Supplementary Material

1. Implementation details

1.1. Forward Mapping

The warping process of point x_o can divided into two steps:

$$\vec{\mathcal{W}} : (x_o, \theta) \rightarrow \vec{\mathcal{W}}_s \rightarrow (x_s, \theta) \rightarrow \vec{\mathcal{W}}_{nr} \rightarrow (x_c) \quad (1)$$

where x_c is a 3D position in canonical volume, θ is the current human pose in SMPL format. The skeleton-driven deformation $\vec{\mathcal{W}}_s$, which represents the coarse deformation produced by joint rotation. It wrap point x_o to x_s(in canonical space). $\vec{\mathcal{W}}_{nr}$ starts from x_s and produces an offset $\triangle x$ to it, $\vec{\mathcal{W}}_{nr}$ provides the non-rigid effects caused by clothing.

Skeleton Motion The skeletal deformation $\vec{\mathcal{W}}_s$ is a kind of linear blend skinning that is caused by skeletal motion and can be represented as:

$$\vec{\mathcal{W}}_s(x_o, \theta) = \frac{\sum_{i=1}^{24} \vec{w}_i(x_o)(\vec{R}_i x_o + \vec{t}_i)}{\sum_{i=1}^{24} \vec{w}_i(x_o)} \quad (2)$$

$\vec{w}_i(x_0)$ is the blending weight of x_o corresponding to *i*-th bone. We choose an explicit volume $\hat{\mathcal{V}}$ under the canonical space to store these value, $w_i = \vec{\mathcal{V}}_i(\vec{R}_i x_o + \vec{t}_i)$, $\vec{R}_i, \vec{t}_i$ can be explicitly computed from body pose θ. We refer the reader to [6] for more details.

Non-rigid Motion. The $\vec{\mathcal{W}}_{nr}$ is considered as a offset $\triangle x$ to the skeleton-driven result x_s. To be specific, point x_o is warped by $\vec{\mathcal{W}}_s$ to the skeleton-driven position x_s. Then, the non-rigid motion MLP estimates the offset to the x_s and gets the final position $x_c = x_s + \triangle x$ in canonical space.

$$\vec{\mathcal{W}}_{nr} : (x_s, \theta) \rightarrow \triangle x \quad (3)$$

1.2. Canonical decomposition network

In Fig. 1 and Fig. 2 we show the detailed architecture of the canonical decomposition network.

We use an 8-layer MLP with width = 256 following NeRF [5]. The network takes positional encoding $\gamma(x_o)$ as input and output normal n, color c, density θ and BRDF b. We apply a skip connection that concatenates $\gamma(x_o)$ to the fifth layer. We adopt ReLU activation after each fully connected layer. The BRDF decoder takes the latent feature h and outputs BRDF b.

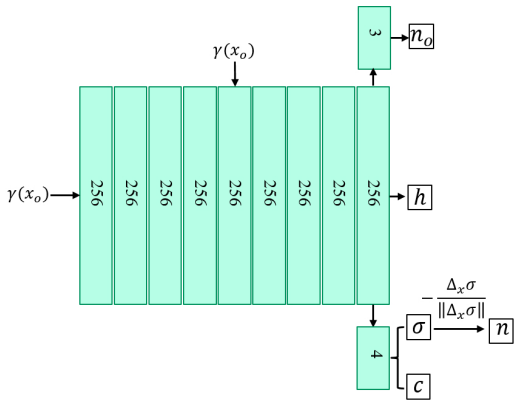

Figure 1. The network architecture of canonical decomposition network. Our network takes positional encoding $\gamma(x_o)$ in the canonical space as input and output normal n_c, color c, density θ, and latent feature h. The inverse gradient is calculated as weak supervision. The latent feature h is used for estimating BRDF parameters.

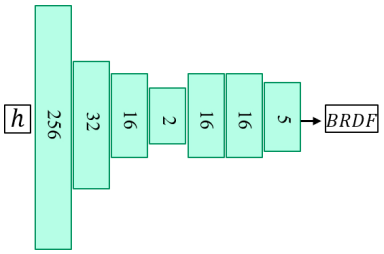

Figure 2. The network architecture of BRDF decoder network. The network takes the latent feature in the canonical space and outputs BRDF parameters.

