PARTE: Part-Guided Texturing for 3D Human Reconstruction from a Single Image

Supplementary Material

Figure S1. **Example of part texture editing.**

In this supplementary material, we present additional technical details and more experimental results that could not be included in the main manuscript due to page limitations. The contents are summarized below:

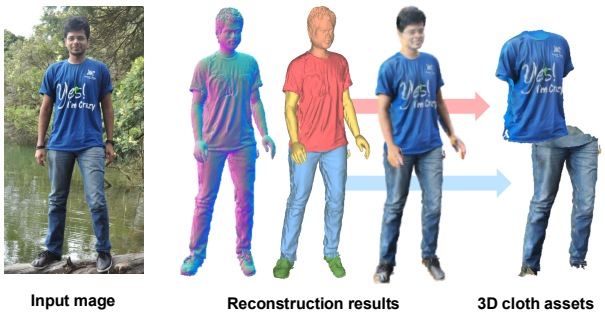


Figure S2. **Example of 3D cloth decomposition.**

S1. Additional applications

Part texture editing. Fig. S1 shows that our framework allows part-aware texture editing, allowing modifications to a specific human part. Given a 3D textured human surface and a part-segmented human surface, PartTexturer can edit one of the part textures. Specifically, after projecting the 3D human surface into 2D space, we can modify the projected image on a target human part via image inpainting methods [68]. Then, the modified image serves as an input for PartDiffusion, which drives PartTexturer. By running PartTexturer, we can obtain a new 3D textured human surface where the target part is updated. Since our proposed PartTexturer takes not only text prompts but also an image as guidance, it enables more precise and detailed 3D editing by referencing the image.

3D cloth decomposition. Fig. S2 shows that our framework enables the decomposition of cloth surfaces from the reconstructed result of our framework. Our framework produces 3D human part segmentation as an intermediate output during reconstruction. Since reconstructed textures are well-aligned with their corresponding parts, 3D cloth surfaces can be obtained by cutting their regions based on the part segmentation. This decomposition is made possible because our framework provides accurate human part segmentation and ensures the reconstructed human textures are aligned with the part segmentation.

S2. Plug-in for other reconstruction methods

Tab. S1 shows that integrating our PARTE with various 3D human reconstruction methods, including 2K2K [23], SiTH [27], HumanRef [94], and SIFU [97], improves texture quality. In this experiment, we utilize the geometry outputs from the 3D human reconstruction methods and apply our framework for part-guided texturing. In the results, our PARTE achieves superior texture reconstruction compared to the original texturing of each method. Our frame-

Figure S3. **Qualitative comparisons of PartDiffusion with other diffusion models for texturing on GT human geometry.**

work can be integrated into various 3D human reconstruction pipelines in a plug-and-play manner, enhancing texture quality without altering or modifying the geometry reconstruction process.

S3. Texturing based on GT human geometry

Fig. S3 and Tab. S2 demonstrate that our framework achieves superior texture reconstruction compared to other diffusion models when evaluated on GT human geometry of THuman2.1 [91]. To evaluate texture reconstruction while eliminating the influence of geometric errors, we compare our framework with other texturing approaches based on GT human geometries. Specifically, we remove textures from the GT human geometries of THuman2.1 and use them as inputs to texturing frameworks. As a result, our framework outperforms other diffusion models in both texture fidelity and alignment across human parts.

S4. More ablation studies

Effectiveness of SegmentNet design. Fig. S4 shows that incorporating front-view part segments enhances the part segmentation. The reconstructed 3D textureless human surface exhibits indistinct boundaries between different human part regions, making it challenging to accurately segment each part. Accordingly, without front-view part segments, SegmentNet produces incorrect 2D part segments, which lead to failures in 3D part segmentation. To address this, we incorporate front-view part segments for the segmentation, which capture semantic cues that are not explicitly represented in the normal map. By leveraging these additional semantic cues, our approach enables more accurate part segmentation.

