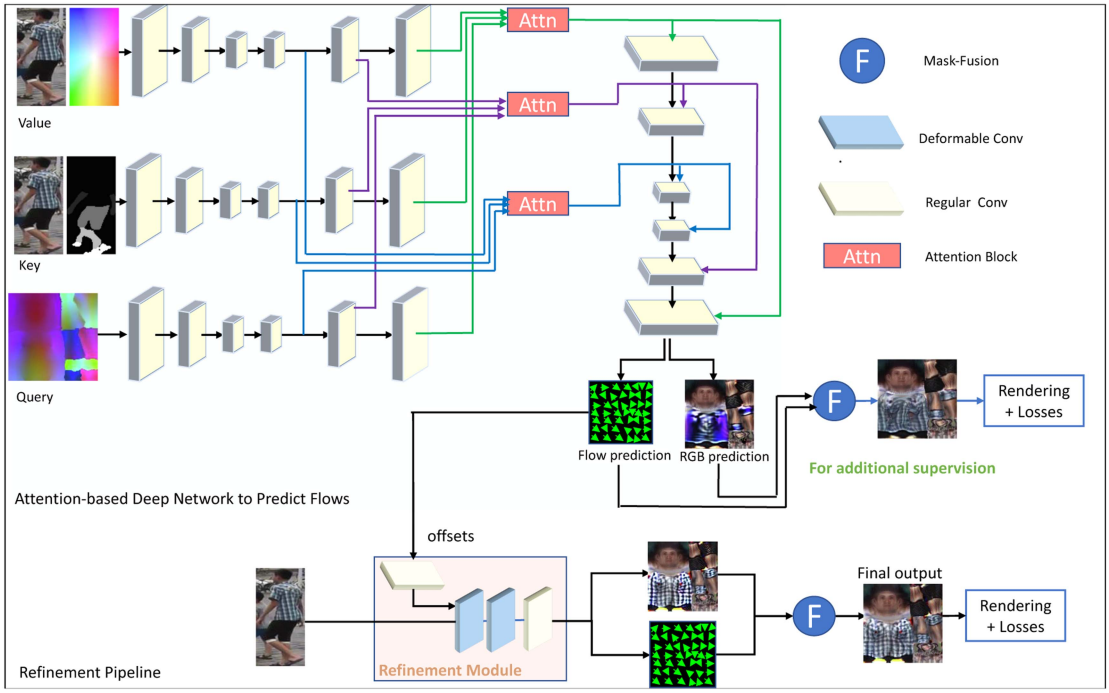

Fig. 2. The overall framework. We introduce a deformable convolution-based refinement module that learns offsets via an attention-based deep network [63]. This framework can handle the challenges of mapping unaligned spatially diverse input images into fixed parametric *uv* coordinates. We supervise the network with a branch depicted for additional supervision. In this way, offsets receive additional direct supervision as well. Mask prediction for the mask-fusion step is omitted from the figure for brevity.

parametric *uv* space. Since inputs and outputs are not aligned as in most computer vision tasks, the network architecture for this task requires special care. The relevant pixels for synthesizing a coordinate in the parametric *uv* space may be far away in the input space. While we can directly copy visible pixels from the input image to *uv* maps after the locations of corresponding pixels are detected, the network needs to synthesize invisible pixels by conditioning on the visible parts. Therefore, we can summarize the goal of the network into two: 1) finding correct offsets for the visible regions so that RGB values from input images can be directly copied, and 2) hallucinating RGB values for invisible pixels in the input image by processing what is visible.

We designed our framework to have the ability to adapt to the geometric variations in the input image. Our initial goal is to find the offsets of the associated pixels for each pixel in the parametric *uv* map. Our second goal is to process these associated pixels to output texture predictions with the offsets. The process of finding offsets is similar to flow predictions. However, instead of only sampling the input image with offsets to the output, we want a robust architecture that can adaptively sample input images to process further for better fidelity. With this motivation, we employ deformable convolutions. Unlike previous architectures that use deformable convolution, the pathway for predicting offsets includes a deep convolutional neural network, as given in Fig. 2. Previous architectures proposed for texture estimation can be used here as the deep network to output flow predictions. We use an attention-based architecture [63] as shown in Fig. 2. This architecture uses a color-encoding of the output UV space as query, 2D part-segmentation of input together with the input image as the key, and input image together with flow field of the image, i.e., 2D coordinates for each pixel as value. Multi-scale features are encoded from these inputs with separate encoder-decoder networks. The encoded value, key, and query features are input into the attention blocks [55]. The key correlates with the query elements to obtain the attention map for the input. Attention maps provide global information to distribute input features to output features. The outputs of attention blocks go through a *Unet*-like architecture, as shown in Fig. 2 to output flow and RGB predictions.

We refer to the module that contains deformable convolutions as the refinement module, as shown in Fig. 2. We calculate the offsets from flow predictions since flow predictions are trained to learn the mapping of the input image to the output. The flow predictions learn the absolute values of input coordinates. We transform them into offsets of pixel coordinates and apply a convolutional layer to them. With the offsets, the deformable convolution operation can sample the input image from far away pixel coordinates and process them to output texture maps. The refinement module takes the input image with the size of 128×64 and outputs features with the dimension of 128×128 based on the offsets in 128×128 spatial size. Deformable convolution operation, by default, expects offsets and input features to be in the same dimension. We modify the deformable convolution implementation to take an arbitrary input size concerning the offsets. The offset size indicates the output dimension, which provides an intuitive flow from input to output. This way, we can deform from 128×64 images to 128×128 texture maps.

We use the mask-fusion method to combine the advantages of RGB texture prediction and texture flow [63]. Our network's output consists of the RGB texture map T_{RGB}, the texture flow

F, and a fusion mask M. M is not included in Fig. 2 for brevity. The mask-fusion process is as follows:

$$T = M \odot f_{sample}(F, I) + (1 - M) \odot T_{RGB}, \quad (1)$$

where f_{sample} refers to the bilinear sampling function that samples textures from the input image, I, by the flow predictions, F, and M is a binary mask. In this way, visible pixels can be taken by the more accurate $f_{sample}(F, I)$, and invisible pixels can be taken from T_{RGB}.

