IntrinsicAvatar: Physically Based Inverse Rendering of Dynamic Humans from Monocular Videos via Explicit Ray Tracing

Shaofei Wang[1,2,3], Božidar Antić[2,3], Andreas Geiger[2,3], Siyu Tang[1]
[1]ETH Zürich [2]University of Tübingen [3]Tübingen AI Center

1. Volume Scattering Derivation

In this section, we derive the volume scattering approximation equation (Eq. (10) in the main paper) from the equation of transfer [20]. The general equation of transfer accounting for both volume emission and volume scattering is as follows:

$$
\begin{aligned}
C_{pbr}(\mathbf{r}) &= \int_{t_n}^{t_f} T(t_n, t)\sigma_t(\mathbf{r}(t))L(\mathbf{r}(t), -\mathbf{d})dt \\
\text{s.t} \quad & \mathbf{r}(t) = \mathbf{o} + t\mathbf{d} \\
& T(t_n, t) = \exp\left(-\int_{t_n}^{t} \sigma_t(\mathbf{r}(s))ds\right) \\
& \sigma_t(\mathbf{r}(t)) = \sigma_a(\mathbf{r}(t)) + \sigma_s(\mathbf{r}(t))
\end{aligned} \tag{1}
$$

As defined in the main paper, σ_s and σ_a are the *scattering* coefficient and the *absorption* coefficient, respectively. They define the probability of light being scattered/absorbed by the participating media per unit length. σ_t is the *attenuation* coefficient, which is the sum of σ_s and σ_a. Physically, it describes the probability of light being either out-scattered or absorbed per unit length, both of which will reduce the amount of radiance that reaches the camera. We refer readers to [21] for detailed explaination on physical meanings of these parameters. With some abuse of notation, we define L as the radiance accounting for both volume emission and volume scattering:

$$
\begin{aligned}
L(\mathbf{r}(t), -\mathbf{d}) &= \frac{\sigma_a(\mathbf{r}(t))}{\sigma_t(\mathbf{r}(t))}L_e(\mathbf{r}(t), -\mathbf{d}) + \frac{\sigma_s(\mathbf{r}(t))}{\sigma_t(\mathbf{r}(t))}L_s(\mathbf{r}(t), -\mathbf{d}) \\
\text{s.t.} \quad & L_s(\mathbf{x}, -\mathbf{d}) = \int_{S^2} f_p(\mathbf{x}, -\mathbf{d}, \bar{\mathbf{d}})L_i(\mathbf{x}, -\bar{\mathbf{d}})d\bar{\mathbf{d}}
\end{aligned} \tag{2}
$$

where L_e is the volume emission radiance, L_s is the volume scattering radiance. Since we assume the scene (human body) does not emit energy itself, L_e should always be 0. L_i is the incoming radiance via either direct illumination or indirect illumination, as described in Eq. (9) in the main paper. $f_p(\mathbf{x}, -\mathbf{d}, \bar{\mathbf{d}})$ is the *phase function* that describes the probability of light scattering from direction $\bar{\mathbf{d}}$ to $-\mathbf{d}$ at point $\mathbf{x}$. Given these facts, Eq. (1) can be re-written as:

$$
\begin{aligned}
C_{pbr}(\mathbf{r}) &= \int_{t_n}^{t_f} T(t_n, t)\sigma_s(\mathbf{r}(t))L_s(\mathbf{r}(t), -\mathbf{d})dt \\
\text{s.t} \quad & \mathbf{r}(t) = \mathbf{o} + t\mathbf{d} \\
& T(t_n, t) = \exp\left(-\int_{t_n}^{t} \sigma_t(\mathbf{r}(s))ds\right) \\
& L_s(\mathbf{x}, -\mathbf{d}) = \int_{S^2} f_p(\mathbf{x}, -\mathbf{d}, \bar{\mathbf{d}})L_i(\mathbf{x}, -\bar{\mathbf{d}})d\bar{\mathbf{d}} \\
& \sigma_t(\mathbf{r}(t)) = \sigma_a(\mathbf{r}(t)) + \sigma_s(\mathbf{r}(t))
\end{aligned} \tag{3}
$$

which corresponds to Eq. (8) in the main paper. We next describe how to further approximate Eq. (3) with discrete samples.

The general idea is to sample offsets from the probability density function (PDF) of $T(t_n, t)$ and approximate the integral with Monte-Carlo integration. Define $\text{pdf}(t)$ as the PDF of t from which we sample M offsets $\{\bar{t}^{(i)}\}_{i=1}^{M}$, we have:

$$C_{pbr}(\mathbf{r}) \approx \frac{1}{M} \sum_{i=1}^{M} \frac{T(t_n, \bar{t}^{(i)}) \sigma_s(\mathbf{r}(\bar{t}^{(i)}))}{\text{pdf}(\bar{t}^{(i)})} L_s(\mathbf{r}(\bar{t}^{(i)}), -\mathbf{d}) \tag{4}$$

in the next two subsections, we describe how to sample from $\text{pdf}(t)$.

1.1. Homogeneous Volume

If we assume homogeneous volume, i.e. $\sigma_t(\mathbf{r}(t)) = \sigma_t$, then we can simplify $T(t_n, t)$ according to Beer's law:

$$T(t_n, t) = \exp\left(-\sigma_t |t - t_n|\right) \tag{5}$$

Sampling from $T(t_n, t)$ is equivalent to sampling from an exponential distribution, where the PDF is given by:

$$\text{pdf}(t) = c \exp\left(-\sigma_t |t - t_n|\right) \tag{6}$$

where c is a normalization constant. The cumulative distribution function (CDF) of t should satisfy:

$$\int_{t_n}^{\infty} c \exp\left(-\sigma_t |t - t_n|\right) dt = -\frac{c}{\sigma_t} \exp\left(-\sigma_t |t - t_n|\right) \Big|_{t_n}^{\infty} = \frac{c}{\sigma_t} = 1 \tag{7}$$

thus $c = \sigma_t$ and we have the following PDF and CDF of t accordingly:

$$\text{pdf}(t) = \sigma_t \exp\left(-\sigma_t |t - t_n|\right) \tag{8}$$

$$P(t) = 1 - \exp\left(-\sigma_t |t - t_n|\right) \tag{9}$$

1.2. Heterogeneous Volume

If the homogeneous assumption is lifted, we can still approximate the integral by dividing the ray segment (t_n, t_f) into intervals and assuming σ_t to be constant within each interval.

