3D Human Avatar Digitization from a Single Image

Zhong Li[*]
OPPO US Research Center
Palo Alto, CA, USA

Lele Chen[*†]
University of Rochester
Rochester, NY, USA

Celong Liu
OPPO US Research Center
Palo Alto, CA, USA

Yu Gao
OPPO US Research Center
Palo Alto, CA, USA

Yuanzhou Ha
OPPO US Research Center
Palo Alto, CA, USA

Chenliang Xu
University of Rochester
Rochester, NY, USA

Shuxue Quan
OPPO US Research Center
Palo Alto, CA, USA

Yi Xu
OPPO US Research Center
Palo Alto, CA, USA

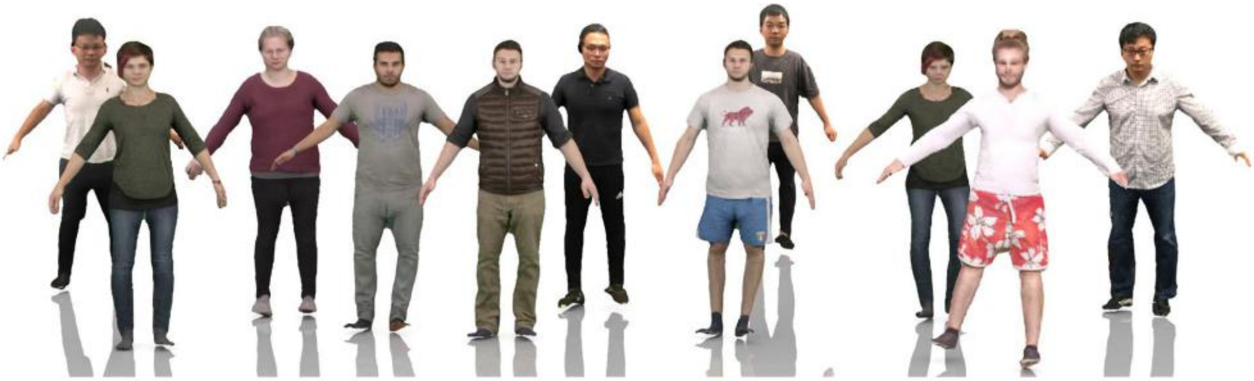

Figure 1: Our approach reconstructs 3D human avatar from a single image. The figure shows our reconstruction results on our own captured data and PeopleSnapshot dataset [Alldieck et al. 2018b].

ABSTRACT

With the development of AR/VR technologies, a reliable and straightforward way to digitize three-dimensional human body is in high demand. Most existing methods use complex equipment and sophisticated algorithms. This is impractical for everyday users. In this paper, we propose a pipeline that reconstructs 3D human shape avatar at a glance. Our approach simultaneously reconstructs the three-dimensional human geometry and whole body texture map with only a single RGB image as input. We first segment the human body part from the image and then obtain an initial body geometry by fitting the segment to a parametric model. Next, we warp the initial geometry to the final shape by applying a silhouette-based dense correspondence. Finally, to infer invisible backside texture from a frontal image, we propose a network we call InferGAN. Comprehensive experiments demonstrate that our solution is robust and effective on both public and our own captured data. Our human avatars can be easily rigged and animated using MoCap data. We developed a mobile application that demonstrates this capability in AR/VR settings.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Mesh models**; **Mesh geometry models**.

KEYWORDS

Human body modeling, 3D reconstruction, Augmented Reality, Deep Learning

ACM Reference Format:
Zhong Li, Lele Chen*, Celong Liu, Yu Gao, Yuanzhou Ha, Chenliang Xu, Shuxue Quan, and Yi Xu. 2019. 3D Human Avatar Digitization from a Single Image. In *The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI '19), November 14–16, 2019, Brisbane, QLD, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3359997.3365707

[*] Both authors contributed equally to this work.
[†] Work was done while Lele Chen was an intern at OPPO US Research Center, Palo Alto, CA, USA

VRCAI '19, November 14–16, 2019, Brisbane, QLD, Australia
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7002-8/19/11…$15.00
https://doi.org/10.1145/3359997.3365707

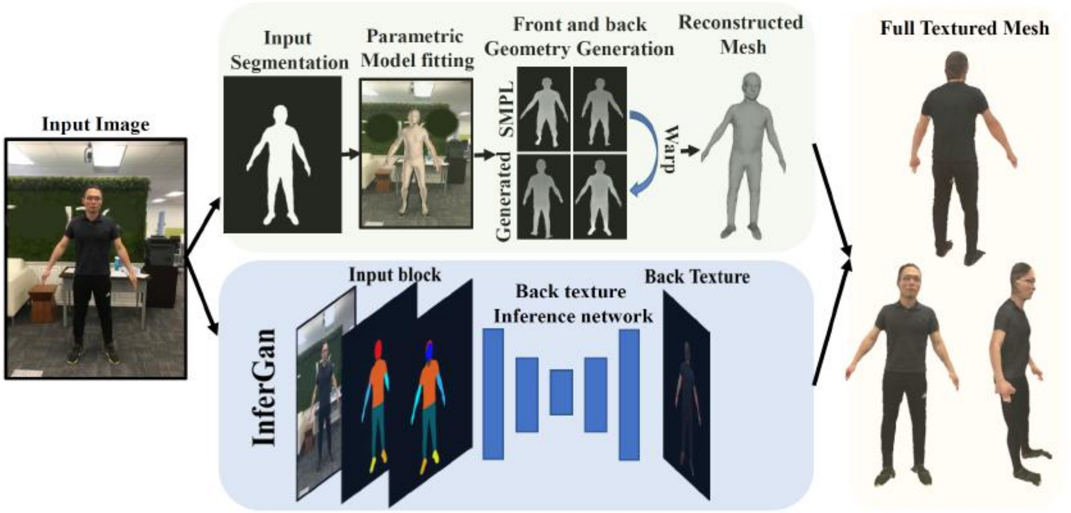

Figure 2: The pipeline of our approach. We reconstruct the geometry and texture respectively and finally generate the full textured result.

