Modeling Detailed Human Geometry with Adaptive Local Refinement

Supplementary Material

1. Modified Sliced Wasserstein Distance

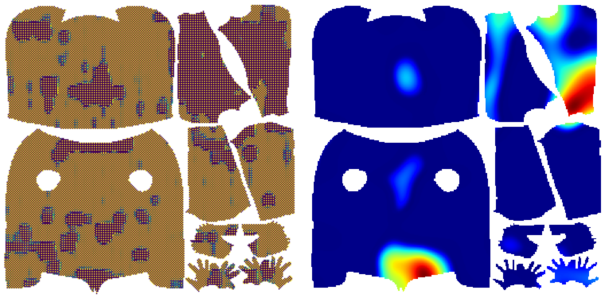
We adopt the sliced Wasserstein distance (SW) [1] to train the refinement module to avoid the local minimum issue from Chamfer distance. For better efficiency, we find a newer variant of SW that further improves the performance of SW on 3D point cloud learning [2, 4]. The computation details of the distance we used are shown in 1. In our application, we choose sample size $N = 150$ and start from 50% of the portfolio size to incrementally reach 100%.

Algorithm 1 Computation of the Distance

Input: Two point sets: X, Y with $|X| = |Y| = N$, positive integer K and L s.t. $K \leq L$ **Output:** sw_K Sample L directions Θ_L on $\mathbb{S}_{d-1}$ Reorder X, Y s.t. $\mathcal{R}I_{X_i}(\cdot, \theta_l) \leq \mathcal{R}I_{X_j}(\cdot, \theta_l)$ for $i \leq j$ for each slice θ_l

For i^{th} point in X, select K slices $\Theta_K^{(i)} \subseteq \Theta_L$ s.t. for $\theta_l \in \Theta_L$ and $\notin \Theta_K$, $S_{\theta_l}(X_k^{(i)}, Y_k^{(i)}) \leq S_{\theta_l}(X_l^{(i)}, Y_l^{(i)})$

Let $w_k^{(n)} \leftarrow W_p\left(\mathcal{R}I_{\mu_n}(X_n, \theta_k), \mathcal{R}I_{\nu_n}(Y_n, \theta_k)\right)$ SW $(\mu, \nu) = \left(\frac{1}{K}\sum_{k=1}^{L}\frac{1}{N}\sum_{n=1}^{N} w_k^{(n)}\right)^{\frac{1}{p}}$



(a) (b)

Figure 1. (a) With transpose convolution. (b) With resize-convolution

2. Implementation Details

Coarse Prediction Module Architecture: Following the naming convention of Pix2PixHD [5]: Let `c7s1-k` denotes a 7×7 Convolution-BatchNorm-ReLU layer with k filters and stride 1. `dk` denotes a 3×3 Convolution-BatchNorm-ReLU layer with k filters, and stride 2. We use reflection padding in the network. `Rk` represents a residual block that contains two 3×3 convolutional layers with k filters on both layers. `uk` denotes a 3×3 TransposeConvolution-BatchNorm-ReLU layer with k filters and stride 2. Our image-to-image translation module is:

`c7s1-64,d128,d256,d512,d1024,R1024, R1024,R1024,R1024,R1024,R1024,R1024, R1024,R1024,u512,u256,u128,u64,c7s1-6`

Attention-head Architecture: We denote `Uk` a resize-convolution layer that contains a bilinear upsampling operation with the factor of 2 and a 3×3 Convolution-BatchNorm-ReLU layer with k filters, and stride 1. Our attention-head G_{att} is:

`U512,U256,U128,U64,c7s1-1`

We train all methods on THUman2.0 dataset except PIFuHD and Tex2Shape, whose training codes are not available. We randomly split 80 % of the data to train and the rest to test. It is augmented by rendering images from 36 views, the same as ICON/ECON.

Checkerboard Artifacts in the Attention Head: In Section 3.3, we mention that we implement resize-convolution layers to reduce the checkerboard artifacts, which is caused by the transpose convolution layers in the traditional upsampling structures. Within the layer, we first upsample the feature map through bilinear interpolation and apply a convolution with padding that retains the map dimension. It successfully eliminates the uneven overlap of the reception field of the transpose convolution operation that spawns the checkerboard artifacts. The comparison of the attention maps is demonstrated in Figure 1.

Refinement Module Architecture: For the refinement module, we use a convolutional encoder and a MLP-based decoder. Denote `Ek` a 4×4 Convolution-BatchNorm-LeakyReLU layer with k filters and stride 2. We choose 0.2 for the negative slope. The refinement encoder is:

`E32,E64,E128,E256,E256,E256`

With N refinement points, the refinement decoder is:

`fc-512,fc-512,fc-512,fc-512,fc-512, fc-512,fc-512,fc-Nx6`

Each `fc` layer is followed by batch normalization and ReLU activation except the last layer.