We supervise the framework by rendering the intermediate results from the deep network and backpropagating the losses, as shown in Fig. 2. In this way, offsets receive more direct supervision as well. We progressively rely more on the supervision the refinement module receives from the final output by turning off the additional supervision.

B. Loss Functions

The loss functions are calculated between the input and rendered images in a self-supervised manner and are explained in this section. Given an input image, I, our model outputs human texture, $T(I)$. This texture, together with mesh, M, and camera, C, predictions from state-of-the-art RSC-Net model [62] which is built on SMPL model [38] are rendered with a differentiable renderer [31] which outputs an image, I^r. We use the same loss functions between I and I^r as in previous works [63] and an uncertainty-based reconstruction loss function and cycle consistency loss, significantly improving the results. This section first reviews the losses we used from previous works. The first loss function is the re-identification loss [56], which minimizes the distance from pedestrian re-identification network (Φ) [52] at different feature layers (j):

$$\mathcal{L}_{reid} = ||\Phi_j(I) - \Phi_j(I^r)||_2^2. \quad (2)$$

Another previously proposed loss objective for this task is the part-style loss function [63]. This loss enforces the similarity between each body part of the rendered human and the input image as measured by Gram-matrix [15].

$$\mathcal{L}_{style} = ||G(M_p \odot \Phi_1(I)) - G\left(M'_p \odot \Phi_1(I^r)\right)||_2^2, \quad (3)$$

where M_p and M'_p are the human part segmentation masks from the 2D human parsing model [23] and 3D mesh, respectively, p indicates the body part, and this loss is calculated for each body part separately and summed together. G stands for Gram-matrix, features are again encoded by the same pedestrian re-identification network (Φ), and only the first layer is used. Another loss function we use is the face structure loss [63], which can be used in our framework to generate accurate and realistic face images. Since all human faces follow the same structure, face structure loss ensures the similarity between estimated textures and synthetically generated textures. Each region of the parametric *uv* map is predetermined; hence, the location of the face in the texture is constant. This loss only checks similarities between faces by applying a fixed mask:

$$\mathcal{L}_{face} = -\frac{1}{N}\sum_{i=1}^{N} s\left(M_{face} \odot T(I), M_{face} \odot F^i_{syn}\right), \quad (4)$$

where $\{F^i_{syn}\}_{i=1}^N$ is a set of synthetically generated human textures obtained from the synthetic human dataset [54], M_{face} is a predefined binary mask that indicates the face region on the texture map, and s is a structure-similarity function [59] that calculates face similarities. The loss function in (4) optimizes the network to output face texture predictions with similar structures as F_{syn}. This results in the generation of plausible face textures while retaining the colors of the input human. The losses we use from previous works are our base losses, and (5) gives the overall base loss.

$$\mathcal{L}_{base} = \lambda_1 \times \mathcal{L}_{reid} + \lambda_2 \times \mathcal{L}_{style} + \lambda_3 \times \mathcal{L}_{face}, \quad (5)$$

where $\lambda_1 = 5000, \lambda_2 = 0.4$, and $\lambda_3 = 0.01$. We take these parameters from previous work and do not tune them.

1) Uncertainty-Based Reconstruction Loss: Previous approaches avoid using pixel-level reconstruction loss between rendered and input images since it performs poorly in generating details. The inaccurate estimation of human body poses and shapes may result in misalignments between the rendered and input images. Due to this problem, 3D human texture estimation models are trained only with re-identification losses that compare features at a high level and style losses that do not use spatial correspondence. On the other hand, it is shown that pixel-level reconstruction losses improve results when the generated and output images are aligned [13], [25], [46], [57]. To take advantage of pixel-level reconstruction loss and be robust to misalignments, we propose to estimate a confidence map, σ, to adjust the reconstruction loss objective. The ground-truth output image, I, and the rendered image, I^r, are compared via the loss given in (6) as was also defined in [60].

$$\mathcal{L}_{url} = -\sum_{x,y} \ln\left(\frac{1}{\sqrt{2\sigma^2_{x,y}}} \exp -\frac{\sqrt{2}|I_{x,y} - I^r_{x,y}|}{\sigma_{x,y}}\right). \quad (6)$$

In this loss objective, the role of the confidence map, σ, is to estimate the aleatoric uncertainty of the model [32]. This uncertainty is the noise associated with the data collection that cannot be reduced with more data. In our setting, it is the noise caused by the misalignment of our data. (x, y) are the spatial pixel coordinates of I. The objective is the negative log-likelihood of a factorized Laplacian distribution with the mean predicted by the model and σ predicted by the confidence model. This way, the model calibrates itself and minimizes the reconstruction loss by optimizing the confidence map [32], [60]. The confidence map estimates which pixels from the rendered image will not be aligned with the ground truth. The confidence model has the *Unet* architecture [48], which takes the input image, I, and outputs the σ. This model is only used during training and is not needed during inference.