1 INTRODUCTION

There is an emerging trend on the automated acquisition of detailed 3D human shape and appearance in both academic community and industry. The generated free-viewpoint video (FVV) can provide the user an immersive viewing experience in many applications such as AR/VR, gaming, virtual try-on, etc. This technology is largely enabled by the availability of bulky 3D acquisition systems and sophisticated reconstruction algorithms. Early work by Kanade and Narayanan [Kanade and Narayanan 2007] used a dome with a diameter of 5 meters and mounted 51 cameras on it to digitize real objects into FVV. Recent capturing setups tend to use industry-level synchronized cameras with higher resolution and speed. For example, the CMU Panoptic studio [Joo et al. 2015] consists of 480 VGA cameras, 31 HD cameras, and 10 Kinect sensors to reconstruct and recover multiple human activities. Industry solutions like Microsoft Holoportation [Orts-Escolano et al. 2016] and 8i [https://8i.com/ [n. d.]] utilize infrared structured light for high-resolution capture with much fewer cameras as well. Despite the high quality of dome capture systems, it is unpractical to use such complicated setup in everyday scenarios. Therefore, there is a need for a simple and easy-to-use setup—one that requires only one photo to digitize 3D human body.

Indeed, many works are on recovering the pose and approximate shape of the human body from one or a few photos, but few address the problem of high-precision reconstruction. Xu et al. [Xu et al. 2018] use a non-rigid body deformation method to reconstruct the human body from a video clip, but it requires a pre-captured template. Alldieck et al. [Alldieck et al. 2018b] reconstruct a high-quality 3D human body, but their method takes long processing time to obtain a complete model of a human body (e.g., 1 minute per frame and with 20 frames). Recently, CNN-based methods [Alldieck et al. 2019b; Saito et al. 2019] achieve high-resolution results by training with a large number of synthetic images. Due to bias in data, these methods do not apply to all types of input. Besides, since only one image is used as input, even full human body is reconstructed, only part of the geometry has color information. Natsume et al. [Natsume et al. 2019] use a lot of synthetic pictures to train a model to infer occluded colors; however, the resulting color lacks photorealism.

To address these challenges, we propose a new pipeline that reconstructs both human geometry and full textures based on a single frontal image of the human body as input (Fig. 2). In the first step, an input image is segmented, and a SMPL [Loper et al. 2015] model is fitted to the body shape segmentation. Since the SMPL model does not align well with the input silhouette, we propose to deform the SMPL model. We do this by finding correspondences between the silhouette of fitted SMPL on the camera and the input 2D silhouette. We then warp the depth map of the SMPL model to its final shape. The back geometry can be reconstructed similarly, and the two pieces of geometry can be stitched together. To recover the invisible texture on the back, we develop a network we call InferGAN. Our model does not need strictly matched front and back color data and can use any pair of photos taken by the same person at different view angles for training. Finally, we use linear blend skinning to animate the reconstructed model. We transfer the skinning weight of SMPL to the reconstructed model and animate it with the existing motion capture data [Gross and Shi 2001]. Our key contributions are as follows:

- a 2D non-rigid deformation algorithm to warp initial geometry to final reconstruction,
- a network we call InferGAN that can predict invisible color on the back of a human,
- and a complete system that reconstructs a full body-colored human model at a glance.

We have conducted comprehensive experiments on existing data and our own captured data (Fig. 1). We demonstrated that our results have good visual quality in terms of geometry and color information.

We also developed a mobile app with ARCore to demonstrate its application in AR.

2 RELATED WORK

The research of 3D human reconstruction is a very broad field, which includes human body pose estimation, single image 3D human body reconstruction, and video-based human body reconstruction. It also involves inference of occluded/invisible texture color. In the following, we will review these methods briefly.

2.1 Human pose estimation

The human pose is, in general, ambiguous on a single image. Therefore, existing methods usually rely on a parametric model of the human body (e.g., SMPL [Loper et al. 2015] or SCAPE [Anguelov et al. 2005]) for fitting a pose to an input image. Early works generally used manual or semi-automatic methods to label human body joints [Guan et al. 2009; Jain et al. 2010; Zhou et al. 2010]. This step later became automated. In recent years, parameters of the SMPL model can be computed by deep neural networks [Kanazawa et al. 2018a; Omran et al. 2018; Pavlakos et al. 2018]. Although the human pose estimation algorithms are evolving, they are struggling to provide detailed body geometry, and cannot fit the model to the contours of a human body accurately.

2.2 Image-based human body reconstruction

Although parametric model-fitting methods capture the shape and pose of the human body well, they do not conform to the actual body contour and fail to recover the details of clothing. To solve this problem, [Alldieck et al. 2018b] extracts mask information from a video of human performance and then uses a silhouette-based visual hull method to reconstruct the body geometry. [Alldieck et al. 2018a] further enhances the previous method by providing better details of clothes, face, and texture. [Alldieck et al. 2019a] uses deep learning plus differentiable render to generate human geometry from video input. The network can be trained end-to-end.