Methods	Texture reconstruction		
	PSNR↑	LPIPS↓	Part IoU↑
2K2K [23]	20.373	0.131	0.515
2K2K [23] + PARTE(Ours)	**20.692**	**0.128**	**0.574**
SiTH [27]	20.692	0.120	0.535
SiTH [27] + PARTE(Ours)	**21.449**	**0.108**	**0.585**
HumanRef [94]	21.302	0.113	0.576
HumanRef [94] + PARTE(Ours)	**22.153**	**0.101**	**0.623**
SIFU [97]	21.491	0.108	0.588
SIFU [97] + PARTE(Ours)	**22.412**	**0.095**	**0.639**
TeCH [34]	21.089	0.108	0.588
TeCH [34] + PARTE(Ours)	**22.175**	**0.096**	**0.641**

Table S1. **Impact of applying PARTE to different 3D reconstruction methods on THuman2.1 [91].**

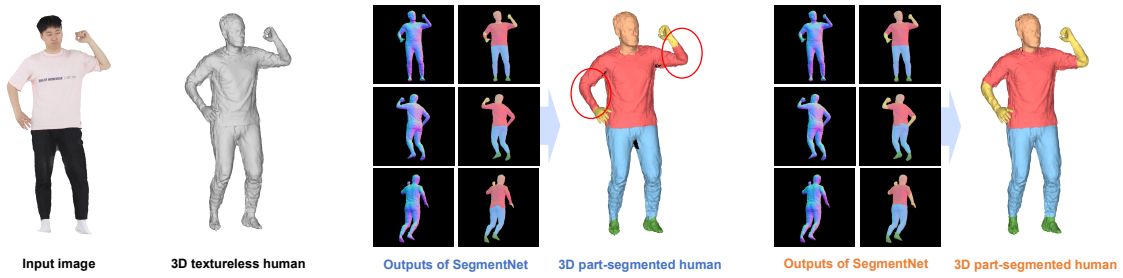
Methods	Texture reconstruction		
	PSNR↑	LPIPS↓	Part IoU↑
StableDiffusion [68] (DreamFusion [63])	27.422	0.048	0.772
Reference U-Net (HumanRef [94])	27.659	0.042	0.815
DreamBooth [70] (TeCH [34])	28.337	**0.039**	0.835
PartDiffusion (PartTexturer, Ours)	**29.315**	**0.039**	**0.857**

Table S2. **Comparisons of texturing results between different diffusion models based on textureless GT human geometry of THuman2.1 [91].**

Effectiveness of PartDiffusion design. Tab. S3 and Fig. S7 show that PartDiffusion effectively generates human images that are accurately aligned with both the input image and part segments, compared to other diffusion networks. For quantitative comparison, we measure PSNR, LPIPS, and Part IoU between generated images and GT counterparts. All other diffusion networks except PartDiffusion struggle to preserve both the human part structure and human appearance from the input image. This limitation leads to inconsistent 3D human texturing, resulting in misaligned textures across human parts. In contrast, our PartDiffusion effectively integrates the input image and part segments, ensuring proper part alignment while generating visually coherent human images. This indicates that PartDiffusion possesses precise prior knowledge of both human part structure and human appearance, enabling more accurate 3D human texturing in PartTexturer. Fig. S5 additionally shows that PartDiffusion is capable of generating human images while preserving the appearance of the input image, even when given in-the-wild images with diverse clothing styles.

S5. Implementation details

We provide an explanation of the implementation details of PartSegmenter and PartTexturer below. PyTorch [60] is used for all implementations.



