PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization – Supplementary Materials

Shunsuke Saito1,2,* Zeng Huang1,2,* Ryota Natsume3∗ Shigeo Morishima3 Angjoo Kanazawa4 Hao Li1,2,5

1University of Southern California  2USC Institute for Creative Technologies  3Waseda University  4University of California, Berkeley  5Pinscreen

Appendix I. Implementation Details

Experimental Setup. Since there is no large scale datasets for high-resolution clothed humans, we collected photogrammetry data of 491 high-quality textured human meshes with a wide range of clothing, shapes, and poses, each consisting of about 100,000 triangles from RenderPeople1. We refer to this database as High-Fidelity Clothed Human Data set. We randomly split the dataset into a training set of 442 subjects and a test set of 49 subjects. To efficiently render the digital humans, Lambertian diffuse shading with surface normal and spherical harmonics are typically used due to its simplicity and efficiency [16, 11]. However, we found that to achieve high-fidelity reconstructions on real images, the synthetic renderings need to correctly simulate light transport effects resulting from both global and local geometric properties such as ambient occlusion. To this end, we use a precomputed radiance transfer technique (PRT) that precomputes visibility on the surface using second-order spherical harmonics of indoor scene from HDRI Haven2 using random rotations around y axis. We render the images by aligning subjects to the image center using a weak-perspective camera model and image resolution of 512 × 512. We also rotate the subjects for 360 degrees in yaw axis, resulting in 360 × 442 = 159,120 images for training. For the evaluation, we render 49 subjects from RenderPeople and 5 subjects from the BUFF data set [21] using 4 views spanning every 90 degrees in yaw axis. Note that we render the images without the background. We also test our approach on real images from the DeepFashion data set [10]. In the case of real data, we use a off-the-shelf segmentation network together with Grab-Cut refinement [13].

Network Architecture. Since the framework of PIFu is not limited to a specific network architecture, one can technically use any fully convolutional neural network as the image encoder. For surface reconstruction, we adopt the stacked hourglass network [12] with modifications proposed by [8]. We also replace batch normalization with group normalization [20], which improves the training stability when the batch sizes are small. Similar to [8], the intermediate features of each stack are fed into PIFu, and the losses from all the stacks are aggregated for parameter update. We have conducted ablation study on the network architecture design and compare against other alternatives (VGG16, ResNet34) in Appendix II. The image encoder for texture inference adopts the architecture of CycleGAN [22] consisting of 6 residual blocks [9]. Instead of using transpose convolutions to upsample the latent features, we directly feed the output of the residual blocks to the following Tex-PIFu.

PIFu for surface reconstruction is based on a multi-layer perceptron, where the number of neurons is (257, 1024, 512, 256, 128, 1) with non-linear activations using leaky ReLU except the last layer that uses sigmoid activation. To effectively propagate the depth information, each layer of MLP has skip connections from the image feature \(F(x) \in \mathbb{R}^{256}\) and depth \(z\) in spirit of [4]. For multi-view PIFu, we simply take the 4-th layer output as feature embedding and apply average pooling to aggregate the embedding from different views. Tex-PIFu takes \(F_C(x) \in \mathbb{R}^{256}\) together with the image feature for surface reconstruction \(F_V(x) \in \mathbb{R}^{256}\) by setting the number of the first neurons in the MLP to 513 instead of 257. We also replace the last layer of PIFu with 3 neurons, followed by tanh activation to represent RGB values.

Training procedure. Since the texture inference module requires pretrained image features from the surface reconstruction module, we first train PIFu for the surface reconstruction and then for texture inference, using the learned image features \(F_V\) as condition. We use RMSProp for the surface reconstruction following [12] and Adam for the texture inference with learning rate of \(1 \times 10^{-3}\)
as in [22], the batch size of 3 and 5, the number of epochs of 12 and 6, and the number of sampled points of 5000 and 10000 per object in every training batch respectively. The learning rate of RMSProp is decayed by the factor of 0.1 at 10-th epoch following [12]. The multi-view PIFu is fine-tuned from the models trained for single-view surface reconstruction and texture inference with a learning rate of $10^{-4}$ and 2 epochs. The training of PIFu for single-view surface reconstruction and texture inference takes 4 and 2 days, respectively, and fine-tuning for multi-view PIFu can be achieved within 1 day on a single 1080ti GPU.

Appendix II. Additional Evaluations

Spatial Sampling. In Table 2 and Figure 1, we provide the effects of sampling methods for surface reconstruction. The most straightforward way is to uniformly sample inside the bounding box of the target object. Although it helps to remove artifacts caused by overfitting, the decision boundary becomes less sharp, losing all the local details (See Figure 1, first column). To obtain a sharper decision boundary, we propose to sample points around the surface with distances following a standard deviation $\sigma$ from the actual surface mesh. We use $\sigma = 3$, 5, and 15 cm. The smaller $\sigma$ becomes, the sharper the decision boundary is the result becomes more prone to artifacts outside the decision boundary (second column). We found that combining adaptive sampling with $\sigma = 5$ cm and uniform sampling achieves qualitatively and quantitatively the best results (right-most column). Note that each sampling scheme is trained with the identical setup as our training procedure described in Appendix I.