2) Cycle Consistency Loss: We also propose a cycle loss to enforce consistency between the textures estimated by the model from input images and the model's renderings for different views. In the cycle process, the model generates the initial texture map, uv_{map}, from the given input image as I by the texture estimation network, T. We render a new image using the estimated textures for a different view obtained from another image of the same person with a renderer, R, Body Mesh

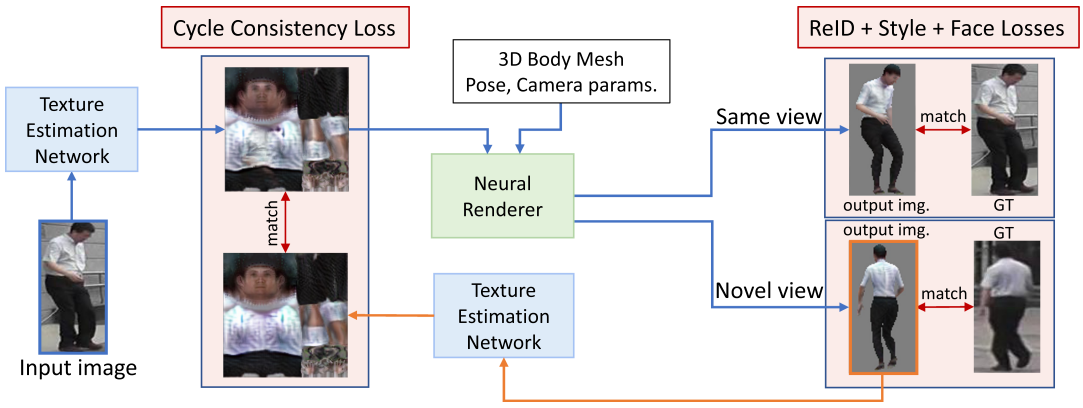

Fig. 3. We predict a texture from an input image and render it for both the same and novel views. From the image rendered with a novel view, we estimate the texture again. We expect the estimated texture to match the texture predicted from the input image.

Parameters, m, and camera parameters, c. From the rendered image, the network estimates the texture again. We expect these two estimated textures to be as close as possible; hence, we calculate a pixel-wise L2 loss between two estimated textures, as given in (7).

$$\mathcal{L}_{cyc} = ||T(I) - T(R(T(I), m, c))||_2. \quad (7)$$

Fig. 3 explains the detailed structure of the cycle consistency. In previous methods, the texture map is estimated from the input image, and it is rendered for the same and novel views to enforce multi-view consistency and minimize base losses, as described in the previous subsection. We also estimate the texture from the novel view rendering to obtain additional guidance for the texture estimation network. This additional loss also helps us to achieve our second objective, where we can compare images at the pixel level.

3) Total Loss: The baseline loss L_{base} defined at (5) is calculated twice for two different rendered images. The first baseline loss, $L_{base-sv}$, is calculated between the input image and the image rendered with the input image's generated texture and camera parameters. Moreover, to improve texture estimation of unseen parts, the second baseline loss, $L_{base-nv}$, is calculated between the ground-truth image from a different view and the image rendered with the generated texture of the input image and camera parameters of the novel-view ground-truth image. $L_{base-nv}$ provides the multi-view consistency. Additionally, we have two new losses. The overall loss function to train our framework is as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{base-sv} + \mathcal{L}_{base-nv} + \lambda_4 * \mathcal{L}_{cyc} + \lambda_5 * \mathcal{L}_{url}, \quad (8)$$

where $\lambda_4 = 0.1$, and $\lambda_5 = 10^{-3}$.

IV. Experiments

Dataset, Architecture, and Training Details: We use the Market-1501 dataset [66] in our experiments. Among the human images of 1501 person identities, we use the same training-testing split as in other works [56], [63]. We additionally run experiments on the DeepFashion dataset [37]. The DeepFashion dataset includes in-shop clothes images. These images show large pose and view variations. We follow the same train/test split from previous work [65]. We estimate the mesh and camera parameters with HMR2.0 [17]. Previous work [65] removes the images that only contain a small human body part and ends up with 20,185 training and 6,639 testing images. We use their split.

The overall architecture includes deformable and traditional convolution layers. A deformable convolution receives initial offsets from a deep attention-based network, starting with the refinement module. This attention-based network has an encoder-decoder architecture where, at each scale, there is an attention block as introduced in [63]. The network has 6 convolution layers, each with a filter size of 3×3 and 128 channel size. There are downsamplings after the second and third convolution layers and upsamplings after the fourth and fifth convolution layers. The first three convolution layers correspond to the encoder, and the other three belong to the decoder. There are skip connections between the encoder and the decoder at each scale, which employs an attention block, and the output of the attention blocks is summed up with the decoded features.

The initial offsets coming from the deep network go through a convolution layer to output 18 channels ($2 \times 3 \times 3$) to be used as the offsets to the deformable convolution layer with filters 3×3 width and height and 128 output channels. 18 channel corresponds to the x, y offsets for each pixel in a kernel (3×3). The offsets have a dimension of $b \times 18 \times 128 \times 128$, and input has a dimension of $b \times 4 \times 128 \times 64$ where channel size of 4 refers to the RGB image with part segmentation concatenated, and b is batch size. We modify the deformable convolution implementation to take an arbitrary input size for the offsets. Offset size indicates what the output dimension will be. Therefore, from the input image with a spatial dimension of 128×64, we output feature maps with a spatial dimension of 128×128. The output here is processed with additional deformable and convolutional layers. We use kernel size 3, stride 1, and padding 1 for each convolution layer. The channels change as {4, 128, 128, 128, 6}. Additionally, we have a skip connection from the deep attention network's predictions to the refinement module. We concatenate the predictions from the deep network with the deformable convolution's output. The final channel size of 6 corresponds to the RGB predictions (3 channels), UV flow

TABLE I
COMPARISONS OF MODELS TRAINED WITH MULTI-VIEW CONSISTENCY AND WITHOUT MULTI-VIEW CONSISTENCY (SINGLE-VIEW IMAGES)

	SSIM ⇑		LPIPS ⇓		CosSim ⇑		CosSim-R ⇑	
	SV	NV	SV	NV	SV	NV	SV	NV
Trained with Multi-view	0.7422	**0.6535**	0.1154	**0.2040**	0.5747	**0.4943**	0.5422	**0.4736**
Trained with Single-view	**0.7706**	0.6494	**0.0963**	0.2150	**0.5823**	0.4809	**0.5496**	0.4585

Results in bold indicate the best in each column.

predictions (2 channels), and fusion mask prediction (1 channel). The RGB and UV flow predictions go through Tanh activation, and mask predictions go through sigmoid activation layers. Each convolution layer in the overall architecture is followed by batch normalization and ReLU layers. We use the same architecture for our experiments on both datasets.