Formally, let us assume the ray segment is divided into $N - 1$ intervals, each defined by $[t^{(1)}, t^{(2)}), \cdots [t^{(i)}, t^{(i+1)}), \cdots [t^{(N-1)}, t^{(N)})$ with $t^{(1)} = t_n, t^{(N)} = t_f$. With our assumption on constant σ_t inside each interval, i.e. $\sigma_t(\mathbf{r}(t)) = \sigma_t(\mathbf{r}(t^{(i)})), \forall t \in [t^{(i)}, t^{(i+1)})$, define $\delta^{(i)} = |t^{(i+1)} - t^{(i)}|$, we define the following:

$$\begin{aligned} T(t^{(i)}, t^{(i+1)}) &= \exp\left(-\sigma(\mathbf{r}(t^{(i)}))\delta^{(i)}\right), \forall i \in \{1, \cdots, N-1\} \\ T(t^{(1)}, t^{(i)}) &= \prod_{j<i} T(t^{(j)}, t^{(j+1)}) \\ T(t^{(1)}, t^{(1)}) &= 1 \end{aligned}$$

To obtain the exact PDF from which we sample t, we extend Eq. (5) such that we sample from $T(t^{(1)}, t)$ that contains a homogeneous part and a heterogeneous part:

$$\begin{aligned} T(t^{(1)}, t) &= T(t^{(i)}, t) T(t^{(1)}, t^{(i)}) \\ &\text{s.t.} \quad t^{(i)} \le t < t^{(i+1)} \end{aligned} \tag{10}$$

where $T(t^{(1)}, t^{(i)})$ is the accumulated transmittance from the heterogeneous volume before $t^{(i)}$. Similar to Eq. (6) and Eq. (7) we can derive the normalization constant as $\sigma_t(\mathbf{r}(t)) = \sigma_t(\mathbf{r}(t^{(i)}))$, thus the PDF of t is:

$$\begin{aligned} \text{pdf}(t) &= \sigma_t(\mathbf{r}(t^{(i)})) T(t^{(i)}, t) T(t^{(1)}, t^{(i)}) \\ &= \sigma_t(\mathbf{r}(t)) T(t^{(1)}, t) \\ &\text{s.t.} \quad t^{(i)} \le t < t^{(i+1)} \end{aligned} \tag{11}$$

Plug Eq. (11) into Eq. (4), one will note that the $T(t_n, t)$ term is in both the numerator and the denominator. Thus Eq. (4) simplifies to:

$$C_{pbr}(\mathbf{r}) \approx \frac{1}{M} \sum_{i=1}^{M} \frac{\sigma_s(\mathbf{r}(\bar{t}^{(i)}))}{\sigma_t(\mathbf{r}(\bar{t}^{(i)}))} L_s(\mathbf{r}(\bar{t}^{(i)}), -\mathbf{d}) \tag{12}$$

Since we define the combined effect of $\frac{\sigma_s(\mathbf{r}(\bar{t}^{(i)}))}{\sigma_t(\mathbf{r}(\bar{t}^{(i)}))}$ and the phase function as a BRDF function, which becomes unrelated to σ_t, while we also need to be able to differentiate wrt. the geometry represented by σ_t, we use quadrature weights $\{w^{(i)}\}$ from NeRF [15], resulting in Eq. (10) in the main paper:

$$\begin{aligned}
C_{pbr}(\mathbf{r}) \approx & \sum_{i=1}^{M} w^{(i)} \frac{\sigma_s(\mathbf{r}(\bar{t}^{(i)}))}{\sigma_t(\mathbf{r}(\bar{t}^{(i)}))} L_s(\mathbf{r}(\bar{t}^{(i)}), -\mathbf{d}) \\
\text{s.t} \quad & \mathbf{r}(t) = \mathbf{o} + t\mathbf{d} \\
& w^{(i)} = T^{(i)} \left(1 - \exp(-\sigma_t(\mathbf{r}(\bar{t}^{(i)})\delta^{(i)})\right) \\
& T^{(i)} = \exp\left(-\sum_{j<i} \sigma_t(\mathbf{r}(\bar{t}^{(j)}))\delta^{(j)}\right) \\
& L_s(\mathbf{r}(\bar{t}^{(i)}), -\mathbf{d}) = \frac{f_p(\mathbf{r}(\bar{t}^{(i)}), -\mathbf{d}, \bar{\mathbf{d}}^{(i)})}{\text{pdf}(\bar{\mathbf{d}}^{(i)})} L_i(\mathbf{r}(\bar{t}^{(i)}, -\bar{\mathbf{d}}^{(i)}) \\
& \sigma_t(\mathbf{r}(t)) = \sigma_a(\mathbf{r}(t)) + \sigma_s(\mathbf{r}(t))
\end{aligned} \tag{13}$$

2. BRDF Definition

As mentioned in the main paper, we use a simplified version of Disney BRDF [2] to model the combined effect of the volumetric albedo $\frac{\sigma_s(\mathbf{r}(\bar{t}^{(i)}))}{\sigma_t(\mathbf{r}(\bar{t}^{(i)}))}$ and the phase function f_p. It takes predicted albedo α, roughness r and metallic m as inputs:

$$\frac{\sigma_s}{\sigma_t} f_p(\omega_o, \omega_i) = \text{BRDF}(\omega_o, \omega_i, \alpha, r, m, \mathbf{n}) \max(\mathbf{n} \cdot \omega_i, 0) \tag{14}$$

where ω_o and ω_i are the outgoing and incoming directions (i.e. surface to camera direction and surface to light direction, respectively). ω_h is the half vector between ω_o and ω_i, i.e. $\omega_h = \frac{\omega_o + \omega_i}{\|\omega_o + \omega_i\|_2}$. $\mathbf{n}$ is the surface normal. The BRDF is defined as follows:

$$\text{BRDF}(\omega_o, \omega_i, \alpha, r, m, \mathbf{n}) = (1 - m)\frac{\alpha}{\pi} + \frac{F(\omega_o, \alpha, m) D(\omega_h, \mathbf{n}, r) G(\omega_o, \omega_i, \mathbf{n})}{4(\mathbf{n} \cdot \omega_o)(\mathbf{n} \cdot \omega_i)} \tag{15}$$