Different from video input, more and more methods begin to reconstruct human body with one single image. [Natsume et al. 2019] uses a neural network to infer different angles of the same person from a single picture and uses the inferred images to reconstruct human body geometry. More recently, [Weng et al. 2019] uses a traditional fitting model plus silhouette warping to estimate an approximate body model. However, for invisible backside texture, this work simply uses a mirror of the frontal texture, which makes the results less realistic.

2.3 Texture inference

When reconstructing a 3D model from a single image, the color information of occluded or invisible parts is hard to obtain. However, we can use a learning-based method to infer the missing data from the frontal view. This kind of problem attributes to the research [Park et al. 2017; Zhou et al. 2016] of single image view synthesis. Given a frontal view of the human body, [Natsume et al. 2019] can predict the backside color texture. However, since the training data is synthetic, their results lack photorealism. Generative models have been used to synthesize novel views of human bodies [Chan et al. 2019; Ma et al. 2017; Sun et al. 2019]. The generated images in these methods are based on pose-to-appearance mapping, not on the segmentation of body parts; therefore, they cannot be used to texture the body geometry directly.

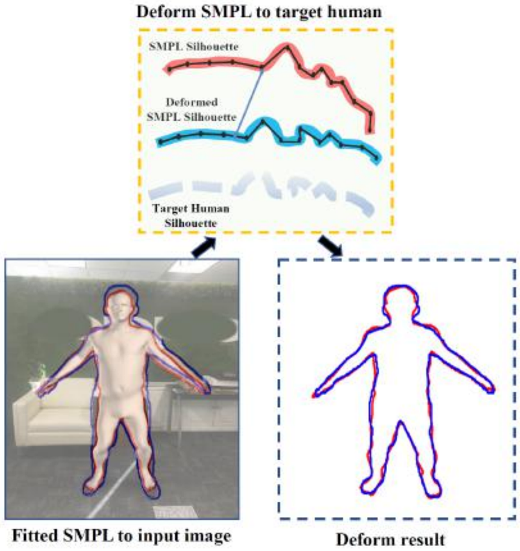

Figure 3: 2D Non-rigid Registration. We aim to find correspondence between the SMPL silhouette (red) and segmented human silhouette (blue). The 2D embedded deformation is illustrated in upper row of the figure.

3 METHOD OVERVIEW

Our algorithm consists of four major steps. As shown in Fig 2, given a single RGB image, we first segment the human body shape using the state-of-the-art human garment segmentation method [Gong et al. 2019]. Then, we fit a SMPL [Loper et al. 2015] deformable parametric model to the body shape segment. To generate the front geometry, we warp the depth map of the SMPL parametric model using correspondence defined based on silhouettes. We then generate the back geometry using back-culling rendering technique. Finally, to recover the occluded back texture, we propose a GAN network structure called InferGAN, which is trained on both real and synthetic images.

4 BODY GEOMETRY RECONSTRUCTION

After segmenting human shape from image using [Gong et al. 2019], we use the method in [Kanazawa et al. 2018b] to fit a SMPL model to the input RGB image. As shown in Fig. 3, the recovered SMPL mesh provides a good initial guess for the shape of the reconstructed human. However, its contour is not consistent with body shape silhouette from the input image. Similar to [Weng et al. 2019], we propose a method to refine the human mesh by finding the correspondence between the depth map of initial SMPL estimation and that of recovered human body. Specifically, instead of warping in 3D, we use a 2D approach. We first deform the person silhouette from input image to match the SMPL silhouette using a 2D non-rigid registration approach. The nearest point from the SMPL silhouette

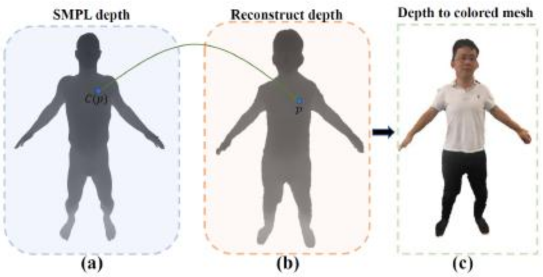

Figure 4: Geometry generation approach. For every point p inside the person mask, we compute its corresponding point $C(p)$ in the SMPL mask. The third column shows the reconstructed result.

to a point on the person silhouette is used to guide the deformation. The registration process produces a warping function that can be used to warp depth map of SMPL model to the final resulting shape. We apply the warping function to both front and back views. The two recovered meshes agree with the silhouettes. In the following, we will discuss the algorithms in details.

4.1 2D Non-Rigid Registration

In order to apply warping function to reconstruct the final shape, we first register the person silhouette to SMPL silhouette. Given a source person silhouette S and SMPL silhouette T, S has κ_S vertices $\{s_i|_{i=1,\dots,\kappa_S}\}$, and T has κ_T vertices $\{t_i|_{i=1,\dots,\kappa_T}\}$, where $s_i, t_i \in \mathbb{R}^2$. We then uniformly sample a set of m graph nodes $G = \{g_1, g_2, ..., g_m\}$ on the silhouette S. We then use a deform graph to represent the movement of silhouette. More specifically, our goal is to solve a set of affine transformations $\boldsymbol{A} = \{\boldsymbol{A}_t\}_{t=1}^m$ and $\boldsymbol{b} = \{\boldsymbol{b}_t\}_{t=1}^m$ that parametrize the movement of the the graph node. After deformation, the new position of a vertex can be written as:

$$s' = f(s, \boldsymbol{A}, \boldsymbol{b}) = \sum_{t=1}^{m} \varpi_t(s)[\boldsymbol{A}_t(s - g_t) + g_t + \boldsymbol{b}_t] \ , \tag{1}$$

where $\varpi_t(s)$ is the weighing factor of a graph node g_t on the silhouette S. In particular, $\varpi_i(s_1) = \max(0, (1 - d(s_1, g_i)^2/R^2)^3)$, where $d(s_1, g_i)$ is the geodesic distance between point s_1 and g_i along the 2D contour line. R is the distance between s_1 and its k nearest neighbor graph nodes in geodesic domain. We use $k = 4$ in our experiment.