Network Architecture. In this section, we show comparisons of different architectures for the surface reconstruction and provide insight on design choices of the image encoders. One option is to use bottleneck features of fully convolutional networks [9, 19, 12]. Due to its state-of-the-art performance in volumetric regression for human faces and bodies, we choose Stacked Hourglass network [12] with a modification proposed by [8], denoted as HG. Another option is to aggregate features from multiple layers to obtain multi-scale feature embedding [2, 7]. Here we use two widely used network architectures: VGG16 [14] and ResNet34 [6] for the comparison. We extract the features from the layers of ‘relu1_2’, ‘relu2_2’, ‘relu3_3’, ‘relu4_3’, and ‘relu5_3’ for VGG network using bilinear sampling based on $x$, resulting in 1472 dimensional features. Similarly, we extract the features before every pooling layers in ResNet, resulting in 1024-D features. We modify the first channel size in PIFu to incorporate the feature dimensions and train the surface reconstruction model using the Adam optimizer with a learning rate of $10^{-3}$, the number of sampling of 10,000 and batch size of 8 and 4 for VGG and ResNet respectively. Note that VGG and ResNet are initialized with models pretrained with ImageNet [5]. The other hyperparameters are the same as the ones used for our sequential network based on Stacked Hourglass.

In Table 1 and Figure 2, we show comparisons of three architectures using our evaluation data. While ResNet has slightly better performance in the same domain as the training data (i.e., test set in RenderPeople dataset), we observe that the network suffers from overfitting, failing to generalize to other domains (i.e., BUFF and DeepFashion.
Table 1: Ablation study on the sampling strategy.

<table>
<thead>
<tr>
<th>Methods</th>
<th>RenderPeople Normal P2S Chamfer</th>
<th>Buff Normal P2S Chamfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform</td>
<td>0.119 5.07 4.23</td>
<td>0.132 5.98 4.53</td>
</tr>
<tr>
<td>$\sigma = 3 \text{cm}$</td>
<td>0.104 2.03 1.62</td>
<td>0.114 6.15 3.81</td>
</tr>
<tr>
<td>$\sigma = 5 \text{cm}$</td>
<td>0.105 1.73 1.55</td>
<td>0.115 1.54 1.41</td>
</tr>
<tr>
<td>$\sigma = 15 \text{cm}$</td>
<td>0.100 1.49 1.43</td>
<td>0.105 1.37 1.26</td>
</tr>
<tr>
<td>$\sigma = 5 \text{cm} + \text{Uniform}$</td>
<td>0.084 1.52 1.50</td>
<td>0.092 1.15 1.14</td>
</tr>
</tbody>
</table>

Table 2: Ablation study on network architectures.

<table>
<thead>
<tr>
<th>Methods</th>
<th>RenderPeople Normal P2S Chamfer</th>
<th>Buff Normal P2S Chamfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16</td>
<td>0.125 3.02 2.25</td>
<td>0.144 4.65 3.08</td>
</tr>
<tr>
<td>ResNet34</td>
<td>0.097 1.49 1.43</td>
<td>0.099 1.68 1.50</td>
</tr>
<tr>
<td>HG</td>
<td>0.084 1.52 1.50</td>
<td>0.092 1.15 1.14</td>
</tr>
</tbody>
</table>

Please see the supplementary video for more results.

**Comparison with Template-based Method.** In Figure 3 and Table 3, we compare our approach with a template based method [1] that takes a dense 360 degrees view video as an input on BUFF dataset. From 3 views we outperform the template based method. Note that Alldieck et al. requires an uncalibrated dense video sequence, while ours requires calibrated sparse view inputs.

**Comparison with Voxel Regression Network.** We provide an additional comparison with Voxel Regression Network (VRN) [8] to clarify the advantages of PIFu. Figure 4 demonstrates that the proposed PIFu representation can align the 3D reconstruction with pixels at higher resolution, while VRN suffers from misalignment due to the limited precision of its voxel representation. Additionally, the generality of PIFu offers texturing of shapes with arbitrary topology and self-occlusion, which has not been addressed by the work of VRN. Note that VRN only is able to project the image texture onto the recovered surface, and does not provide an approach to do texture inpainting on the unseen side.
Results on General Objects. In this work, we focused largely on clothed human surfaces. A natural question is how it extends to general object shapes. Our preliminary experiments on the ShapeNet dataset [3] in a class agnostic setting reveals new challenges as shown in Figure 5. We speculate that the greater variety of object shapes makes it difficult to learn a globally coherent shape from only pixel-level features. Note that recently [18] extend the idea of PIFu by explicitly combining global features and local features, demonstrating globally coherent and locally detailed reconstruction for general objects is possible.

Results on Video Sequences. We also apply our approach to video sequences obtained from [17]. For the reconstruction, video frames are center cropped and scaled so that the size of the subjects are roughly aligned with our training data. Note that the cropping and scale is fixed for each sequence. Figure 6 demonstrates that our reconstructed results are reasonably temporally coherent even though the frames are processed independently.

References

Figure 6: Results on video sequences obtained from [17]. While ours uses a single view input, the ground truth is obtained from 8 views with controlled lighting conditions.