Supplementary Material
Diffusion-FOF: Single-view Clothed Human Reconstruction via Diffusion-based Fourier Occupancy Field

Yuanzhen Li, Fei Luo*, Chunxia Xiao*
School of Computer Science, Wuhan University, China
yuanzhen@whu.edu.cn, luofei@whu.edu.cn, cxxiao@whu.edu.cn

1. Implementation Details

We implemented our method using a single NVIDIA GeForce RTX 3090Ti GPU with the PyTorch framework [4]. We use Adam optimizer [3] to optimize the back-view image and geometric prediction networks. In the back-view image prediction, we set the learning rate to 10^{-4}, the batch size to 6, and 14 epochs. In the geometric reconstruction, we set the learning rate to 4×10^{-5}, the batch size to 8, and 20 epochs.

2. Training and Testing Details

We provide the pseudo-code of the geometric training and test procedures in Algorithm 1 and Algorithm 2, respectively. Algorithm 3 serves as a supplementary component to both Algorithms 1 and Algorithm 2. Figure 1 illustrates the generation process of the geometric model during inference.

3. More Geometric Reconstruction Results

Figure 2 and Figure 3 provide more geometric results to demonstrate the superiority of our method further. In Figure 2, the second input image is from Renderpeople [5], and the other two input images are from 2K2K [2]. Compared with these methods, our method can effectively reconstruct loose-fitting clothes and generate more realistic details in invisible areas. In Figure 3, the first input image is from 2K2K [2], and the other images are from the Internet. Compared with these methods, our method can effectively reconstruct the geometry of children and generate more realistic details in invisible areas.

References

[1] Qiaojun Feng, Yebin Liu, Yu-Kun Lai, Jingyu Yang, and Kun Li. Fof: Learning fourier occupancy field for monocular real-time human reconstruction. In *NeurIPS*, 2022. 3, 4

[2] Sang-Hun Han, Min-Gyu Park, Ju Hong Yoon, Ju-Mi Kang, Young-Jae Park, and Hae-Gon Jeon. High-fidelity 3d human digitization from single 2k resolution images. In *CVPR*, 2023. 1

[3] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[4] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS*, 2017. 1

[5] Renderpeople. https://renderpeople.com/, 2014. 1

[6] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, pages 84–93, 2020. 3, 4

[7] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J. Black. ECON: Explicit Clothed humans Optimized via Normal integration. In *CVPR*, 2023. 3, 4

[8] Xueting Yang, Yihao Luo, Yuliang Xiu, Wei Wang, Hao Xu, and Zhaoxin Fan. D-IF: Uncertainty-aware Human Digitization via Implicit Distribution Field. In *ICCV*, 2023. 3, 4

[9] Zechuan Zhang, Li Sun, Zongxin Yang, Ling Chen, and Yi Yang. Global-correlated 3d-decoupling transformer for clothed avatar reconstruction. In *NeurIPS*, 2023. 3, 4

*Chunxia Xiao and Fei Luo are co-corresponding authors

Algorithm 1: Training

```
def train(I_a, I'_b, fof_gt) :
    # Wavelet transform
    wavelet_gt = DWT (fof_gt)
    # Encoder image features
    feat_high, feat_low = image_encoder (I_a, I'_b)
    # Add noise to wavelet_gt
    t, eps = uniform (0, 1), normal (mean=0, std=1)
    wavelet_noise = sqrt (alpha_cumprod(t)) × wavelet_gt + sqrt (1-alpha_cumprod(t)) × eps
    # Predict wavelet
    wavelet_pr = Denoise-Net (wavelet_noise, feat_low, t)
    # inverse wavelet transform
    fof_pr = IWT (wavelet_pr)
    # fof refinement
    fof_refine = refine-Net (fof_pr, feat_high)
    # Set loss
    loss = loss_function (wavelet_pr, wavelet_gt, fof_pr, fof_refine, fof_gt)
    return loss
```

Algorithm 2: Testing

```
def test(I_a, I'_b, steps, td=1) :
    # steps: sample steps; td: time difference
    wavelet_t = normal (mean=0, std=1)
    # Encoder image features
    feat_high, feat_low = image_encoder (I_a, I'_b)
    for step in range (steps) :
        # Time intervals
        t_now = 1 - step / steps
        t_next = max (1 - (step + 1 + td) / steps, 0)
        # Predict wavelet from wavelet_t
        wavelet_pr = Denoise-Net (wavelet_t, feat_low, t_now)
        # Update wavelet_t at t_next
        wavelet_t = update (wavelet_t, wavelet_pr, t_now, t_next)
    fof_pr = IWT (wavelet_pr)
    fof_refine = refine-Net (fof_pr, feat_high)
    return fof_refine
```

Algorithm 3: Update

```
def alpha_cumprod (t, ns=0.0002, ds=0.00025) :
    # cosine noise schedule
    n = torch.cos((t + ns)/(1 + ds) × math.pi/2)^(-2)
    return -torch.log(n-1, eps=1e-5)

def update (wavelet_t, wavelet_pr, t_now, t_next) :
    α_now = alpha_cumprod (t_now)
    α_next = alpha_cumprod (t_next)
    eps = 1/sqrt(1-α_now) × (wavelet_t - sqrt(α_now) × wavelet_pr)
    wavelet_next = sqrt(α_next) × wavelet_pr + sqrt(1 - α_next) × eps
    return wavelet_next
```