Once we have constructed the deform graph, to deform the person silhouette S to match the SMPL silhouette T, we minimize the following energy function to solve for a set of A and b:

$$E_{\text{total}} = \lambda_{\text{rigid}} E_{\text{rigid}} + \lambda_{\text{smooth}} E_{\text{smooth}} + E_{\text{fit}} \tag{2}$$

Term E_{rigid} enforces rigidity and property of the rotation matrix, and thus is defined as:

$$E_{\text{rigid}} = \sum_{\text{G}} ||A_i^T A_i - \mathbb{I}||_F^2 + (\det(A_i) - 1)^2 \ , \tag{3}$$

where $\mathbb{I}$ is the identity matrix.

The second term E_{smooth} enforces spatial smoothness of the geometric deformation and it is defined as:

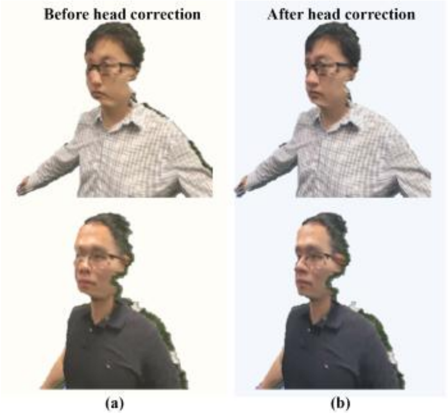

Figure 5: Head correction results. Left column shows the results before correction, and right column shows the results after correction. We correct the misalignment in face region.

$$E_{\text{smooth}} = \sum_{\text{G}} \sum_{k \in \Omega(i)} ||A_i(g_k - g_i) + g_i + b_i - (g_k + b_k)||^2 \ , \tag{4}$$

where $\Omega(i)$ refers to node i's k nearest neighbors.

Lastly, the data term E_{fit} is similar to a 2D form of Iterative Closest Point (ICP), which measures the vertex displacements between the source and target silhouette line segments. The data term includes two parts: point-to-point distances and point-to-plane distances:

$$\boldsymbol{E}_{\text{fit}} = \sum_{i \in P} \lambda_{\text{point}} ||s_i' - t_{s_i}||^2 + \lambda_{\text{plane}} ||n_i^T (s_i' - t_{s_i})||^2 \ , \tag{5}$$

where $s_i' = f(s_i, a, b)$, t_{s_i} is the closest point of s_i in the target SMPL silhouette. n_i^T is the corresponding normal of t_{s_i}. In our experiments, we use $\lambda_{\text{point}} = 0.1$ and $\lambda_{\text{plane}} = 1$.

The overall energy function E_{total} can be optimized using iterative Gauss-Newton algorithm. We use $\lambda_{\text{rigid}} = 50$ and $\lambda_{\text{smooth}} = 25$ in our experiments. The process is shown in Fig. 3.

4.2 Front Mesh Generation

Once we have a mapping M between person silhouette and fitted SMPL silhouette, we can generate the front mesh. For each s_i in $S = \{s_1, s_2, ..., s_\kappa\}$, we have a corresponding point on SMPL silhouette boundary $M(S) = \{M(s_1), M(s_2), ..., M(s_\kappa)\}$. Next, for each pixel p inside the person mask, we find its corresponding pixel $C(p)$ inside the SMPL mask. In order to compute the dense correspondence, we construct a function that transfers the silhouette correspondence to inside-mask correspondence. Similar to [Weng et al. 2019], we use Mean Value Coordinates (MVC) [Floater 2003] as our warping function. MVC expresses a point inside a planar triangulation as a convex combination of its boundary points. More specifically, we can represent point p using a weighted combination of the set of vertices of person silhouette:

$$p = \sum_{i=1}^{\kappa} w_i(p) s_i \ . \tag{6}$$

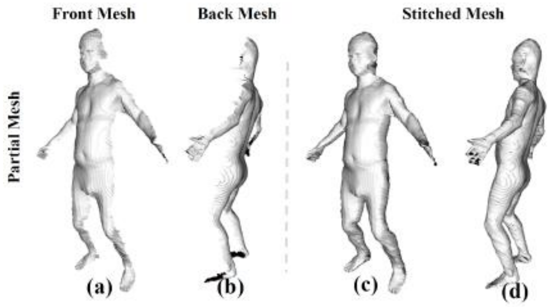

Figure 6: Stitch the front and back meshes. We show front and back meshes on the left, and the stitched one on the right.