1.3. Details about training

Losses: For the photometric loss and the re-render loss, MSE and LPIPS are employed. We choose VGG as the backbone of LPIPS. We apply adaptive weights $\overrightarrow{\lambda}$ for the photometric loss and $\overleftarrow{\lambda}$ for the re-render loss. We set $\overrightarrow{\lambda} = 0.9^{\frac{iter}{5000}}$, $\overleftarrow{\lambda} = 1 - \overrightarrow{\lambda}$, $\lambda_m = 10$ and $\lambda_s = 0.0005$ for ZJU-Mocap and $\lambda_s = 0.0001$ for synthetic dataset.

Progressive training: Our network uses inverse gradient as

Figure 3. Decomposition and relighting result on the dynamic human. We show the addition decomposition and relighting result of 3 subjects (377, 386, 393) on the ZJU-Mocap dataset.

weak supervision. The normal loss can be described as:

$$L_n = \lambda_n \left\| n_o - (-\frac{\nabla_x \sigma}{\|\nabla_{\boldsymbol{x}} \sigma\|}) \right\| \tag{4}$$

Where n_o is the final normal predicted by our network. We apply an adaptive weight λ_n to the normal loss and set different values for different datasets. For the ZJU-Mocap dataset, we set $\lambda_n = 0.05, (iteration < 20000)$ and $\lambda_n = 0.2, (iteration > 20000)$. For the synthetic dataset, due to the error of the initial estimated SMPL parameters, we set λ_n to be 0.01 and 0.05 in the two stages, respectively.

Our work employs the inverse gradient prior to initializing the network and adopt an MLP network $\mathcal{F}_{n_c}$ to predict the normal $\boldsymbol{n_c}$ in canonical space: (Eq. (5)):

$$\boldsymbol{n_c} = \begin{cases} -\frac{\nabla_x \sigma}{\|\nabla_{\boldsymbol{x}} \sigma\|} & \text{step} <= \text{t}; \\ \mathcal{F}_{n_c}(x, \theta) & \text{step} > \text{t}; \end{cases} \tag{5}$$

We set different values for t when training different datasets. In general, the larger the t is, the more accurate the normal and metalness parameters, but it will lead to overfitting, and the network will perform poorly in some light conditions. We set $t = 20000$ on the ZJU-Mocap dataset ($t = 20000$ for subject 392) and $t = 50000$ on the Synthetic dataset.

Training: We use 64 samples per ray and regularize the fourth layer of the BRDF decoder network with a $\mathfrak{L}^2$ norm with a scale of 0.01. The network is trained on 4 NVIDIA Tesla V100 GPUS for two days (400K iterations).

2. Results on ZJU-Mocap Dataset

For ZJU-MoCap, we trained models for six subjects (313, 377, 386, 387, 392, 393). We use images captured by camera one as input and directly apply camera matrices, body pose, and segmentation provided by the dataset.

Fig. 3 shows the additional decomposition and relighting results on the dynamic human. Our method can decompose a dynamic human into normal and plausible BRDF parameters and re-render the humans in different illuminations. Fig. 5 shows the additional novel view normal map and re-rendered results. Our method can render the decomposed parameters and re-light the human in free viewpoints. In the supplementary video, we also provide a detailed video comparison with Relighting4D and Relighting4DS and animated results. The results show that our method can provide a more accurate normal and re-rendered result and outperforms Relighting4D and Relighting4DS. Fig. 4 shows the visual comparison with HumanNeRF, our method is capable of relighting under unseen illuminations while HumanNeRF can't.

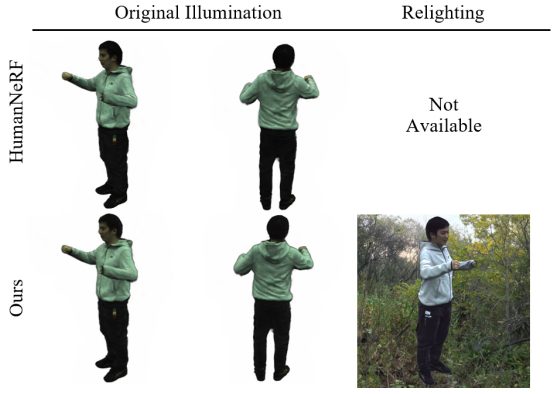

Figure 4. Visual comparison with HumanNeRF. Our method can relighting the dynamic human under unseen illuminations.