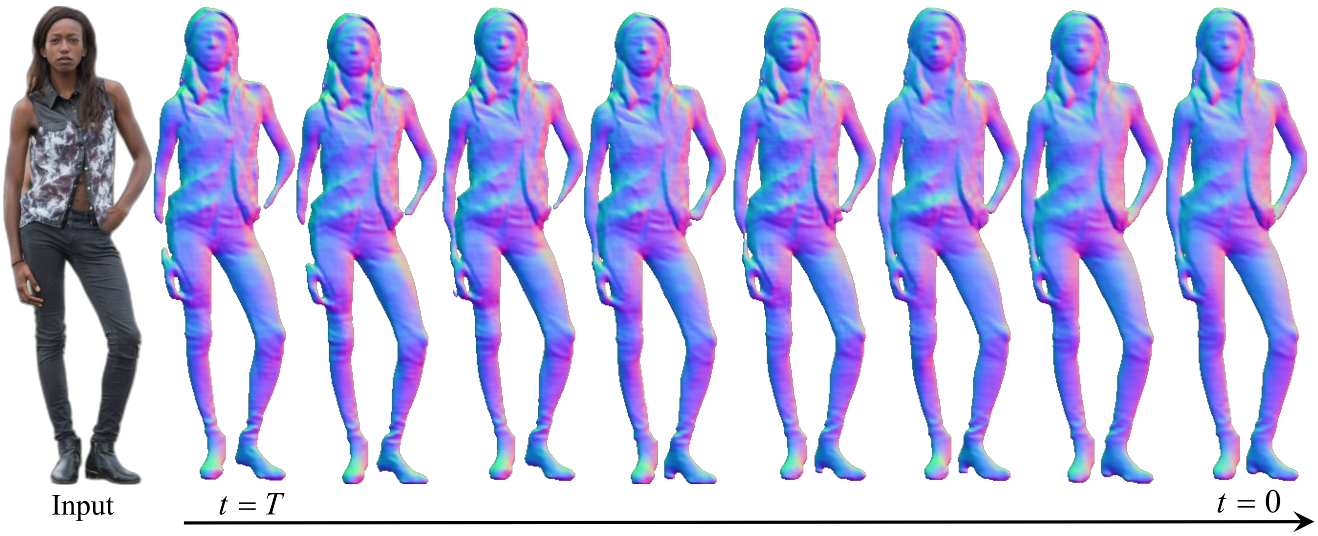

Figure 1. The geometry generation process during inference.

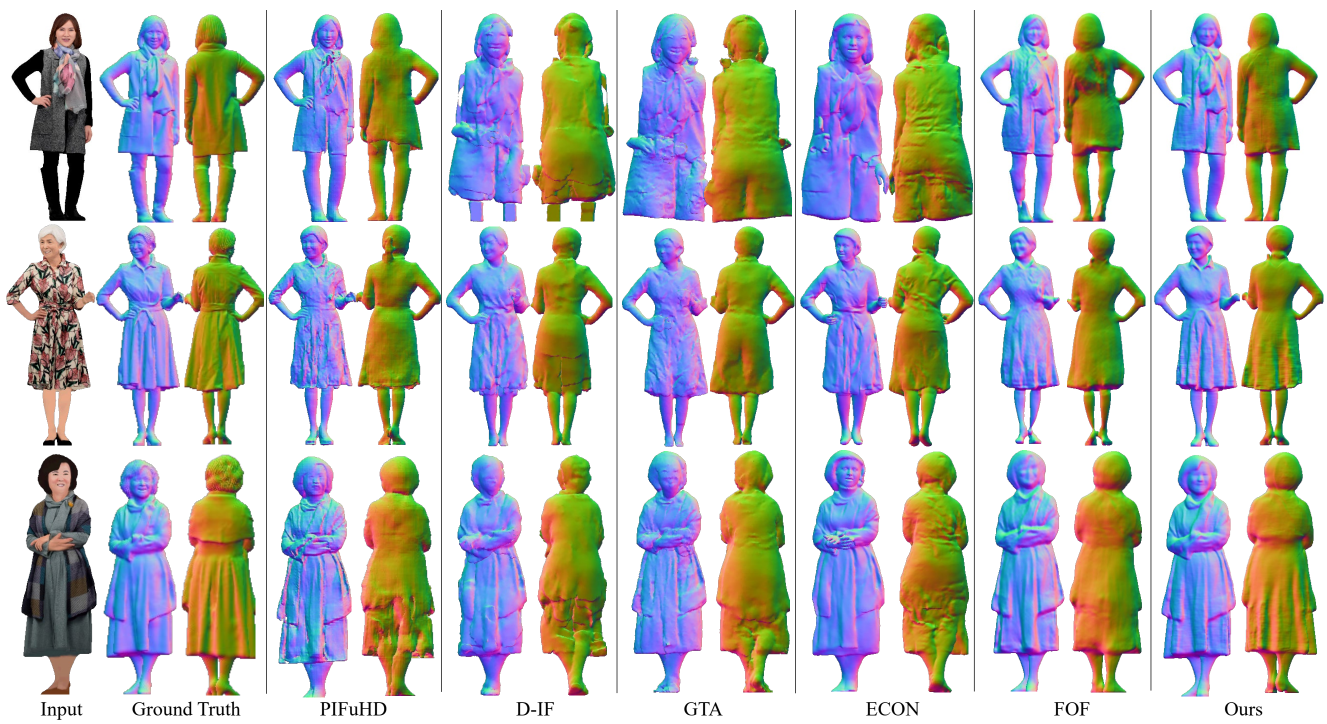

Figure 2. Qualitative comparison with state-of-the-art single-view clothed human reconstruction methods: PIFuHD [6], D-IF [8], GTA [9], ECON [7], and FOF [1].

Figure 3. Qualitative comparison with state-of-the-art single-view clothed human reconstruction methods: PIFuHD [6], D-IF [8], GTA [9], ECON [7], and FOF [1].