Next, with already computed correspondence M and MVC function described above, we substitute s_i with $M(s_i)$. The warping function is represented as below:

$$C(p) = \sum_{i=1}^{\kappa_S} w_i(p) M(s_i) \ . \tag{7}$$

As shown in Fig 4, for every pixel p inside the person mask, we can compute its corresponding pixel $C(p)$ inside the SMPL mask. We apply this warp function to the SMPL depth map to compute the depth map of our final front mesh:

$$Z(p) = Z_{\mathrm{SMPL}}(C(p)) \ . \tag{8}$$

Once we obtain per-pxiel depth, we build the triangular mesh using geometry constraint. The process of our reconstruction is shown in Fig. 4.

Face Alignment: As shown in Figure 5 (left), after warping, the face part might suffer from distortion. It is especially obvious if viewpoint is far away from the frontal camera view. During the parametric model fitting process, although the general fitting on body shapes (e.g., torso, upper body, lower body) is reasonable, the prediction of the head pose is often inaccurate due to the fact that fitting method does not consider facial landmarks during optimization process.

We therefore develop a two-step approach to fix the face geometry. We first roughly align the head pose after parametric model fitting. Given image I, we first detect person face region and subsequently detect 2D facial landmarks. In our experiment, we use 7 landmarks (corners of two eyes, nose, corners of mouth). Then, we solve the 3D head pose R_{head} by minimizing the re-projection error of 3D landmark points on image I:

$$\min \ ||Proj(R_{\mathrm{head}} \cdot \chi) - \chi_{\mathrm{2d}}|| \ , \tag{9}$$

where $Proj$ is the projection from 3D world coordinate to 2D camera coordinate, χ is the set of 3D landmarks, and χ_{2d} are their 2D projections on image I.

4.3 Front and Back Geometry Stitching

After constructing the front mesh, we aim to recover the back geometry of the person, and finally stitch front and back pieces together. To generate back geometry with a matching silhouette, an intuitive idea is to set the virtual camera looking at the back of the person, and render the fitted SMPL model. However, with perspective projection, the sillouette of the back rendering will not be the same as front view. We instead use back-face culling technique. In particular, we use the same camera viewpoint as rendering the front SMPL model. Then, instead of rendering the nearest triangle, we render the farthest triangle to obtain the corresponding silhouette image and mask from the back view as shown in Fig. 6. We then stitch the front piece and back piece together by creating connecting triangles along the two boundaries.

4.4 Animation Reconstruction

To animate reconstructed human mesh, we can transfer the parametric model's skinning map W_{SMPL} using the warping function described in Equation 7. That is:

$$W(p) = W_{\mathrm{SMPL}}(C(p)) \ . \tag{10}$$

5 MULTI-VIEW TEXTURE INFERRING

To recover the front texture, we project the image onto the geometry. For the occluded back part, we propose an automated process named InferGAN. The InferGAN transfers the input texture from one input RGB image (I_o) to another RGB image (I_t) based on the input body parsing audiences P_o and P_t. The intuitive assumption is that, in latent space, the distance between I_o and I_t should approximately equal to the distance between P_o and P_t. Mathematically, we can formulate the assumption as:

$$\Theta_{\mathrm{img}}(I_o) - \Theta_{\mathrm{img}}(I_t) \approx \Theta_{\mathrm{seg}}(P_o) - \Theta_{\mathrm{seg}}(P_t) \ , \tag{11}$$

$$I_t \approx \Theta^R(\Theta_{\mathrm{img}}(I_o) + (\Theta_{\mathrm{seg}}(P_o) - \Theta_{\mathrm{seg}}(P_t))) \ , \tag{12}$$

where Θ_{img}, Θ_{seg} are image encoder and parsing encoder, respectively. The Θ^R is an image decoder based on the convoluted latent feature. To save computing resources, we combine Θ_{img} and Θ_{seg} together into Θ. Specifically, the InferGAN encodes the texture information from I_o and inpaints new texture pixels into the target body parts (P_t) (e.g., back parts during the inference stage).

We show all the steps of InferGAN in Fig. 7. The size of input/target image is 512 × 512. The yellow box indicates data preprocessing, which extracts clothing parsing segmentation (P_o and P_t) using the method described in [Gong et al. 2019]. Then we transfer the parsing segmentation (P_o and P_t) into one-hot vector map (size of 20 × 512 × 512) since human clothing is defined as 20 parts in [Gong et al. 2019]. We also compute the contours (C_o and C_t) of the human body as inputs to our InferGAN. In our empirical study, we find that the contour (size of 1 × 512 × 512) can help the network to yield sharper boundaries. Then we randomly perform affine transformation on images and vector maps to increase the diversity of geometry. The green box shows the training schema of InferGAN. To save computing resources, we concatenate all the inputs (I_o, P_o, P_t, C_o and C_t) in channel dimension. After concatenation, the input size is 45 × 512 × 512. The encoder Θ, which consists of five convolution blocks, learns the texture information from I_o and learns the geometry information from P_o, C_o and P_t, C_t. Then we pass the latent feature to 9 residual blocks [He et al. 2016] to further increase the encoding capability. The image decoder Θ_R consists of five transpose convolutional layers and one hyper-tangent activation function. The InferGAN is trained with three loss functions: GAN loss, perceptual loss computed on VGG