The confidence model has 6 convolution layers, each with a filter size of 3×3 and 128 channel size. Again, each convolution layer is followed by batch normalization and ReLU layers. There are downsamplings after the second and third convolution layers and upsamplings after the fourth and fifth convolution layers. The first three convolution layers correspond to the encoder, and the other three belong to the decoder. There are skip connections between the encoder and decoder at each scale. Encoded features and decoded features are summed up via skip connections. The last convolutional layer reduces the channel size to 1, followed by the softplus activation function. This network is only used during training.

We use the Adam optimizer and train our framework with a batch size of 16 and a learning rate of 1×10^{-3} and betas $= (0.9, 0.999)$ for 200 epochs. We do not use a learning rate scheduling and keep the learning rate the same for 200 epochs.

Evaluation Metrics: Following the previous works, we use various metrics for evaluation. SSIM [59] and LPIPS [64] find pixel- and feature-level similarities between the output and ground-truth images. Additionally, cosine similarities of person ReID features [52] are computed to evaluate a high-level semantic similarity, e.g., how likely the rendered human is the same person from the ground-truth image. Following previous work [61], we calculate this metric with two different networks, PCB [52] and TorchReid [69], resulting in CosSim and CosSim-R metrics, respectively.

Whenever a metric is used to evaluate the input and rendered images from the same view, only the visible texture estimates are considered since that evaluation protocol measures the estimated texture from the same view. We also use the metrics to evaluate a rendered image for a novel view for which we have the ground truth to overcome this limitation. Given an input image from one view, we render the estimated 3D human texture based on a camera view and pose estimate from another. We evaluate the results for all the other available views for each person. We refer to the same view evaluation as *SV* and the novel view as *NV* in the tables.

We experiment with the reliability of novel view results by comparing baseline methods trained with and without multi-view consistency. Multi-view consistency is broadly used on datasets that contain multiple images taken from different views of the same object [63], [65]. One view is input in these settings, and loss functions are calculated on the same input view and a novel view. The multi-view consistency improves texture estimation quality for the invisible regions because networks receive gradients from the novel image target for invisible parts in the input image.

Table I compares our baseline models, trained with multi-view and single-view data, which refers to using multi-view consistency and not, respectively. The traditional evaluation results on the same view show that the single-view baseline model achieves significantly better scores. This is because it is tuned to estimate the visible areas perfectly, producing better results on the same but worse on a novel view, as seen in Fig. 4. The quality of renderings in novel views is usually more valuable and the primary purpose of 3D models. Table I shows the evaluation of the quality of novel view renderings. This setup indicates the limitation of previously used evaluation protocol by 3D human texture estimation models. In this work, we are interested in improving the quality of both the same and novel views, and we keep track of both of these metrics.

A. Comparison With the State-of-the-Art

We compare our method against the state-of-the-art 3D human texture estimation methods: HPBTT [65], RSTG [56], TexGlo [61], and Texformer [63]. Quantitative results for Same-view (SV) and Novel-view (NV) are presented in Table II. We calculate the results of NV of the methods with released models of RSTG and Texformer. First, we compare the methods on the Market-1501 dataset. Since HPBTT uses a different data split, we train it with our split with their released code. HPBTT [65] outputs texture flow with a regular convolutional neural network that takes body segmentation and pose as input. Since the method outputs texture flow, the colors match the input image, which results in a relatively good performance on SSIM and LPIPS metrics for the same view. On LPIPS, HPBTT even achieves the second-best result. However, its results are poor when measured with CosSim and CosSim-R metrics due to the artifacts in the final renderings. RSTG [56] and TexGlo [61] generate results in fewer artifacts, which leads to relatively better performance on CosSim and CosSim-R metrics. On the other hand, their results do not match the input images in colors and fine details, which causes significantly worse SSIM and LPIPS scores. Texformer [63] is closest to our work with good metrics overall. In our work, we significantly improve their results. Our proposed method achieves consistently better results than the competing methods on all metrics for both same and novel view evaluations with slightly more parameters than Texformer and significantly fewer parameters than the others.

Fig. 5 presents the qualitative results of our method and competing methods. The HPBTT model generates results with

Fig. 4. Input/same view and novel view ground-truth images are provided in the first two columns. Other columns show the baseline-trained results with multi-view consistency and trained without multi-view consistency (with single-view images). Blue arrows show that models trained with single-view images better reconstruct the same view but not the novel one. Therefore, instead of only evaluating the models based on the reconstruction of the same view, which misses the point of 3D models, we additionally evaluate the models from a novel view.

TABLE II
EVALUATION RESULTS OF OUR METHOD AND STATE-OF-THE-ART COMPETING METHODS

		SSIM ⇑		LPIPS ⇓		CosSim ⇑		CosSim-R ⇑		Params. (M)
Dataset	Method	SV	NV	SV	NV	SV	NV	SV	NV	
Market-1501	HPBTT [65]	0.7380	0.6496	0.1148	0.2156	0.5336	0.4697	0.5077	0.4508	42.3
	RSTG [56]	0.6735	0.6283	0.1778	0.2421	0.5282	0.4717	0.4924	0.4454	13.4
	TexGlo [61]	0.6658	-	0.1776	-	0.5408	-	0.5048	-	16.1
	Texformer [63]	0.7422	0.6535	0.1154	0.2040	0.5747	0.4943	0.5422	0.4736	7.6
	Ours	**0.7611**	**0.6544**	**0.1003**	**0.2040**	**0.5858**	**0.4963**	**0.5538**	**0.4758**	8.2
DeepFashion	HPBTT [65]	**0.7610**	**0.7364**	0.2433	**0.2637**	0.6066	0.5792	0.5939	0.5639	42.3
	Texformer [63]	0.6932	0.5465	0.2189	0.3595	0.7724	0.6248	0.7244	0.5961	7.6
	Ours	0.7512	0.5557	**0.1938**	0.3579	**0.7929**	**0.6254**	**0.7417**	**0.5983**	8.2

Results in bold indicate the best in each column, and underscored results indicate the second.

many artifacts, especially in the invisible regions. RSTG does not suffer from these severe artifacts; however, the method outputs blurry predictions lacking fine details. Texformer achieves better results than HPBTT and RSTG but still does not achieve high fidelity to the input images, especially on the challenging patterns, e.g., check shirts. Finally, our method achieves high-quality results with fine details, high-fidelity colors, and texture patterns.