in which the term $(1 - m)\frac{\alpha}{\pi}$ is the diffuse component while the remaining are specular components. For the specular component, F is the Fresnel-Schlick approximation to the exact Fresnel term, D is the isotropic GGX microfacet distribution [5] and G is Smith's shadowing term. They are defined as follows:

$$F(\omega_o, \alpha, m) = F_0(\alpha, m) + (1 - F_0(\alpha, m)) 2^{(-5.55473\omega_o \cdot \omega_h - 6.98316)\omega_o \cdot \omega_h} \tag{16}$$

$$D(\omega_h, \mathbf{n}, r) = \frac{r^2}{\pi((\mathbf{n} \cdot \omega_h)^2(r^2 - 1) + 1)^2} \tag{17}$$

$$\begin{aligned}
G(\omega_o, \omega_i, \mathbf{n}) &= G_1(\omega_o, \mathbf{n}) G_1(\omega_i, \mathbf{n}) \\
\text{s.t.} \quad F_0(\alpha, m) &= 0.04(1 - m) + \alpha m \\
G_1(\omega, \mathbf{n}) &= \frac{(\mathbf{n} \cdot \omega)}{(\mathbf{n} \cdot \omega)(1 - k) + k} \quad \text{and} \quad k = \frac{(r + 1)^2}{8}
\end{aligned} \tag{18}$$

note that for interpolating F, instead of using the typical Schlick approximation, we use the spherical Gaussian approximation [9, 10] which is slightly more efficient. G_1 is the *Schlick-GGX* approximation to the exact Smith's shadowing term.

3. Implementation Details

In this section, we provide more details about the implementation of our method.

3.1. Loss Function

In this subsection, we define $I^{(p)} \in [0,1]^3$ as the p-th pixel's color of the input image, $C_{rf}(\mathbf{r}^{(p)}) \in [0,1]^3$ as the predicted pixel color of the radiance field, $C_{pbr}(\mathbf{r}^{(p)}) \in [0,1]^3$ as the predicted pixel color of the physically based rendering, $\mathbf{M}^{(p)} \in \{0,1\}$ as the p-th pixel's ground truth binary mask value, $O^{(p)} \in [0,1]$ as the predicted ray opacity of the p-th pixel from the SDF-density field. Further, let P denote the set of all pixels in a training batch. We define the following loss functions:

Radiance Field (RF) Loss: We use L1 loss to measure the difference between the predicted pixel color from the radiance field and the input image:

$$\mathcal{L}_{\text{RF}} = \frac{1}{|P|} \sum_{p \in P} \left| C_{rf}(\mathbf{r}^{(p)}) - I^{(p)} \right| \tag{19}$$

Physically Based Rendering (PBR) Loss: We use L1 loss to measure the difference between the predicted pixel color from the physically based rendering and the input image:

$$\mathcal{L}_{\text{PBR}} = \frac{1}{|P|} \sum_{p \in P} \left| C_{pbr}(\mathbf{r}^{(p)}) - I^{(p)} \right| \tag{20}$$

Mask Loss: We use binary cross entropy loss to measure the difference between the predicted ray opacity and the ground truth binary mask $\mathbf{M}$:

$$\mathcal{L}_{\text{mask}} = \frac{1}{|P|} \sum_{p \in P} \left[\mathbf{M}^{(p)} \log O^{(p)} + (1 - \mathbf{M}^{(p)}) \log(1 - O^{(p)}) \right] \tag{21}$$

Eikonal Loss: We also apply Eikonel regularization to analytical gradient of the predicted SDF value at the canonical locations $\{\mathbf{x}_c^{(s)} = \text{LBS}^{-1}(\mathbf{x}_o^{(s)})\}_{s \in \mathcal{S}}$, where $\mathbf{x}_o^{(s)}$ is a sampled point on a ray in the observation space. $\mathcal{S}$ is the set of all sampled points in a training batch during ray marching of the radiance field, excluding those removed by occupancy grids.

$$\mathcal{L}_{\text{eikonal}} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \left(\left\| \nabla \text{SDF}(\mathbf{x}_c^{(s)}) \right\|_2 - 1 \right)^2 \tag{22}$$

Curvature Loss: We apply curvature regularization [22] to the same set of points on which we compute the Eikonal loss. The curvature loss is defined as follows:

$$\mathcal{L}_{\text{curv}} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \left(\mathbf{n}^{(s)} \cdot \mathbf{n}_\epsilon^{(s)} - 1 \right)^2 \tag{23}$$

where $\mathbf{n}^{(s)}$ is the analytical normal at the canonical location $\mathbf{x}_c^{(s)}$, i.e. *normalized* analytical gradient of the SDF $\mathbf{n}^{(s)} = \frac{\nabla \text{SDF}(\mathbf{x}_c^{(s)})}{\|\nabla \text{SDF}(\mathbf{x}_c^{(s)})\|_2}$. $\mathbf{n}_\epsilon^{(s)}$ is the analytical normal of a nearby point $\mathbf{x}_c^{(s)} + \epsilon \mathbf{t}^{(s)}$, here $\epsilon = 0.0001$ and $\mathbf{t}^{(s)}$ is a random direction that is tangential to normal direction $\mathbf{n}^{(s)}$.

Local Smoothness Loss: We apply local smoothness regularization on predicted albedo α, roughness r and metallic m values, in a similar way to [8, 24]:

$$\mathcal{L}_{\text{smoothness}} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \left[\sum_{i=1}^{N^{(p)}} w^{(p,i)} \Delta\alpha^{(p,i)} + \sum_{i=1}^{N^{(p)}} w^{(p,i)} \Delta r^{(p,i)} + \sum_{i=1}^{N^{(p)}} w^{(p,i)} \Delta m^{(p,i)} \right] \tag{24}$$

$$\text{s.t.} \quad \Delta\alpha^{(p,i)} = \frac{\alpha^{(p,i)} - \alpha_\epsilon^{(p,i)}}{\max(\max(\alpha^{(p,i)}, \alpha_\epsilon^{(p,i)}), 1e-6)}$$

$$\Delta r^{(p,i)} = \frac{r^{(p,i)} - r_\epsilon^{(p,i)}}{\max(\max(r^{(p,i)}, r_\epsilon^{(p,i)}), 1e-6)}$$

$$\Delta m^{(p,i)} = \frac{m^{(p,i)} - m_\epsilon^{(p,i)}}{\max(\max(m^{(p,i)}, m_\epsilon^{(p,i)}), 1e-6)}$$

where $N^{(p)}$ is the number of samples on ray p. $w^{(p,i)}$ is the quadrature weight of the i-th sample on ray p. $\alpha_\epsilon^{(p,i)}$, $r_\epsilon^{(p,i)}$, $m_\epsilon^{(p,i)}$ are albedo, roughness and metallic queried at a perturbated location near the i-th sample of ray p.