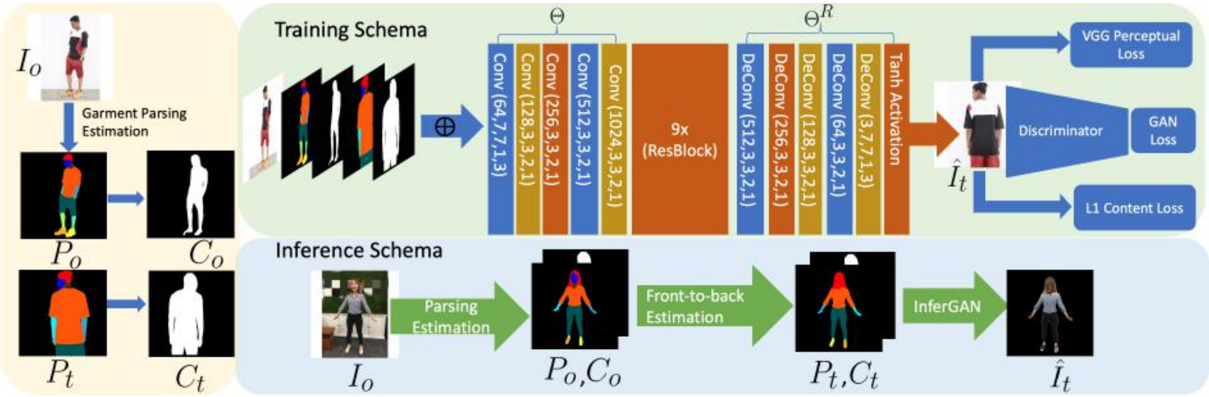

Figure 7: The architecture of InferGAN. The yellow part indicates the data pre-processing. The green box illustrates the training schema and the blue box shows the inference schema.

feature extractor ($\mathcal{F}_{\text{VGG}}$)[Simonyan and Zisserman 2014], and L1 pixel loss. Thus, the total loss can be expressed as:

$$\begin{aligned}\mathcal{L} &= \ell_{\text{gan}} + \ell_{\text{perceptual}} + \ell_{\text{pixel}}\\ &= \mathbb{E}_{I_t,P_t,C_t}[\log \text{D}(P_t, C_t, I_t)]+\\ &\quad \mathbb{E}_{I_o,P_o,C_o,P_t,C_t}[\log(1 - \text{D}(P_t, C_t, \text{G}(I_o, P_o, C_o, P_t, C_t))]+\\ &\quad \sum_{n=1}^{5} \frac{1}{2^n} \times \|\mathcal{F}^n_{\text{VGG}}(\text{G}(I_o, P_o, C_o, P_t, C_t)) - \mathcal{F}^n_{\text{VGG}}(I_t)\|^1_1+\\ &\quad \|\text{G}(I_o, P_o, C_o, P_t, C_t)) - I_t\|^1_1 \quad , \end{aligned} \tag{13}$$

where D and G are discriminator and generator, respectively. The $\ell_{\text{perceptual}}$ is computed on the output features of five different layers of VGG19. $\mathcal{F}^n_{\text{VGG}}(I_t)$ is the layer feature of VGG19 on image I_t. We give different scales to different layer features. We found that GAN loss and perceptual loss will enforce the network to yield a sharper image since we compute the loss in feature space. It is worth noting that during training stage, the input image and ground truth image do not need to be a front-view image and back-view image. The blue box in Fig. 7 indicates the inference stage for our back-view texture synthesizing using InferGAN. We first estimate the clothing parsing P_o and contour C_o from input image I_o. Then we estimate the back-view clothing parsing and contour $(\hat{P_t}, \hat{C_t})$ based on the computed P_o. Then we pass all the inputs to InferGAN to synthesize the back-view texture $\hat{I_t}$, which is used to texture the back geometry of a human.

6 EXPERIMENTS

We tested our method on public data and our own collected data to demonstrate the reliability and effectiveness of our approach. For our own data, we use a mobile phone camera at a resolution of 4032× 3024. A-pose is used since it is convenient for subsequent animation reconstruction. We deployed our reconstruction algorithm on a server. The reconstruction time is about 1 minute on a PC with CPU Intel Core i7-5820K, 32GB memory and a Titan X GPU.

6.1 Geometry reconstruction

The reconstructed model is shown in Fig. 9. The first column shows input images, the second column shows the reconstructed geometry of human body models, and the third and fourth columns show the results of observing the model from different angles after applying full texture. Our algorithm restores the shape of the human body accurately. The reconstructed geometry silhouette fits the input image very well. We also reconstruct the texture from one single image as input.

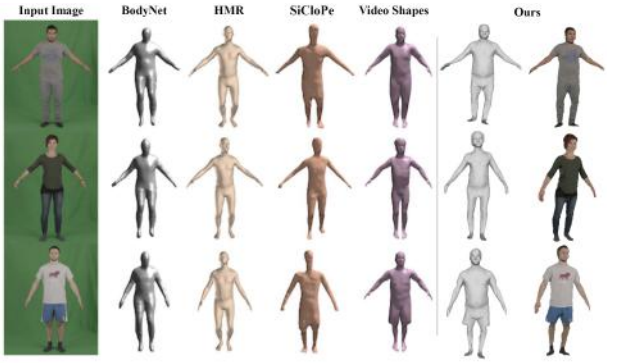

Figure 8: Visual comparison with other human shape reconstruction methods. From left to right: input images, BodyNet [Varol et al. 2018], HMR [Kanazawa et al. 2018b], SICLOPE [Natsume et al. 2019], Video Shapes [Alldieck et al. 2018b], and ours.