(a) w/o front-view image **(b) PartSegmenter (Ours)**

Figure S4. **Ablation study for SegmentNet design.**

Figure S5. **Image generation examples of PartDiffusion.**

S5.1. PartSegmenter

Geometry reconstruction. To reconstruct a 3D textureless human surface from a single image, we employ an off-the-shelf reconstruction method, TeCH [34]. TeCH uses Deep Marching Tetrahedra [74] (DMTet) as a geometric representation of a 3D human. DMTet represents 3D geometry based on a tetrahedral grid structure, where each 3D query point on the tetrahedral grid predicts a signed distance from the 3D geometry surface. Based on the 3D representation, we initially optimize it based on the naked human body, SMPL-X [61] human mesh, which is estimated from PIXIE [18]. After initialization, the geometry is optimized to capture fine human details, with three types of losses: reconstruction loss, SDS loss, and regularization loss. Reconstruction loss is defined as the L2 distance between the normal rendering results of DMTet and the predicted normal maps based on Sapiens [38] normal estimator. SDS loss enforces the geometry's normal rendering results to match the real image knowledge learned by the diffusion model [69]. Regularization loss inhibits implausible geometry through Laplacian smoothing [7]. We used Adam [40] optimizer with a base learning rate 1×10^{-3} with a weight decay of 5×10^{-4}. The optimization was done for 10,000 steps with a single NVIDIA A100 40GB GPU. After the optimization, we convert the DMTet into a textureless 3D human surface with Marching Tetrahedra (MT) [15] algorithm.

3D part segmentation. For 3D part segmentation from a 3D textureless human surface, we first render multiple normal maps from 30 uniformly distributed viewpoints. Then, the normal maps are forwarded into SegmentNet to obtain part segments corresponding to the viewpoints. For the front viewpoint, which aligns with the input image, we utilize the image segmentation method Sapiens [38] instead of SegmentNet. The pixel labels of the part segments are unprojected onto the 3D human surface and used for voting.

Methods	Image generation		
	PSNR↑	LPIPS↓	Part IoU↑
StableDiffusion [68]	10.719	0.491	0.138
InstanceDiffusion [81]	16.125	0.202	0.563
PartDiffusion w/o image segments	15.391	0.234	0.492
PartDiffusion w/o text prompts	19.422	0.134	0.803
PartDiffusion (Ours)	**20.109**	**0.119**	**0.854**

Table S3. **Ablation studies on image generation quality among different diffusion networks on THuman2.1 [91].**

Methods	PSNR↑	LPIPS↓	Part IoU↑
SiTH [23]	20.200	0.155	0.480
HumanRef [80]	20.896	0.143	0.442
TeCH [29]	21.090	0.123	0.489
PARTE (Ours)	**21.698**	**0.113**	**0.512**

Table S4. **Quantitative comparisons with existing 3D human reconstruction methods, on 4D-DRESS [79].**

By aggregating the 30 part segments, we assign each surface vertex the most frequently occurring part label as the final label, resulting in a 3D part-segmented human surface.

Training details of SegmentNet. Our SegmentNet is designed by modifying the off-the-shelf image segmentation network, Sapiens-1b [38]. We apply the publicly released pre-trained weights to all Transformer layers of SegmentNet while keeping them frozen. Then, we insert self-attention layers after the first $L = 10$ Transformer layers out of the total 40 layers in Sapiens. For training SegmentNet, we utilize weighted cross-entropy loss by following Sapiens. Data augmentation, including scaling, rotation, flipping, and color jittering, is performed in training. The weights are updated by AdamW [52] optimizer with a batch size of 2. The initial learning rate is set to 5×10^{-4} and linearly reduced to 0 over training. We train SegmentNet for 5 epochs with a single NVIDIA A100 40GB GPU.

S5.2. PartTexturer

3D human texturing. In 3D human texturing, we optimize an MLP network that predicts an RGB color value at the input 3D coordinate. The MLP network is implemented by using a fully-connected layer with 32 hidden dimension and ReLU activations. It takes 3D coordinates of the human surface as input, after applying the hash positional encoding with a maximum resolution of 2048. In Eq. (2), we use the classifier-free guidance [28] strategy with a guidance scale of 100 for noise estimation. The noise levels are defined at randomly selected timesteps within the range $[0.02, 0.98]$. We used Adam [40] optimizer to optimize the network with an exponentially decaying learning rate starting from 1×10^{-2}. We optimize the network for 4,000 steps with a batch size of 4 on a single NVIDIA A100 40GB GPU.