For the human model reconstruction experiment, we render texture and positional maps with resolution 256×256 for the model input, which yields 50669 points. The input RGB image is first transformed into an IUV image through DensePose, which contains UV coordinates per part. Then we utilize a preset mapping to map the IUV to the partial UV on the right. Moreover, we choose $|X_r| = 10000$ for the refinement module, making the final point cloud with a size of 60669 points. Note that all UV positional maps are stacked with corresponding normal maps within the entire framework, as shown in Figure 2. The resultant mesh

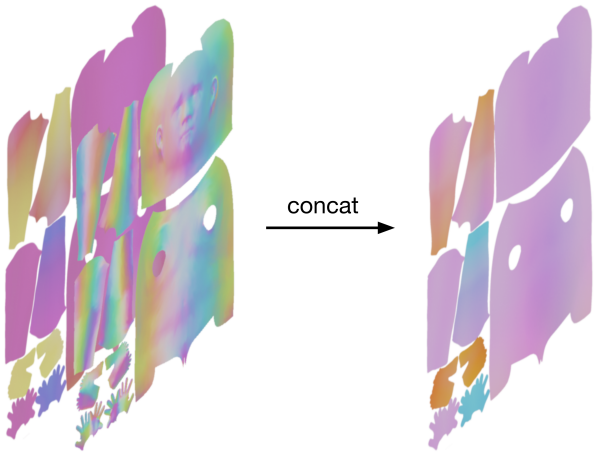


Figure 2. UV maps mentioned and drawn are the concatenation of a positional map and a normal map.

is computed via point set normal computation and Poisson Surface Reconstruction [3]. Each module is trained with the Adam optimizer with a learning rate of 10^{-4} for 300 epochs. The models are trained and inferred on a single NVIDIA RTX3090 GPU. ***We will open-source our project after publication***.

3. Additional Results

Presented in Figure 4 and Figure 5 are further comparative results visualized under the input image view. Our model consistently exhibits the highest degree of detail, matching or even surpassing the quality demonstrated by the best benchmark models, such as PIFuHD and ECON. Note that although promising in visible areas, we've demonstrated that PIFuHD and ECON tend to falter when confronted with occluded views, often yielding unrealistic body shapes. We illustrate the per-vertex Chamfer distance in Figure 3. It provides a visual comparison of our method with the aforementioned state-of-the-art methods. Our model creates a similar level of visual appearance compared with more complex methods, such as ICON and ECON, and due to the 2D UV learning, the overall shape is closer to the ground truth as shown in the per-vertex error visualization, which therefore, quantitatively illustrates other methods' lack of shape accuracy from non-input views. Note that due to the potential of loss of certain details during poisson reconstruction, the per-vertex error is computed using the raw point cloud input. We present videos of panoramic views of generated models in the package as well.

References

[1] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015. 1

[2] Bang Du, Kunyao Chen, Haochen Zhang, and Truong Nguyen. Select-sliced wasserstein distance for point cloud learning. In *International Conference on 3D Vision (3DV)*, 2024. 1

[3] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, page 0, 2006. 2

[4] Trung Nguyen, Quang-Hieu Pham, Tam Le, Tung Pham, Nhat Ho, and Binh-Son Hua. Point-set distances for learning representations of 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10478–10487, 2021. 1