We compare our method with other 3D human body reconstruction methods on PeopleSnapshot dataset [Alldieck et al. 2018b]. Bodynet [Varol et al. 2018] is a voxel-based method to estimate the pose and shape of the human body. SICLOPE [Natsume et al. 2019] relies on synthetic masks from difference views to reconstruct human shape details. HMR [Kanazawa et al. 2018a] estimates the pose and details of the human body from the SMPL parametric model. Video-based method [Alldieck et al. 2018b] uses 120 images of the same person at different angles to merge into a complete human body model. But this method slows down the whole process by optimizing the pose calculated on each frame. In Fig. 8, we show side-by-side comparison with all the above-mentioned methods. Our results have more details than the first three methods. When compared with the results of [Alldieck et al. 2018b] using 120

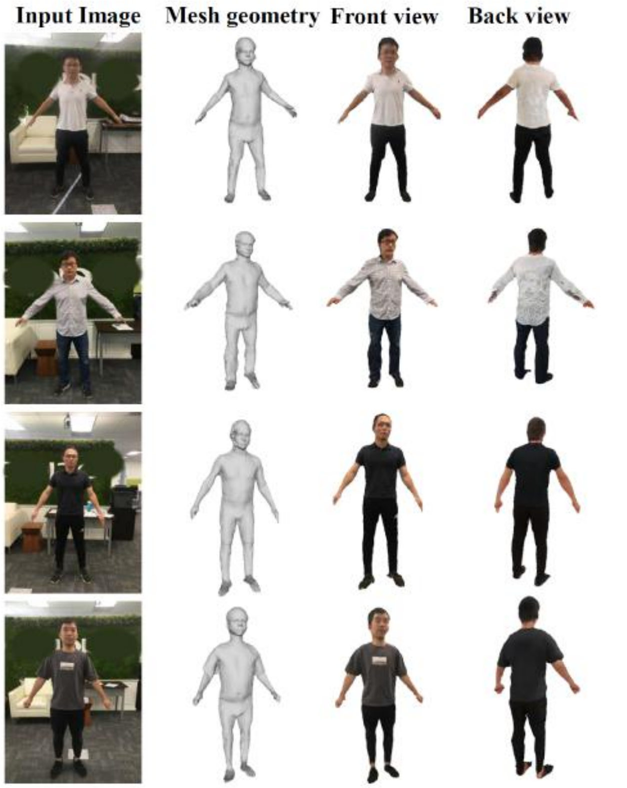

Figure 9: 3D human digitization results. The first column shows input images, second column shows reconstructed geometry, and last two columns are textured mesh from different views.

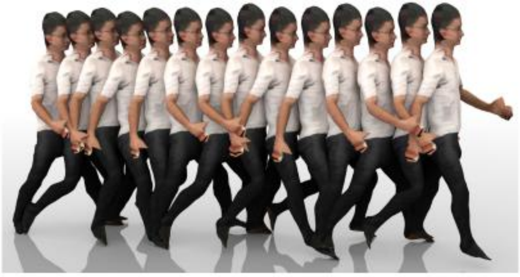

Figure 10: Animation result. The reconstructed model is jogging. We demonstrate that the animation deformation looks plausible and sufficient for certain AR/VR applications.

frames as input, our results are comparable, but with much lower computational cost.

To animate our model, we transfer the SMPL parameters to the reconstructed model and apply MoCap data from CMU dataset [Gross and Shi 2001]. Fig. 10 shows an animation sequence when We apply jog sequence to the model.

6.2 Texture inference

We train and test our InferGAN module on DeepFasion dataset [Liu et al. 2016]. We follow the training set/testing set split in [Liu et al. 2016]. The learning rate for InferGAN is 0.0002. We adopt Adam [Kingma and Ba 2014] optimizer (β_1 = 0.5 and β_2 = 0.999) in all experiments. In addition, random cropping, affine transformation and flipping are used to augment data. The network is trained for 20 epochs.

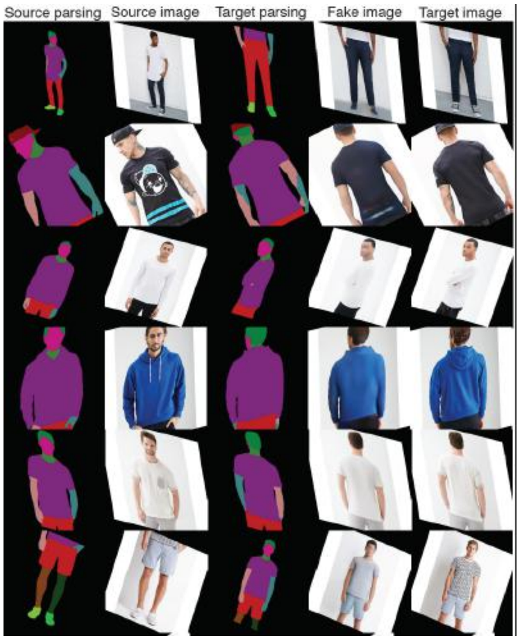

Figure 11: The samples randomly selected from testing set results synthesized by InferGAN. The first two column are input source parsing and input source image, respectively. The identities in testing set is excluded from the training set.

Fig. 11 shows qualitative results of InferGAN on the testing set of DeepFasion dataset. We can infer the target-view image texture by any input-view image texture. For instance, our InferGAN can yield realistic bottom textures based the whole body source input. Second row and fourth row shows the back-view results inferred by front-view image. It is worth noting that in the sixth row, the input image is bottom part and the target image is whole body. Our InferGAN can synthesize reasonable texture for upper part while keeping the original texture for bottom part. Since the goal of our InferGAN is to synthesize missing textures in back-view for our whole system, we do not regularize the quality of the synthesized face region (e.g., third row and fifth row).