Figure S6. **Failure case of our proposed framework.**

Training details of PartDiffusion. We use three types of encoders in our PartDiffusion. Part encoder uses a ConvNeXt-T [51] architecture following [81]. For the image encoder, we design a new module which consists of 4 ConvNeXt blocks, where the layer depths of each block are $[3, 3, 3, 1]$ and the feature sizes of each block being $[16, 32, 64, 12]$. The prompt encoder follows the structure of CLIP [66] encoder. For training PartDiffusion, we adopt the pre-trained weights of InstanceDiffusion [81] as the initial weights for the part encoder and the subsequent self-attention layer. The prompt encoder and all layers of the diffusion network are initialized with pre-trained CLIP [66] and StableDiffusion [68], respectively, and are kept frozen during training. To train the network, we acquire sets of front-view images, novel-view images, novel-view part segments, and text prompts. The front- and novel-view images are obtained by rendering 3D human scans from two randomly selected viewpoints. Then, novel-view part segments are extracted from the novel-view images using Sapiens [38]. The text prompts are automatically generated from the front-view images using the off-the-shelf text captioning model BLIP [44]. Using these data, we train the network by minimizing the L2 distance between the estimated noise and the target noise, following the conventional training strategy of diffusion networks. AdamW [52] optimizer is used for the training with a base learning rate of 5×10^{-5}. We train PartDiffusion for 36,000 steps with a batch size of 4 on a single NVIDIA A100 40GB GPU.

S6. More comparison results

We provide more qualitative results of our PARTE on THuman2.1 [91] and SHHQ [19]. Fig. S8 demonstrates that PARTE achieves significantly superior texture reconstruction compared to previous 3D human reconstruction methods, demonstrating better part alignment and visual fidelity. Figs. S9, S10, and S11 illustrate that our framework effectively handles in-the-wild scenarios. Tab. S4 shows that our PARTE also outperforms the existing reconstruction methods on 4D-DRESS [79], a dataset that has accurate 3D part labels. For evaluation on 4D-DRESS, we uniformly sample 16 GTs from its test set.

Figure S7. **Qualitative comparison of image generation with various diffusion networks and PartDiffusion, on THuman2.1 [91].**

S7. Limitations and future works

Unseen cloth types. Fig. S6 illustrates the failure cases of our framework when reconstructing unseen cloth types (*e.g.*, dresses) that are not included in the training dataset. Our training set is labeled based on the pre-defined part categories of Sapiens [38], which is limited to classifying clothes into two types, upper- and lower-clothes. However, this categorization does not include dresses or multi-layered outfits, resulting in segmentation failures for such cases. These segmentation failures lead to incorrect human texturing. We aim to extend our framework to handle various cloth styles by enriching the training data with more diverse cloth samples.

Fine-grained part segmentation. Our framework segments the human into $n = 5$ part categories, primarily focusing on broad regions. However, real-world human appearance includes detailed human body parts (*e.g.*, hair and eyes) with various accessories (*e.g.*, hat, glasses, and watch), which are not explicitly segmented in our framework. A potential future direction is to incorporate fine-grained part segmentation to improve texture reconstruction by accurately distinguishing these intricate elements.

Figure S8. **Qualitative comparison of PARTE with SiTH [27], HumanRef [94], and TeCH [34], on THuman2.1 [91] and SHHQ [19].** We highlight their representative failure cases with red circles.

Figure S9. **More qualitative results of PARTE on in-the-wild images of SHHQ [19].**

Figure S10. **More qualitative results of PARTE on in-the-wild images of SHHQ [19].**

Figure S11. **More qualitative results of PARTE on in-the-wild images of SHHQ [19].**