[5] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 1

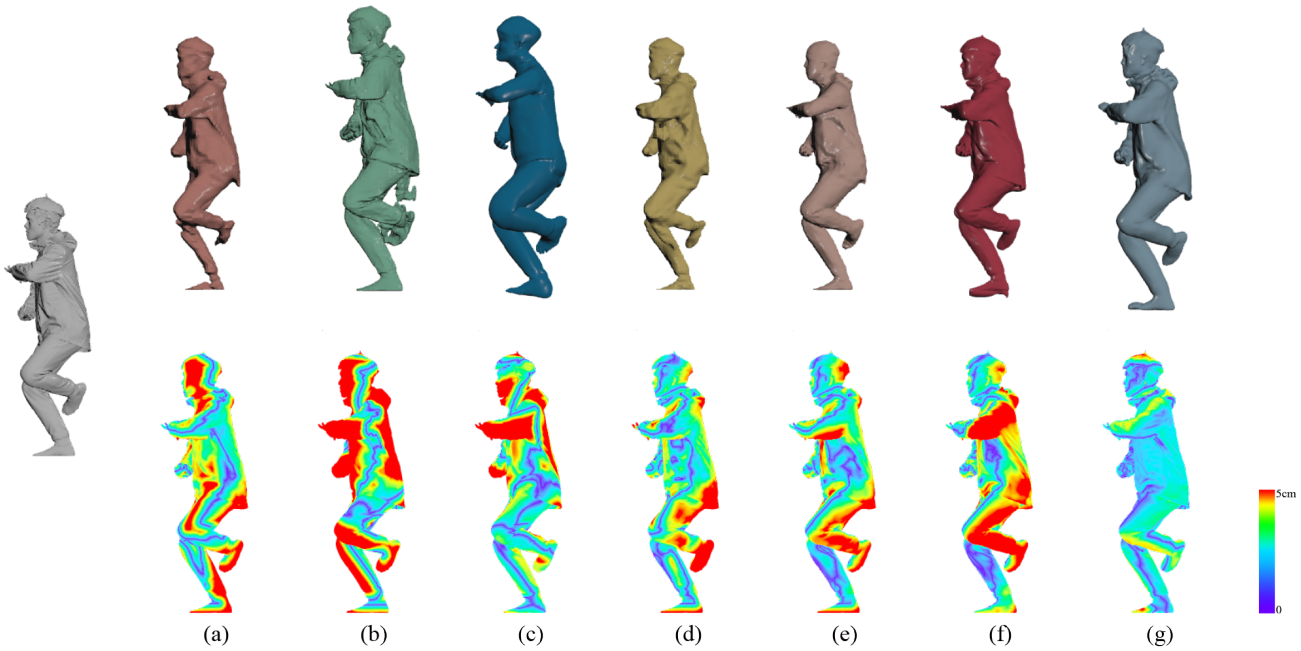


Figure 3. Ground-truth mesh (left) and result-error pairs from seven comparison methods: (a) PIFu, (b) PIFuHD, (c) Tex2Shape, (d) PaMIR, (e) ICON, (f) ECON, and (g) Ours.

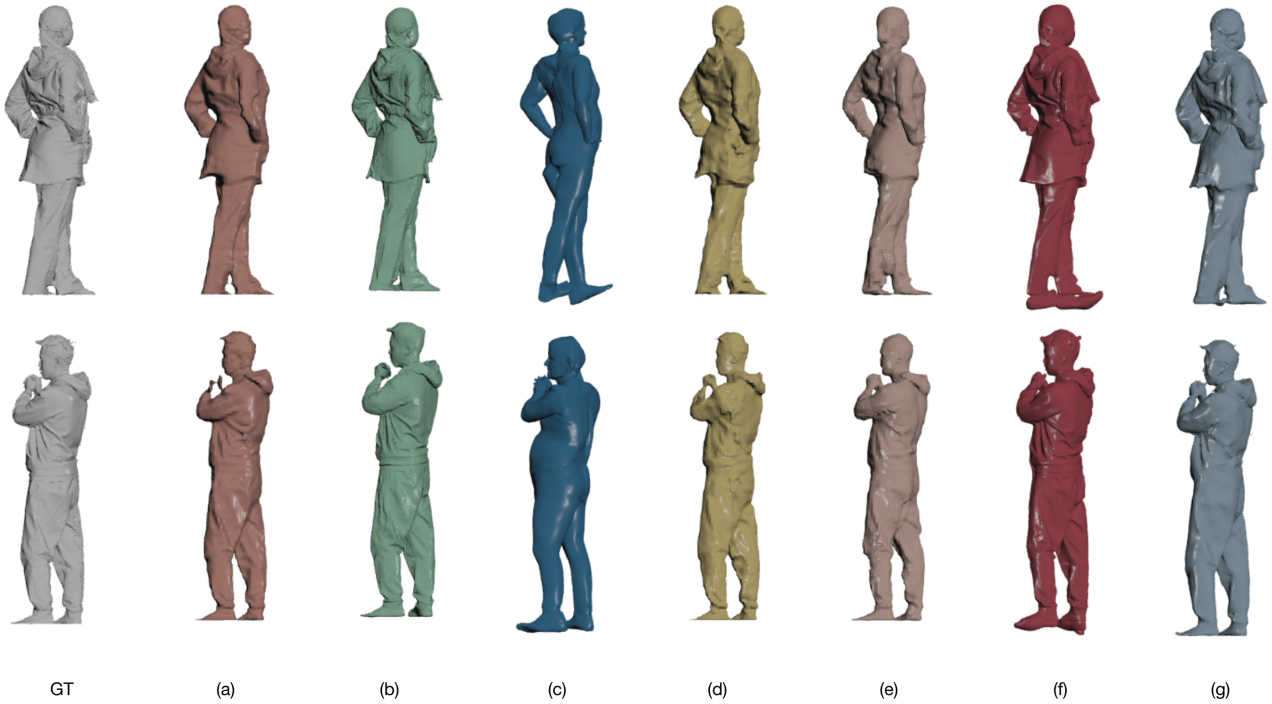


Figure 4. (a) PIFu, (b) PIFuHD, (c) Tex2Shape, (d) PaMIR, (e) ICON, (f) ECON, and (g) Ours. Our model presents the finest class of details on visible regions.

Figure 5. (a) PIFu, (b) PIFuHD, (c) Tex2Shape, (d) PaMIR, (e) ICON, (f) ECON, and (g) Ours.