Lipschitz Bound Loss: Lastly, we apply the Lipschitz bound loss [13] to enforce Lipschitz smoothness of the material MLP. [22] uses the same technique to regularize the radiance MLP. Formally, given an MLP's i-th layer $y = \text{actv}(W_i x + b_i)$ along with a trainable Lipschitz bound k_i, the layer is reparameterized as

$$y = \text{actv}\left(\widehat{W}_i x + b_i\right), \quad \widehat{W}_i = \text{norm}(W_i, \text{softplus}(k_i)) \tag{25}$$

where $\text{norm}(\cdot, \cdot)$ normalizes the weight matrix W_i by rescaling each row such that the row sum's absolute value is less than or equal to the $\text{softplus}(k_i)$. The Lipschitz bound loss is defined as follows:

$$\mathcal{L}_{\text{Lip}} = \prod_{i=1}^{L} \text{softplus}(k_i) \tag{26}$$

where L is the number of layers in the MLP.

Combined Loss: The final loss function is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{RF}} + \lambda_{\text{PBR}}\mathcal{L}_{\text{PBR}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{eikonal}}\mathcal{L}_{\text{eikonal}} + \lambda_{\text{curv}}\mathcal{L}_{\text{curv}} + \lambda_{\text{smoothness}}\mathcal{L}_{\text{smoothness}} + \lambda_{\text{Lip}}\mathcal{L}_{\text{Lip}} \tag{27}$$

where we set $\lambda_{\text{PBR}} = 0.2$, $\lambda_{\text{mask}} = 0.1$, $\lambda_{\text{eikonal}} = 0.1$, $\lambda_{\text{smoothness}} = 0.01$. We set $\lambda_{\text{curv}} = 1.5$ for the first 12.5k iterations and 0 after that. We set $\lambda_{\text{Lip}} = 1e-5$ after 12.5k iterations and 0 before that.

3.2. Albedo Evaluation

For evaluating the predicted albedo image, we first align the predicted albedo with the ground truth albedo. Given N samples on a ray, the predicted albedo of a ray $\mathbf{r}$ is defined as follows:

$$\hat{A}(\mathbf{r}) = \sum_{i=1}^{N} w^{(i)}\alpha^{(i)} \tag{28}$$

we compute per-channel scaling factors $\mathbf{s} = (s_r, s_g, s_b)$ to align the predicted albedo with the ground truth albedo. Given $A_r^{(p)}$ as the ground truth red albedo of the p-th pixel, the following equation is computed for s_r:

$$s_r = \frac{\sum_{p\in P} \hat{A}_r(\mathbf{r}^{(p)}) \cdot A_r^{(p)}}{\sum_{p\in P} \hat{A}_r(\mathbf{r}^{(p)}) \cdot \hat{A}_r(\mathbf{r}^{(p)})} \tag{29}$$

while s_g and s_b are computed similarly. We evaluate image quality metrics (i.e. PSNR, SSIM, LPIPS) on the aligned predicted albedo. We visualize the aligned predicted albedo on synthetic data and the non-aligned predicted albedo on real data.

3.3. Additional Implementation Details

We use a mixture of 64 spherical Gaussians to represent environment lighting during training. During relighting, we do not use indirect illumination as the learned radiance field on training data is not applicable to the new lighting condition. We clip the pixel prediction from both the radiance field and the PBR to $[0, 1]$ and apply standard gamma correction to obtain the final image in sRGB space. For a fair comparison, we also integrate light importance sampling into R4D for relighting, which directly samples 1024 directions on the high-resolution environment map.

We also implement the pose optimization module following [23]. This module is enabled for the SyntheticHuman-Relit dataset as the motion is more complex compared to other datasets, while the original ground-truth SMPL estimations from [19] are also slightly misaligned with the image.

To stay consistent with R4D and [11, 24], we also calibrate our albedo prediction to the range $[0.03, 0.8]$. We note this technique is especially useful when the subject wears near-black clothes (i.e. albedo < 0.1 for all channels).

Temporal Occupancy Grids: A common technique to reduce computation is to maintain an occupancy grid to mark occupied voxels during training and skip unoccupied voxels during ray marching/tracing [1, 3, 12, 16] . This also applies to temporal reconstruction as one can define the occupancy grid as the union of all shapes from different time steps [7]. To further reduce the computational cost, we employ a 4D occupancy grid structure [18] in which we maintain a 64^3 occupancy grid for each training frame. At the beginning of training, we first use a single occupancy grid for all frames. After we have attained a reasonable SDF we re-initialize the occupancy grid for each frame using the learned canonical SDF.

4. SyntheticHuman-Relit Dataset

To properly compare with R4D on relighting of training poses, we created a new dataset, SyntheticHuman-Relit, which is a subset of the SyntheticHuman dataset [19] relit using new material and lighting conditions. The dataset consists of 2 synthetic humans (*Jody* and *Leonard*), each rotating in front of a fixed camera.

We note that the original SyntheticHuman dataset was rendered under studio lighting and the materials were overly specular compared to real humans. We thus adjusted the materials and re-rendered the dataset under more natural lighting conditions. See a comparison of the original SyntheticHuman dataset and the new SyntheticHuman-Relit dataset in Fig. 1.

To test relighting on training poses, we further re-rendered each training pose of the SyntheticHuman-Relit dataset under a random environment map that was not used in the training set.

SyntheticHuman - Jody | SyntheticHuman - Leonard | SyntheticHuman-Relit - Jody | SyntheticHuman-Relit - Leonard

Figure 1. **Comparison between the SyntheticHuman dataset and the SyntheticHuman-Relit dataset.** Note that the SyntheticHuman dataset is overly specular compared to real humans, while the light sources are also studio-like. In contrast, SyntheticHuman-relit adopts a more diffuse appearance which is closer to real humans, while the subjects are lit under natural, outdoor illumination.