We also conduct a user study on the Market-1501 dataset with 30 samples among 20 users. We set an A/B test and provide users with input images, and predictions rotated in 360°, along the azimuth, in GIF format so they can see the inconsistencies in the texture easily. We limit our user study to our method versus Texformer since they significantly outperform the previous works, as shown in Table II and Fig. 5. The left-right order is randomized to ensure fair comparisons. We ask users to select the best result according to fidelity to the input image and whether the output looks realistic and high-quality overall. Users select ours instead of Texformer 71% of the time (50% is a tie). These results are consistent with reported metrics and qualitative results.

Next, we compare methods on the DeepFashion dataset in Table II. We train HPBTT [65], Texformer [63], and our method on this dataset since they are the top-3 methods from the Market-1501 dataset. As shown in Table II, our method achieves an even more significant improvement on this dataset. HPBTT achieves better SSIM scores. HPBTT's SSIM scores are also high on the Market-1501 dataset. However, it is also shown that the SSIM score sometimes prefers blurred images. However, all the other metrics are poor due to the artifacts in the final renderings. As shown in Fig. 6, the same behavior is observed there as HPBTT outputs blurred images for novel views. Our method achieves significantly better CoSim and CoSim-R scores. Our method also shows significant improvements over Texformer and achieves better color consistency between the visible and invisible pixels. It may be because the dataset has more variations in the poses, and our contributions achieve significant improvements on this challenging dataset.

We also compare with coordinate-based texture inpainting for pose-guided human image generation [19]. CoordInpaint has two pipelines. First, dense poses are estimated by the Dense Pose method [20], and then they are converted to SMPL coordinates using a predefined mapping (provided with the DensePose). The first pipeline obtains a complete body texture through an inpainting network. The output of the first pipeline is a texture map that can be used to render humans with different poses. In the second pipeline, images are rendered for a target pose and further processed in the image space. The second pipeline, therefore, does not output a 3D model but refines the results in image space. Hence, our method is comparable with the output of the first pipeline. We provide the comparison results in Figs. 6

Fig. 5. Qualitative results of our method and state-of-the-art competing methods on the Market-1501 dataset.

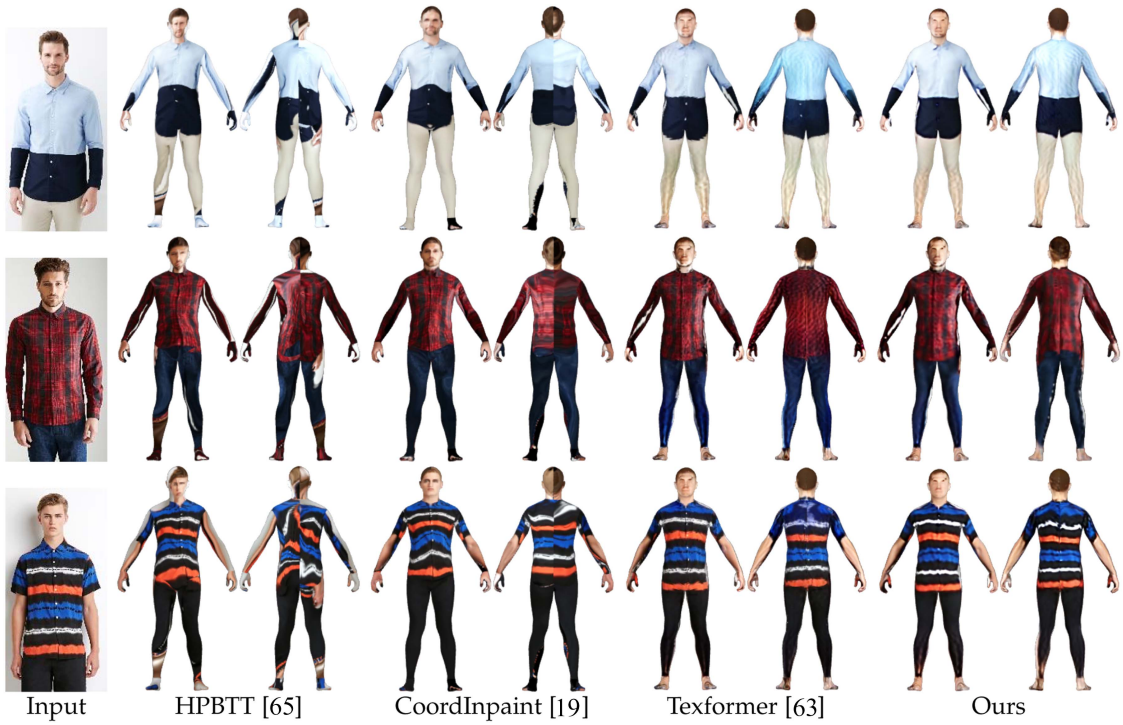

Fig. 6. Qualitative results of our method and state-of-the-art competing methods on the DeepFashion dataset.