6.3 AR/VR Application

We also developed an application on mobile platform. We first use the mobile phone to take a photo of the human body. Then, the application sends the photo to a server for reconstruction process. Finally, the server sends the reconstructed models back to the mobile phone for animation sequence playback. As shown in Fig 12, we use Google Inc.'s ARCore platform to place the virtual 3d model in a real scene. Please see the supplementary material for more videos of dynamic models.

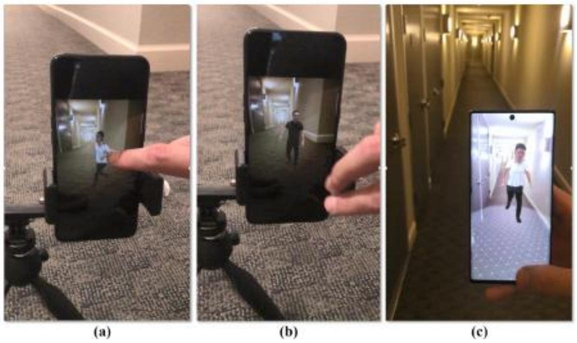

Figure 12: AR/VR application on mobile phone. (a) and (b) shows two animated models rendered in real environment. (c) With ARCore, we can view the animated model from free viewpoint.

7 CONCLUSION

In this paper, we present a complete pipeline that reconstructs human body from a single image. We propose to use 2D non-rigid registration to warp geometry from an initial estimate to the final reconstruction. For texture extraction, we propose InferGAN, which uses front texture to infer textures at different view angles. Our method not only reconstructs the full geometry and detailed texture of a human body using only one single image, but also obtains an animated model by the weight and joint locations of the parametric transfer model. We have also developed a mobile application to showcase our capability, where the reconstruction pipeline sits on a server.

Our approach has some limitations. First, we need the human subject to face the camera so that we can capture a frontal view of the person. Second, we require that the subject's limbs and body are occlusion-free. Finally, since SMPL is a skinned human body model, certain garment details are lost during reconstruction (e.g., shoes). In the future, we aim to develop an end-to-end pipeline that is more general and can handle occlusion cases.

REFERENCES

Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. 2019a. Learning to Reconstruct People in Clothing from a Single RGB Camera. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1175–1186.

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018a. Detailed Human Avatars from Monocular Video. In *International Conference on 3D Vision*. 98–109. https://doi.org/10.1109/3{DV}.2018.00022

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018b. Video based reconstruction of 3d people models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8387–8397.

Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. 2019b. Tex2Shape: Detailed Full Human Body Geometry from a Single Image. *arXiv preprint arXiv:1904.08645* (2019).

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: shape completion and animation of people. In *ACM transactions on graphics (TOG)*, Vol. 24. ACM, 408–416.

Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. 2019. Everybody Dance Now. *International Conference on Computer Vision (ICCV)* (2019).

Michael S Floater. 2003. Mean value coordinates. *Computer aided geometric design* 20, 1 (2003), 19–27.

Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. 2019. Graphonomy: Universal Human Parsing via Graph Transfer Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7450–7459.

Ralph Gross and Jianbo Shi. 2001. The cmu motion of body (mobo) database. (2001).

Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. 2009. Estimating human shape and pose from a single image. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 1381–1388.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

https://8i.com/. [n. d.]. Real human holograms for augmented, virtual and mixed reality. *Accessed:2017-10-03* ([n. d.]).

Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. 2010. Moviereshape: Tracking and reshaping of humans in videos. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 148.

Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. 2015. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*. 3334–3342.

Takeo Kanade and PJ Narayanan. 2007. Virtualized reality: perspectives on 4D digitization of dynamic events. *IEEE Computer Graphics and Applications* 27, 3 (2007), 32–40.

Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. 2018a. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7122–7131.

Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2018b. End-to-end Recovery of Human Shape and Pose. In *Computer Vision and Pattern Regognition (CVPR)*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. 2016. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)* 34, 6 (2015), 248.

Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. 2017. Pose guided person image generation. In *Advances in Neural Information Processing Systems*. 406–416.

Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. 2019. Siclope: Silhouette-based clothed people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4480–4490.

Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. 2018. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *2018 International Conference on 3D Vision (3DV)*. IEEE, 484–494.

Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. 2016. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 741–754.

Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. 2017. Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of the ieee conference on computer vision and pattern recognition*. 3500–3509.

Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. 2018. Learning to estimate 3D human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 459–468.

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. *arXiv preprint arXiv:1905.05172* (2019).

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

Wei Sun, Jawadul H. Bappy, Shanglin Yang, Yi Xu, Tianfu Wu, and Hui Zhou. 2019. Pose Guided Fashion Image Synthesis Using Deep Generative Model. In *Proceedings of KDD 2019 Workshop AI for Fashion*.

Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. 2018. Bodynet: Volumetric inference of 3d human body shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 20–36.

Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. 2019. Photo wake-up: 3d character animation from a single photo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5908–5917.

Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. 2018. Monoperfcap: Human performance capture from monocular video. *ACM Transactions on Graphics (ToG)* 37, 2 (2018), 27.

Shizhe Zhou, Hongbo Fu, Ligang Liu, Daniel Cohen-Or, and Xiaoguang Han. 2010. Parametric reshaping of human bodies in images. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 126.

Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. 2016. View synthesis by appearance flow. In *European conference on computer vision*.

Springer, 286–301.