5. Additional Quantitative Results

The per-subject and average metrics of R4D, R4D*, and Ours are reported in Tab. 1. We also tested a variant of our approach that does not calibrate the albedo into the range $[0.03, 0.8]$, denoted as Ours†. Since R4D* and Ours achieve overall better performance than their variants (R4D and Ours†) on the RANA dataset, we only report R4D* and Ours on the SyntheticHuman-Relit dataset in Tab. 2. We also additionally report ARAH [23]'s results on geometry reconstruction, evaluated by the normal error metric. Albedo estimation and relighting are not evaluated as ARAH does not predict the intrinsic properties of avatars.

6. Additional Qualitative Results

We present additional qualitative results on the RANA dataset in Fig. 2 and Fig. 3, while Fig. 4 shows additional qualitative results on the SyntheticHuman-Relit dataset. We also present more qualitative results on the PeopleSnapshot dataset in Fig. 5 and Fig. 6. We additionally show qualitative results on the ZJU-MoCap [17] dataset in Fig. 7.

7. Limitations and Future Work

Since we focus on video sequences for people holding still and rotating in front of the camera, we did not consider pose-dependent non-rigid motion, similar to the assumption of [6, 7]. Our approach can also fail if estimated poses or segmentation masks are too noisy. Furthermore, our canonical pose representation is not suitable for the animation of very loose clothes such as skirts or capes.

Our approach is also relatively slow at inference time, requiring about 20 seconds to render a single 540x540 image on a single RTX 3090 GPU. Regardless, our model's outputs are fully compatible with existing physically based rendering pipelines, further acceleration can be achieved by using more optimized implementation of volumetric scattering at inference time.

In the future, we plan to extend our approach to more challenging scenarios, such as modeling large pose-dependent non-rigid deformations. Applying physically based inverse rendering for relightable scenes and avatar reconstruction is also a promising direction [4, 14].

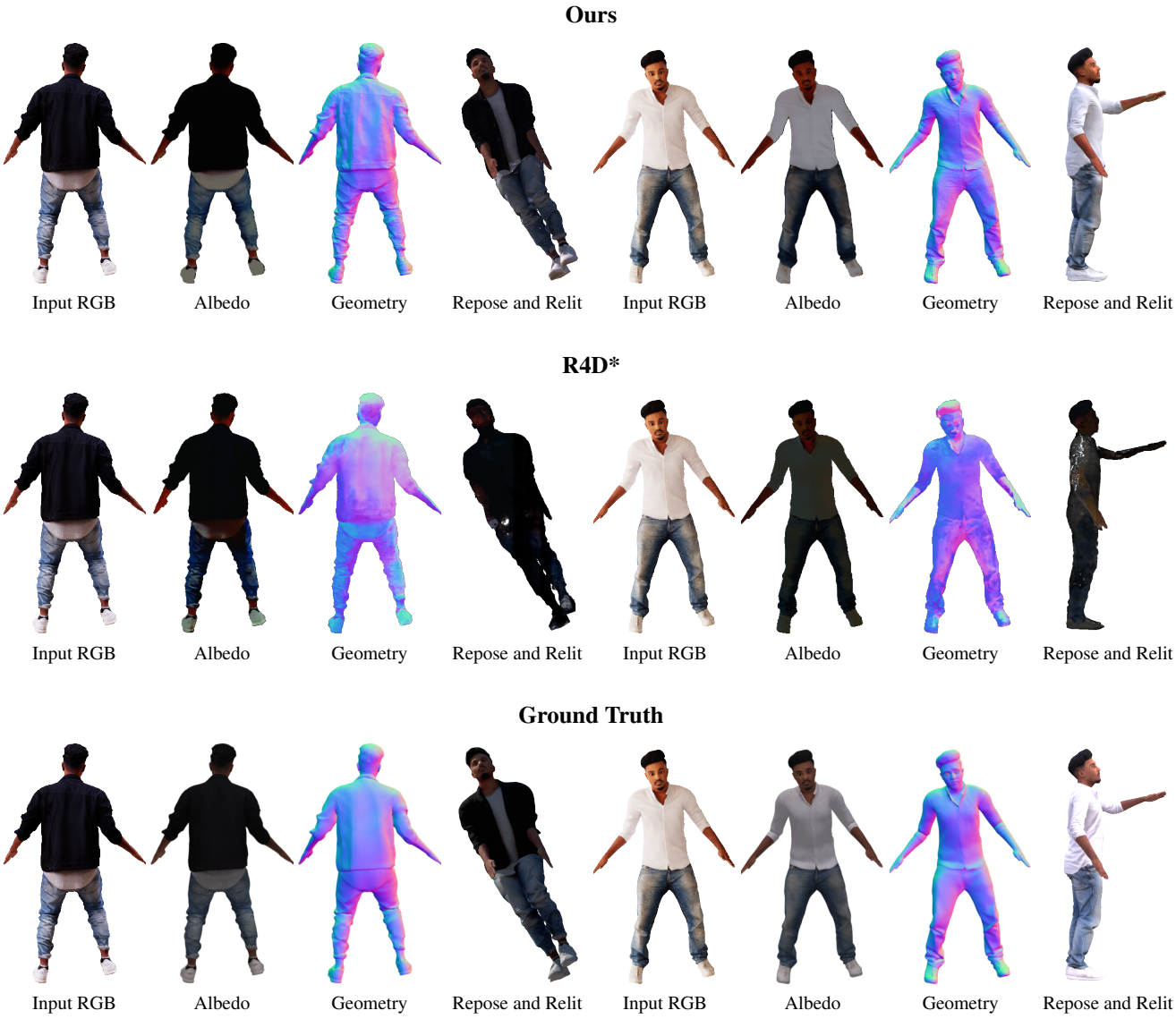

Figure 2. **Additional qualitative results on the RANA dataset.** We note that our method removes the shadow from the estimated albedo, whereas R4D* bakes shadow into albedo (column 2). On another subject, we produce albedo close to ground truth while R4D* produces overly dark albedo (column 6).