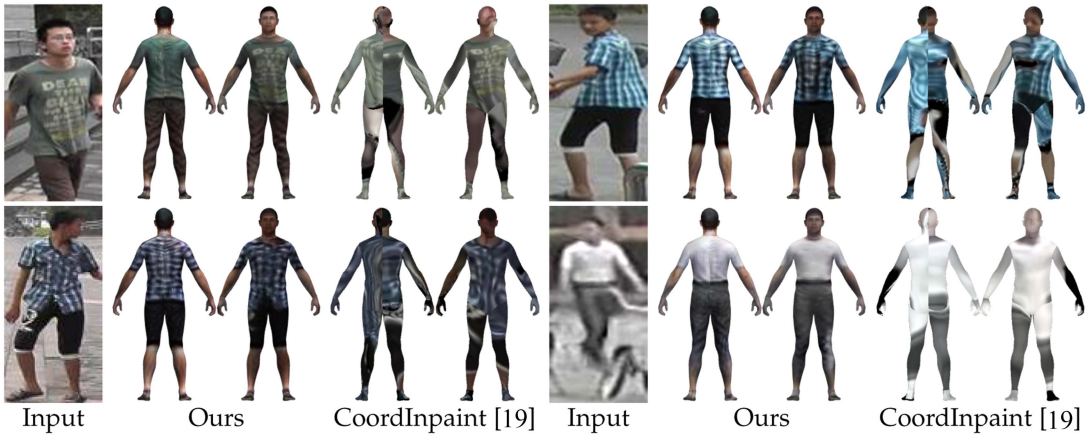

Fig. 7. Qualitative results of our method and Coordinate-based texture inpainting [19] on the Market-1501 dataset.

TABLE III
ABLATION STUDY ON THE MARKET-1501 DATASET

	SSIM ⇑		LPIPS ⇓		CosSim ⇑		CosSim-R ⇑	
	SV	NV	SV	NV	SV	NV	SV	NV
BL (Baseline)	0.7422	0.6535	0.1154	0.2040	0.5747	0.4943	0.5422	0.4736
BL + Conv Refine	0.7196	0.6520	0.1293	0.2052	0.5701	0.4976	0.5343	0.4758
BL + Deformable Refine	0.7490	0.6511	**0.1038**	0.2046	**0.5887**	**0.4976**	**0.5568**	**0.4775**
BL + Deformable Refine only RGB	0.7413	0.6500	0.1087	0.2039	0.5847	0.4957	0.5530	0.4746
BL + URL	**0.7560**	**0.6540**	0.1053	**0.2020**	0.5813	0.4802	0.5481	0.4767
BL + Cycle	0.7448	0.6503	0.1107	0.2027	0.5837	0.4968	0.5502	0.4761

and 7 forDeepFashion and the Market-1501 datasets, respectively. CoordInpaint's first stage, where texture is estimated, fails to output realistic invisible parts. Since the CoordInpaint model has two pipelines and achieves good results after its second pipeline, we do not include their work in our main comparisons.

B. Ablation Study

We conduct an ablation study to show the improvements of each proposed contribution as provided in Table III and Fig. 8. Our baseline model (BL) only includes the deep network from Fig. 2 without the final branch, which is the deformable convolution-based module. BL is only trained with base losses described in Section III-B. We then test each proposed change separately. First, we add a deformation-based refinement module to the architecture (BL+Deformable Refine). Next, we experiment with proposed loss objectives by adding uncertainty-based reconstruction loss (BL+URL) and cycle consistency loss (BL+Cycle).

Adding a deformable convolution-based refinement module improves all metrics. To measure the effectiveness of the proposed refinement module, we experiment with the improvements that come from the increased capacity (additional learning parameters). To test that, we experiment with a shallow UNET architecture, which is convolution-based, as the refinement module. As shown in Table III (BL+Conv Refine), the additional layers do not improve the results. As shown in Fig. 8, the refinement module improves challenging scenarios where the shirt stripes have superior quality. In Fig. 9, we visualize the corresponding offsets of deformable convolution for marked

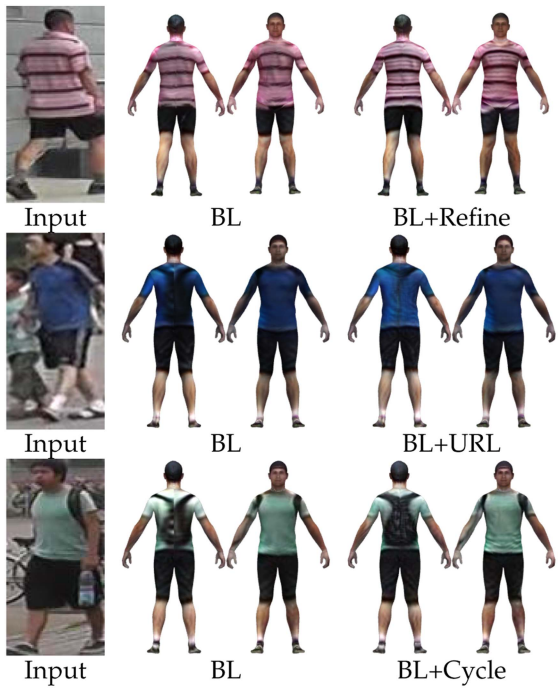

Fig. 8. Qualitative results of Ablation Study.

points from the UV map. We mark the same coordinates for the two examples in each row. As shown in the input image, the offsets are in different locations in the input based on the content. The texture estimation task fits well with deformable convolution. We also test if RGB texture estimations alone

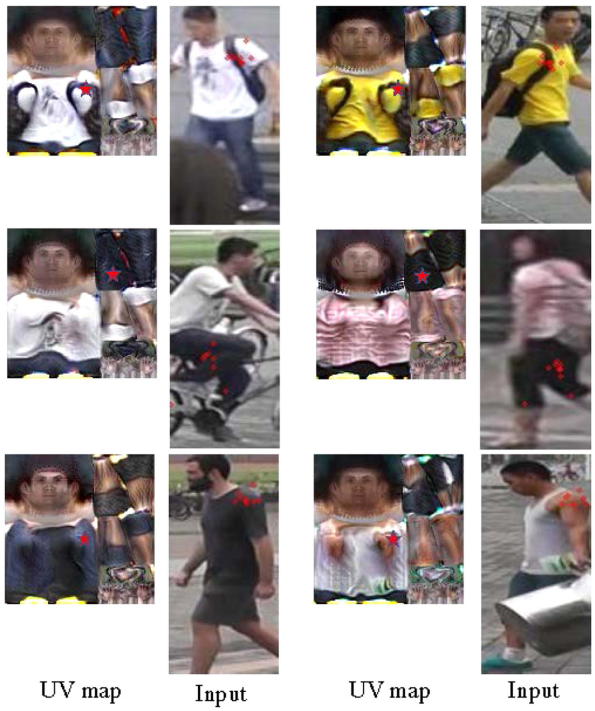

Fig. 9. Visualization of offsets in input images.