3. Details about Synthetic Dataset

To further validate our method. We build a challenging synthetic dataset with complex actions. The synthetic

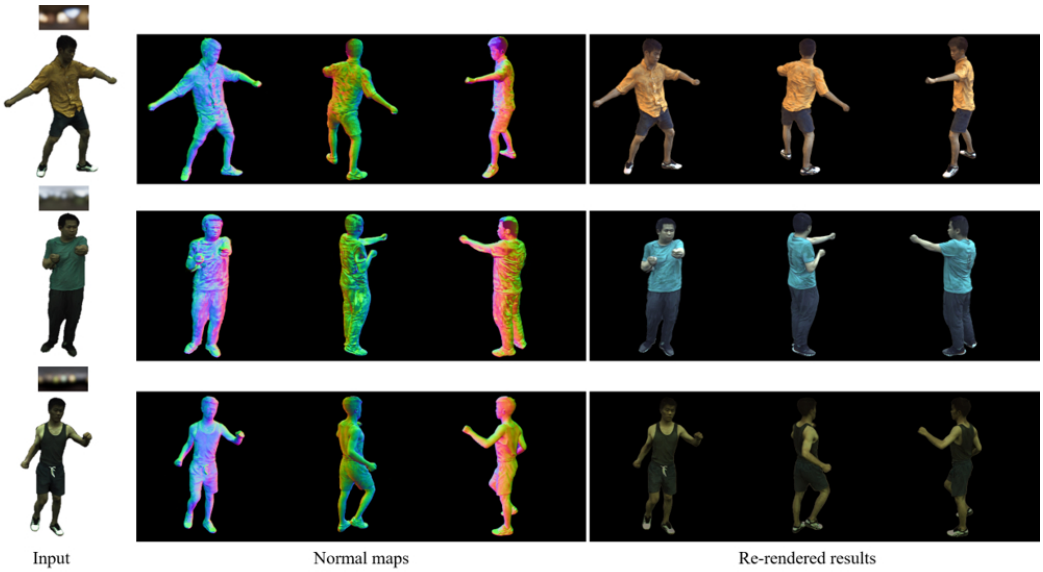

Figure 5. Novel view results on ZJU-Mocap dataset. We show normal maps and re-rendered novel view results on 3 subjects (377, 386, 393) on the ZJU-Mocap dataset.

dataset uses publicly available 3D characters and is driven using dance movements [2]. The dataset is rendered using Blender engine under [3] eight different light conditions.

3.1. Implementation Details

The synthetic dataset is re-rendered using publicly available HDRI maps [1]: courtyard, moonless golf, nature reserve forest, photo studio, portland landing pad, spruit sunrise, studio small, venice sunset. Each monocular video consists 510 frames. We choose the one rendered under photo studio for training, and others for evaluation (Fig. 6 and Fig. 7). We use VIBE [4] to estimate the body pose/camera parameters.

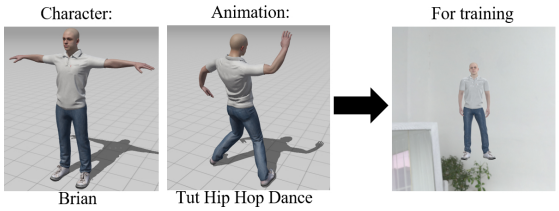

Figure 6. We chose Brian as the character and animated using Tut hip hop dance. The whole network is trained using video rendered under photo studio.

3.2. Re-rendered results

In this section, we show the re-rendered results over 8 environments (Fig. 8) light conditions. Our method gives a result that is close to the true value in all ambient light. Notice that there are some highlights as well as hands that do not work so well. The reasons for these errors are the overly complex dance movement poses and the poor estimation of the initial SMPL model. But the rendering is still very reasonable in terms of the overall effect.

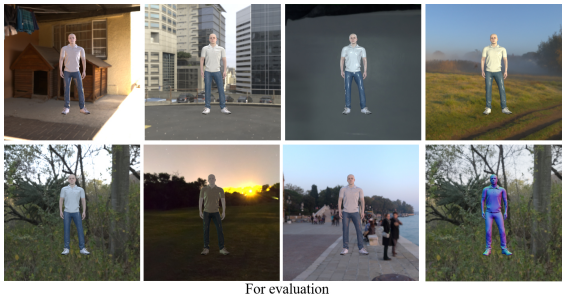

Figure 7. Visualization of evaluation datasets and normal map.