References

[1] Alex Yu and Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[2] Brent Burley. Physically-based shading at disney. In *Proc. of SIGGRAPH*, 2012. 3

[3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022. 5

[4] Chen Guo, Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Vid2avatar: 3d avatar reconstruction from videos in the wild via self-supervised scene decomposition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6

[5] Eric Heitz. Sampling the ggx distribution of visible normals. *Journal of Computer Graphics Techniques (JCGT)*, 7(4):1–13, 2018. 3

[6] Umar Iqbal, Akin Caliskan, Koki Nagano, Sameh Khamis, Pavlo Molchanov, and Jan Kautz. Rana: Relightable articulated neural avatars. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 6

[7] Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Instantavatar: Learning avatars from monocular video in 60 seconds. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5, 6

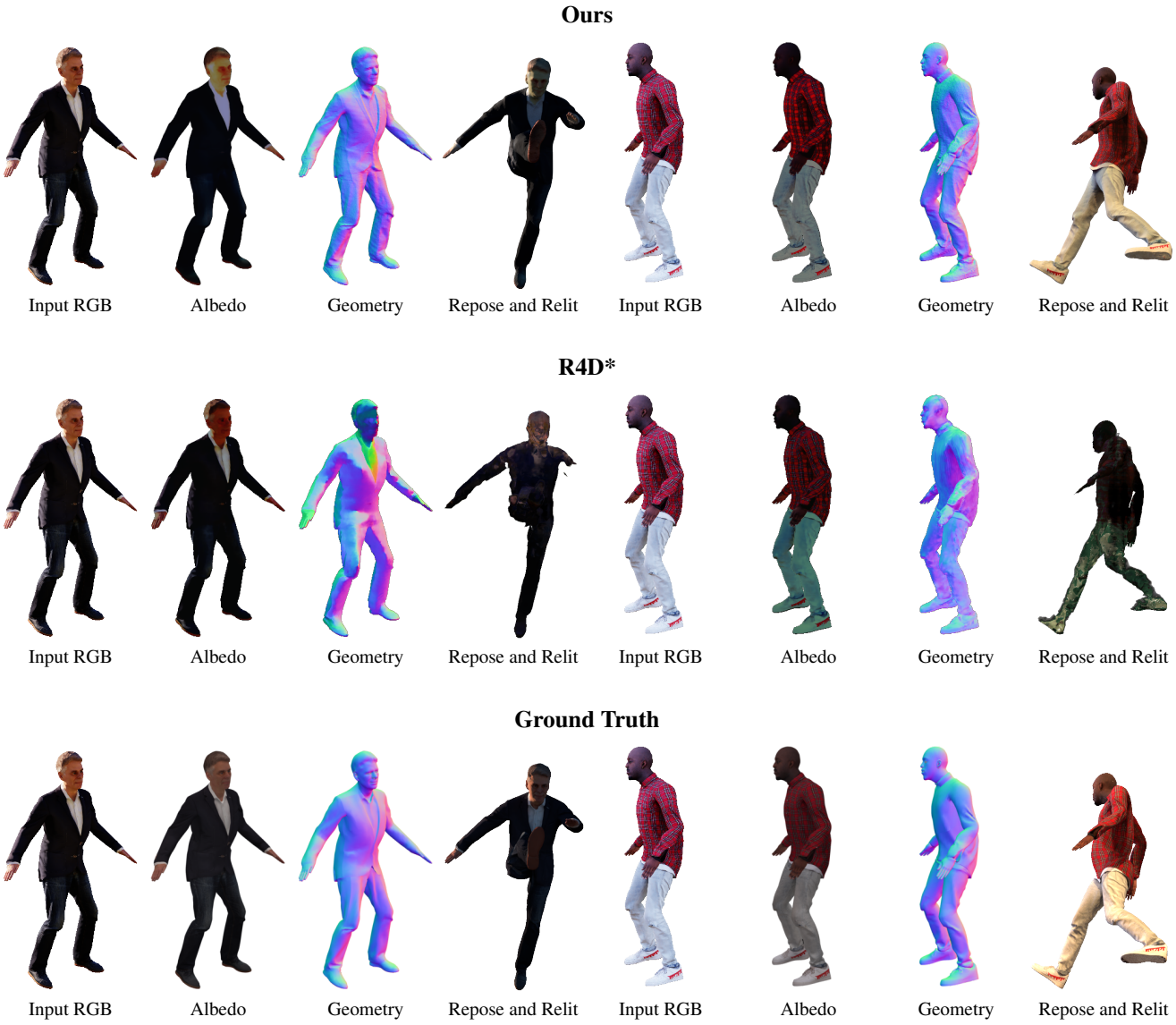

Figure 3. **Additional qualitative results on the RANA dataset.** R4D* demonstrates various failures in albedo estimation such as baked lighting. The normal estimation is also inaccurate. In contrast, our method produces accurate normals and in general does not bake lighting into albedo.

[8] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4

[9] Brian Karis. Real shading in unreal engine 4. In *Proc. of SIGGRAPH*, 2013. 3

[10] Sébastien Lagarde. Spherical gaussian approximation for blinn-phong, phong and fresnel. https://seblagarde.wordpress.com/2012/06/03/spherical-gaussien-approximation-for-blinn-phong-phong-and-fresnel/, 2012. 3

[11] Greg Ward Larson and Rob Shakespeare. *Rendering with Radiance: The Art and Science of Lighting Visualization*. Morgan Kaufmann Publishers Inc., 1998. 5

[12] Ruilong Li, Hang Gao, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: Efficient sampling accelerates nerfs. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 5

[13] Hsueh-Ti Derek Liu, Francis Williams, Alec Jacobson, Sanja Fidler, and Or Litany. Learning smooth neural functions via lipschitz regularization. In *Proc. of SIGGRAPH*, 2022. 5

[14] Jia-Wei Liu, Yan-Pei Cao, Tianyuan Yang, Eric Zhongcong Xu, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Hosnerf:

Figure 4. **Additional qualitative results on the SyntheticHuman-Relit dataset.** We note that R4D* tends to bake lighting and shadows not only into albedo but also into predicted normals. In contrast, our normals are obtained from the underlying SDF field and thus would not have such baked effects.