without the flow estimation are enough to achieve good results in Table III. When we compare Deformable Refine versus Deformable Refine only RGB, we observe that additional flow estimation slightly improves the results so we decide to keep it. URL significantly improves the SSIM and LPIPS metrics by providing pixel-level reconstruction loss. The second example in Fig. 8 has superior color fidelity to the input image. The third row of Fig. 8 shows that the cycle consistency loss significantly improves the prediction of invisible regions. As can be seen, a more realistic backpack is predicted by only seeing its straps with the cycle consistency loss.

V. Conclusion

We propose a framework to refine 3D human texture estimation from a single image. We use a deformable convolution-based refinement module to adaptively sample an input image for better quality. We also introduce an uncertainty-based reconstruction and novel cycle consistency losses responsible for our high-fidelity texture estimation. We show several qualitative and quantitative improvements compared to the state-of-the-art methods. We hope our work will inspire future research to test their texture inferences for view generalization by evaluating novel inferences.

Limitations: Regarding limitations, following the previous works, our framework uses the human body model from SMPL [38], which is unsuitable for loose-fitting clothes, e.g., long loose skirts. To overcome this limitation, the proposed contributions can be combined with more advanced body models, such as TightCap [9], a data-driven approach to capture both the human shape and dressed garments.

References

[1] T. Alldieck, M. Magnor, B. L. Bhatnagar, C. Theobalt, and G. Pons-Moll, "Learning to reconstruct people in clothing from a single RGB camera," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1175–1186.
[2] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll, "Detailed human avatars from monocular video," in *Proc. Int. Conf. 3D Vis.*, 2018, pp. 98–109.
[3] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll, "Video based reconstruction of 3D people models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8387–8397.
[4] T. Alldieck, G. Pons-Moll, C. Theobalt, and M. Magnor, "Tex2Shape: Detailed full human body geometry from a single image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2293–2303.
[5] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll, "Multi-garment net: Learning to dress 3D people from images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5420–5430.
[6] A. Bhattad, A. Dundar, G. Liu, A. Tao, and B. Catanzaro, "View generalization for single image textured 3D models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6081–6090.
[7] Y. Cao, G. Chen, K. Han, W. Yang, and K.-Y. K. Wong, "JIFF: Jointly-aligned implicit face function for high quality single view clothed human reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2719–2729.
[8] W. Chen et al., "Learning to predict 3D objects with an interpolation-based differentiable renderer," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9609–9619.
[9] X. Chen, A. Pang, W. Yang, P. Wang, L. Xu, and J. Yu, "TightCap: 3D human shape capture with clothing tightness field," *ACM Trans. Graph.*, vol. 41, no. 1, pp. 9:1–9:17, Nov. 2021.
[10] J. Dai et al., "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.
[11] A. Dundar, J. Gao, A. Tao, and B. Catanzaro, "Fine detailed texture learning for 3D meshes with generative models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 14563–14574, Dec. 2023.
[12] A. Dundar, J. Gao, A. Tao, and B. Catanzaro, "Progressive learning of 3D reconstruction network from 2D GAN data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 2, pp. 793–804, Feb. 2024.
[13] A. Dundar, K. Sapra, G. Liu, A. Tao, and B. Catanzaro, "Panoptic-based image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8070–8079.
[14] J. Gao et al., "GET3D: A generative model of high quality 3D textured shapes learned from images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 31841–31854.
[15] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2414–2423.
[16] S. Goel, A. Kanazawa, and J. Malik, "Shape and viewpoint without keypoints," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 88–104.
[17] S. Goel, G. Pavlakos, J. Rajasegaran, A. Kanazawa, and J. Malik, "Humans in 4D: Reconstructing and tracking humans with transformers," 2023, *arXiv:2305.20091*.
[18] A. Grigorev et al., "StylePeople: A generative model of fullbody human avatars," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5151–5160.
[19] A. Grigorev, A. Sevastopolsky, A. Vakhitov, and V. Lempitsky, "Coordinate-based texture inpainting for pose-guided human image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12135–12144.
[20] R. A. Güler, N. Neverova, and I. Kokkinos, "DensePose: Dense human pose estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7297–7306.
[21] C. Guo, T. Jiang, X. Chen, J. Song, and O. Hilliges, "Vid2Avatar: 3D avatar reconstruction from videos in the wild via self-supervised scene decomposition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 12858–12868.
[22] T. He, J. Collomosse, H. Jin, and S. Soatto, "Geo-PIFu: Geometry and pixel aligned implicit functions for single-view human reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 9276–9287.
[23] H. Huang et al., "EANet: Enhancing alignment for cross-domain person re-identification," 2018, *arXiv: 1812.11369*.
[24] Z. Huang, Y. Xu, C. Lassner, H. Li, and T. Tung, "ARCH: Animatable Reconstruction of Clothed Humans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3093–3102.