Figure 8. More qualitative results on the synthetic dataset. Although the movement of characters throughout the video is complex. Our method still gives plausible re-lighting effects. The reasons for some of the poor details are the complex motion, and the initialized SMPL model needs to be more accurate.

3.3. Normal Prior

Here we show the visual re-rendered result and normal of different normal estimation methods (re-rendered result under pose 'courtyard') (Fig. 9). Our method combines the advantages of inverse gradient and MLP estimating, which can give out the most reasonable result.

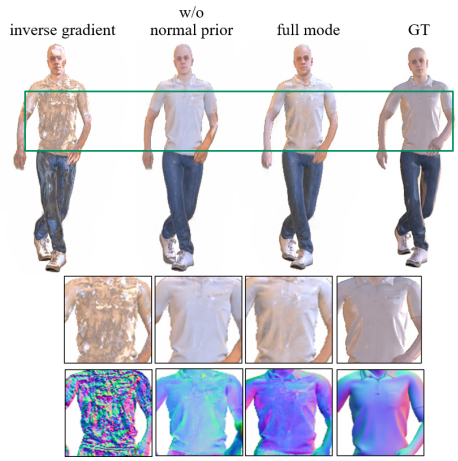

Figure 9. Visualization of different normal estimating approaches. Our method, which uses an MLP to estimate normal and inverse gradient as weak supervision gives the most reasonable result.

3.4. Inverse mapping

Fig. 10 shows the visual result of models trained with or without Inverse mapping network. The result is re-rendered under 'venice sunset.'

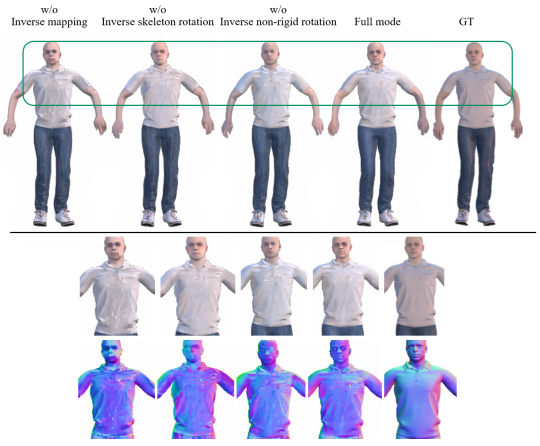

Figure 10. Visualization of different models trained with or without inverse mapping network. The inverse mapping network can build point-to-point connections, which work well when the motion is complex.

3.5. Failure case

As we mentioned before, due to the complexity of the motion, our method may have errors in some scenes with complex hand movements. As shown in the figure, since the 3D human pose and shape estimation algorithm cannot give an accurate prediction of the distal joints, the re-rendering results may have an impact during complex motions. In addition, some of the unreasonable highlights given by the Blender engine may also affect the results (Fig. 11). In addition, the Blender engine could not model non-rigid motion, which causes the normal processed by our method on the synthetic dataset is not as good as the real dataset.

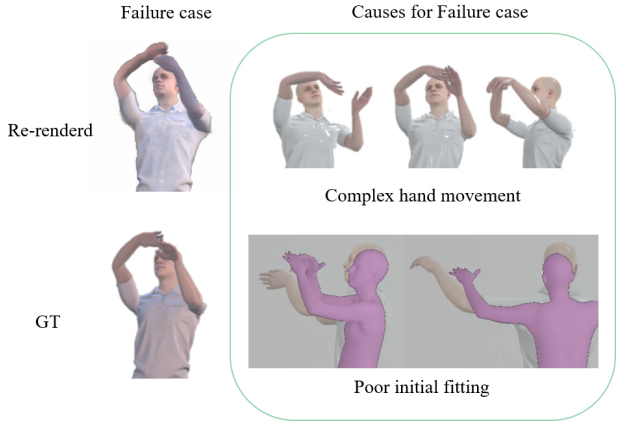

Figure 11. Visualization of failure case and causes.

References

[1] Poly haven. In *https://www.polyhaven.com*, 2020. 3

[2] Adobe. Mixamo. In *https://www.mixamo.com*, 2020. 3

[3] Community. B.o.: Blender - a 3d modelling and rendering package. In *Blender Fundation*, 2018. 3

[4] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020. 3

[5] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[6] Chung–Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16189–16199, 2022. 1