Dynamic human-object-scene neural radiance fields from a single video. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 6

[15] 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. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 3

[16] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash

encoding. *ACM Trans. on Graphics*, 2022. 5

[17] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6

[18] Sida Peng, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Representing volumetric videos as dynamic mlp maps. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5

[19] Sida Peng, Shangzhan Zhang, Zhen Xu, Chen Geng, Boyi Jiang, Hujun Bao, and Xiaowei Zhou. Animatable neural implicit surfaces for creating avatars from videos. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2024. 5, 6

[20] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*, chapter 14. The MIT Press, 4th edition, 2023. 1

[21] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*, chapter 11. The MIT Press, 4th edition, 2023. 1

[22] Radu Alexandru Rosu and Sven Behnke. Permutosdf: Fast multi-view reconstruction with implicit surfaces using permutohedral lattices. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4, 5

[23] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022. 5, 6

[24] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul E. Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. on Graphics*, 40(6):237:1–237:18, 2021. 4, 5

Subject	Method	Albedo			Normal	Relighting (Novel Pose)		
		PSNR ↑	SSIM ↑	LPIPS ↓	Error ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Subject 01	ARAH	-	-	-	12.89 °	-	-	-
	R4D	20.64	0.7673	0.2199	64.07 °	11.73	0.7865	0.2028
	R4D*	20.04	0.8525	0.2079	33.61 °	18.22	0.8425	0.1612
	Ours†	23.69	0.7998	0.1916	**11.35** °	18.35	0.8727	**0.1200**
	Ours	**24.11**	**0.8679**	**0.1827**	12.05 °	**18.48**	**0.8859**	0.1219
Subject 02	ARAH	-	-	-	11.92 °	-	-	-
	R4D	15.14	0.8089	0.2926	30.20 °	15.08	0.8361	0.1954
	R4D*	12.13	0.7690	0.2599	28.34 °	14.38	0.8128	0.1787
	Ours†	20.25	0.8733	0.1898	**9.27** °	18.86	0.8781	0.1336
	Ours	**20.94**	**0.8892**	**0.1854**	9.29 °	**19.08**	**0.8812**	**0.1323**
Subject 05	ARAH	-	-	-	9.78 °	-	-	-
	R4D	19.66	0.8223	0.2484	31.18 °	16.59	0.8354	0.1916
	R4D*	19.74	0.8151	0.2488	26.14 °	**17.72**	0.8469	0.1780
	Ours†	21.06	0.8159	0.2262	**9.51** °	17.40	0.8750	0.1466
	Ours	**22.24**	**0.8591**	**0.2071**	9.52 °	17.47	**0.8769**	**0.1453**
Subject 06	ARAH	-	-	-	12.06 °	-	-	-
	R4D	17.26	0.5954	0.3466	81.79 °	7.31	0.7567	0.2821
	R4D*	21.57	0.7992	0.2177	25.83 °	17.54	0.8866	0.1636
	Ours†	21.07	0.7093	0.2241	8.91 °	17.89	0.8647	0.1294
	Ours	**22.94**	**0.8233**	**0.1928**	**8.89** °	**18.14**	**0.8932**	**0.1271**
Subject 33	ARAH	-	-	-	10.28 °	-	-	-
	R4D	17.95	0.8335	0.1900	27.53 °	16.08	0.8202	0.1960
	R4D*	18.35	0.8426	0.1887	25.24 °	16.78	0.8173	0.1859
	Ours†	**21.78**	0.8395	**0.1259**	**9.07** °	17.62	0.8352	**0.1332**
	Ours	21.67	**0.8703**	0.1351	9.52 °	**18.03**	**0.8426**	0.1366
Subject 36	ARAH	-	-	-	11.62 °	-	-	-
	R4D	20.38	0.9091	0.1844	43.44 °	15.99	0.8200	0.1899
	R4D*	23.80	**0.9100**	0.1611	24.76 °	17.05	0.8574	0.1707
	Ours†	24.30	0.7946	0.1739	**9.09** °	17.25	0.8520	0.1308
	Ours	**24.88**	0.8900	**0.1324**	9.22 °	**17.46**	**0.8726**	**0.1284**
Subject 46	ARAH	-	-	-	**10.38** °	-	-	-
	R4D	16.40	0.8381	0.1455	32.64 °	16.05	0.8289	0.1720
	R4D*	18.13	0.8777	0.1238	33.27 °	16.30	0.8338	0.1649
	Ours†	22.17	0.9314	0.0744	10.41 °	16.89	0.8377	**0.0965**
	Ours	**22.47**	**0.9391**	**0.0725**	10.69 °	**17.08**	**0.8406**	0.1000
Subject 48	ARAH	-	-	-	**10.13** °	-	-	-
	R4D	18.50	0.8502	0.3037	30.67 °	16.10	0.8224	0.1840
	R4D*	12.10	0.7370	0.2264	21.84 °	14.98	0.7985	0.1776
	Ours†	23.28	0.9075	**0.1838**	10.32 °	19.50	0.8823	**0.1307**
	Ours	**23.36**	**0.9137**	0.1857	10.49 °	**19.70**	**0.8849**	0.1313
Average	ARAH	-	-	-	11.13 °	-	-	-
	R4D	18.24	0.7780	0.2414	42.69 °	14.37	0.8133	0.2017
	R4D*	18.23	0.8254	0.2043	27.38 °	16.62	0.8370	0.1726
	Ours†	22.20	0.8339	0.1737	**9.74** °	17.97	0.8622	**0.1276**
	Ours	**22.83**	**0.8816**	**0.1617**	9.96 °	**18.18**	**0.8722**	0.1279

Table 1. **Metrics on the RANA dataset.**

Subject	Method	Albedo			Normal	Relighting (Training Pose)		
		PSNR ↑	SSIM ↑	LPIPS ↓	Error ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Jody	ARAH	-	-	-	15.79 °	-	-	-
	R4D*	17.95	0.7275	0.2319	33.51 °	21.85	0.9012	0.1277
	Ours	**23.10**	**0.8353**	**0.1584**	**13.90** °	**22.24**	**0.9336**	**0.1055**
Leonard	ARAH	-	-	-	15.96 °	-	-	-
	R4D*	25.67	**0.8838**	0.1841	25.93 °	23.23	0.9216	0.1296
	Ours	**26.98**	0.7872	**0.1568**	**14.45** °	**24.23**	**0.9490**	**0.0954**
Average	ARAH	-	-	-	15.88 °	-	-	-
	R4D*	21.81	0.8057	0.2080	29.72 °	22.57	0.9123	0.1283
	Ours	**25.04**	**0.8113**	**0.1567**	**14.18** °	**23.24**	**0.9413**	**0.1005**