[25] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5967–5976.
[26] Y. Jafarian et al., "Normal-guided garment uv prediction for human re-texturing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4627–4636.
[27] W. Jiang, K. M. Yi, G. Samei, O. Tuzel, and A. Ranjan, "NeuMan: Neural human radiance field from a single video," 2022, *arXiv:2203.12575*.
[28] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7122–7131.
[29] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 386–402.
[30] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.
[31] H. Kato, Y. Ushiku, and T. Harada, "Neural 3D mesh renderer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3907–3916.
[32] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5580–5590.
[33] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3D human pose and shape via model-fitting in the loop," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2252–2261.
[34] Y. Lan, X. Meng, S. Yang, C. C. Loy, and B. Dai, "Self-supervised geometry-aware encoder for style-based 3D GAN inversion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 20940–20949.
[35] V. Lazova, E. Insafutdinov, and G. Pons-Moll, "360-degree textures of people in clothing from a single image," in *Proc. Int. Conf. 3D Vis.*, 2019, pp. 643–653.
[36] X. Li et al., "Self-supervised single-view 3D reconstruction via semantic consistency," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 677–693.
[37] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1096–1104.
[38] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 248.
[39] M. Mardani, G. Liu, A. Dundar, S. Liu, A. Tao, and B. Catanzaro, "Neural FFTs for universal texture image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 14081–14092.
[40] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "NeRF in the wild: Neural radiance fields for unconstrained photo collections," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7210–7219.
[41] B. Mildenhall, P.P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, 2021.
[42] A. Mir, T. Alldieck, and G. Pons-Moll, "Learning to transfer texture from clothing images to 3D humans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7023–7034.
[43] R. Natsume et al., "SiCloPe: Silhouette-based clothed people," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4480–4490.
[44] N. Neverova, R. A. Guler, and I. Kokkinos, "Dense pose transfer," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 123–138.
[45] A. Pang, X. Chen, H. Luo, M. Wu, J. Yu, and L. Xu, "Few-shot neural human performance rendering from sparse RGBD videos," 2021, *arXiv:2107.06505*.
[46] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2337–2346.
[47] S. Peng et al., "Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9054–9063.
[48] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, Springer, 2015, pp. 234–241.
[49] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "PiFu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2304–2314.
[50] S. Saito, T. Simon, J. Saragih, and H. Joo, "PiFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 84–93.
[51] S.-Y. Su, F. Yu, M. Zollhoefer, and H. Rhodin, "A-NeRF: Articulated neural radiance fields for learning human shape, appearance, and pose," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 12278–12291.
[52] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 480–496.
[53] D. Svitov, D. Gudkov, R. Bashirov, and V. Lempitsky, "Dinar: Diffusion inpainting of neural textures for one-shot human avatars," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 7062–7072.
[54] G. Varol et al., "Learning from synthetic humans," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4627–4635.
[55] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[56] J. Wang, Y. Zhong, Y. Li, C. Zhang, and Y. Wei, "Re-identification supervised texture generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11846–11856.
[57] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.
[58] Y. Wang et al., "Learning dense uv completion for human mesh recovery," 2023, *arXiv:2307.11074*.
[59] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[60] S. Wu, C. Rupprecht, and A. Vedaldi, "Unsupervised learning of probably symmetric deformable 3D objects from images in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1–10.
[61] X. Xu, H. Chen, F. Moreno-Noguer, L. A. Jeni, and F. De laTorre, "3D human pose, shape and texture from low-resolution images and videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4490–4504, Sep. 2022.
[62] X. Xu, H. Chen, F. Moreno-Noguer, L. A. Jeni, and F. D. l. Torre, "3D human shape and pose from a single low-resolution image with self-supervised learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 284–300.
[63] X. Xu and C. C. Loy, "3D human texture estimation from a single image with transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 13829–13838.
[64] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.
[65] F. Zhao, S. Liao, K. Zhang, and L. Shao, "Human parsing based texture transfer from single image to 3D human via cross-view consistency," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 14326–14337.
[66] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1116–1124.
[67] Z. Zheng, T. Yu, Y. Wei, Q. Dai, and Y. Liu, "DeepHuman: 3D human reconstruction from a single image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7739–7749.
[68] T. Zhi, C. Lassner, T. Tung, C. Stoll, S. G. Narasimhan, and M. Vo, "TexMesh: Reconstructing detailed human texture and geometry from RGB-D video," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 492–509.
[69] K. Zhou and T. Xiang, "Torchreid: A library for deep learning person re-identification in PyTorch," 2019, *arXiv: 1910.10093*.
[70] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets V2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9308–9316.

Said Fahri Altindis received the BSc degree in computer science from TED university, Ankara, Turkey, in 2020, and the MSc degree in computer science from Bilkent University, Ankara, Turkey, in 2023. His research interests include domain adaptation, instance segmentation, and generative models.

Adil Meric received the BSc degree in computer science from Bilkent University, Ankara, Turkey, in 2022. He is currently working toward the MSc degree with the School of Computation, Information and Technology, Technical University of Munich. His research interests include image feature manipulation, object localization, generative models, and robustness enhancement of deep neural networks.

Yusuf Dalva (Graduate Student Member, IEEE) received the BSc and MSc degrees in computer science from Bilkent University, Turkey. He is currently working toward the PhD degree with Virginia Tech University. His research interests include image understanding and feature manipulation, generative models, and robustness enhancement of deep neural networks.

Uğur Güdükbay (Senior Member, IEEE) received the BS degree in computer engineering from the Middle East Technical University, Ankara, Turkey, in 1987, and the MSc and PhD degrees in computer engineering and information science from Bilkent University, Ankara, Turkey, in 1989 and 1994, respectively. He conducted research as a postdoctoral fellow with the Human Modeling and Simulation Laboratory, University of Pennsylvania. Currently, he is a professor with the Department of Computer Engineering, Bilkent University. His research interests include deep learning for human modeling and animation, conversational virtual agents, personality and emotion synthesis, crowd simulation, rendering, and visualization. He is a senior member of ACM.

Aysegul Dundar received the BSc degree in electrical and electronics engineering from Bogazici University, Turkey, in 2011, and the PhD degree from Purdue University, in 2016. She is an assistant professor of computer science with Bilkent University, Ankara, Turkey. Previously, she was a sr. research scientist with NVIDIA, California, USA. In CVPR 2018, she won first place in the Domain Adaptation for Semantic Segmentation Competition in the Workshop on Autonomous Vehicle Challenge. Her current research focuses on domain adaptation, image segmentation, and generative models for image synthesis and manipulation.