Table 2. **Metrics on the SyntheticHuman-Relit dataset.**

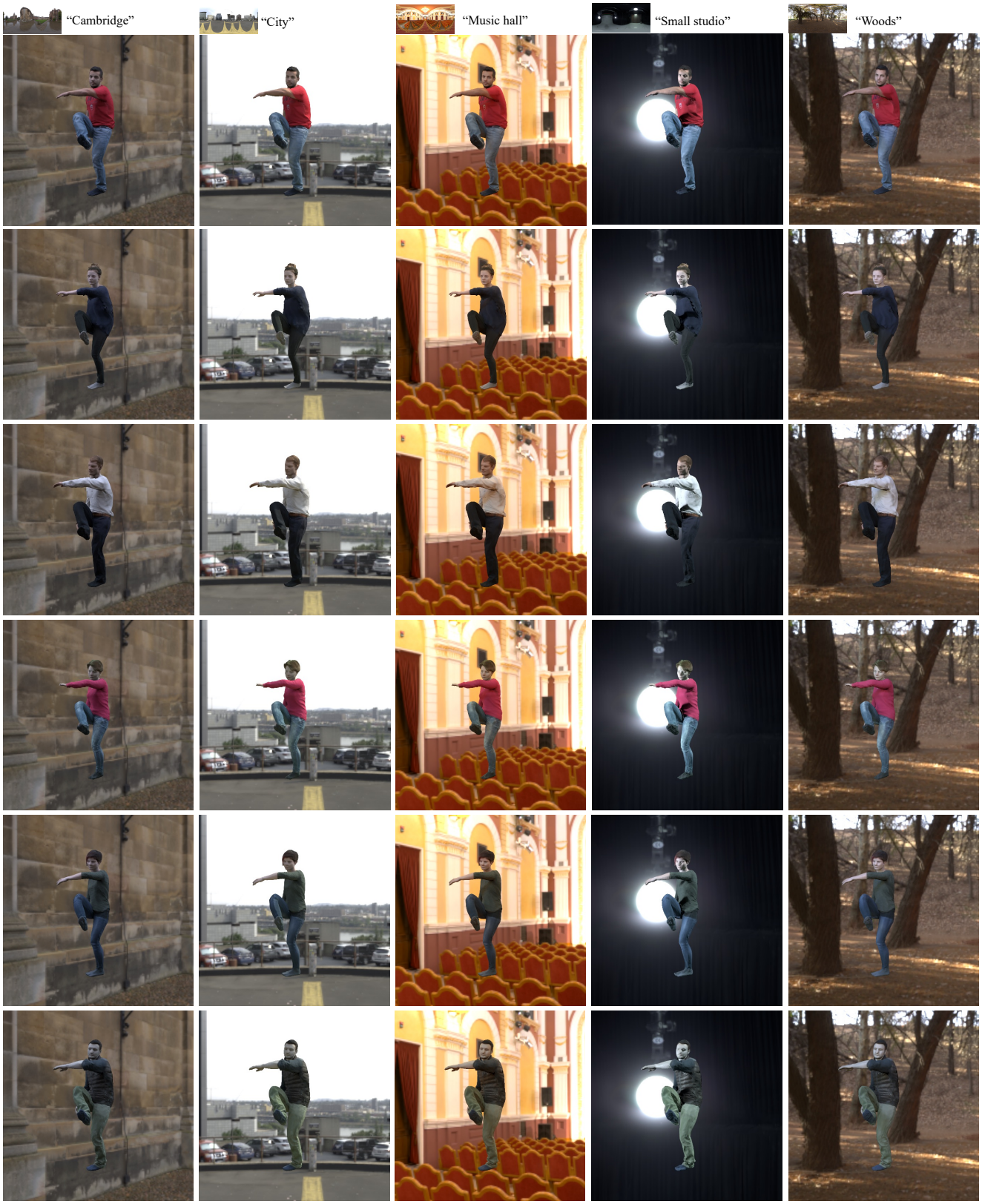

Figure 5. **Additional qualitative results on the PeopleSnapshot dataset.** We show 6 subjects from the dataset under novel pose and novel illuminations.

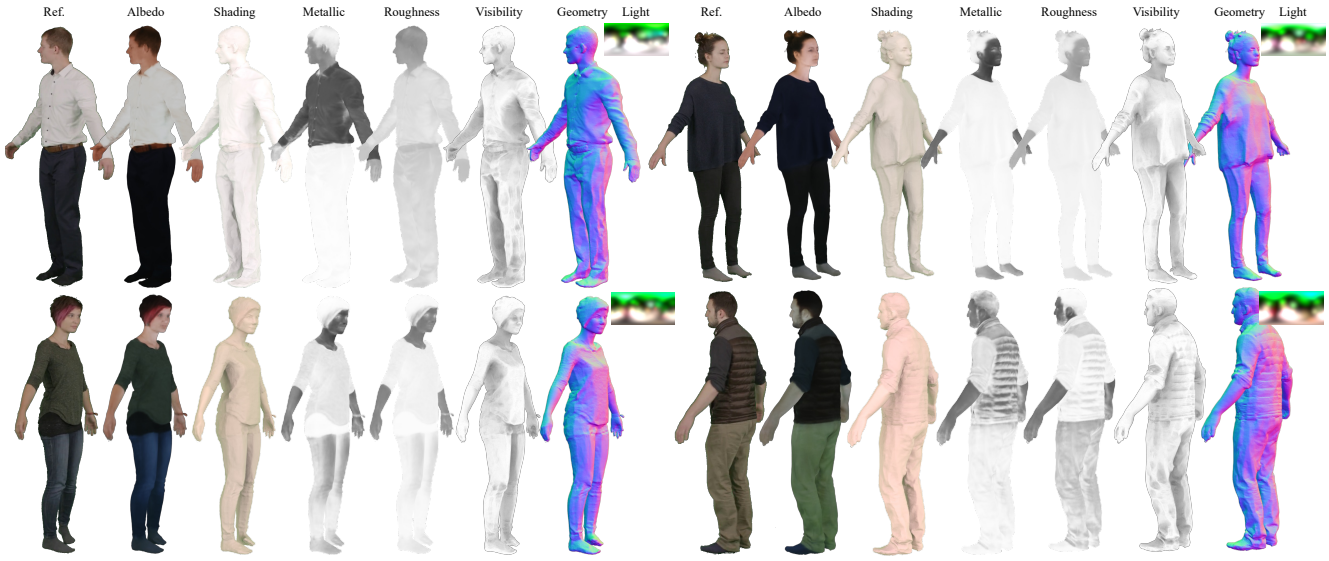

Figure 6. **Additional results on learned intrinsic properties from the PeopleSnapshot dataset.**

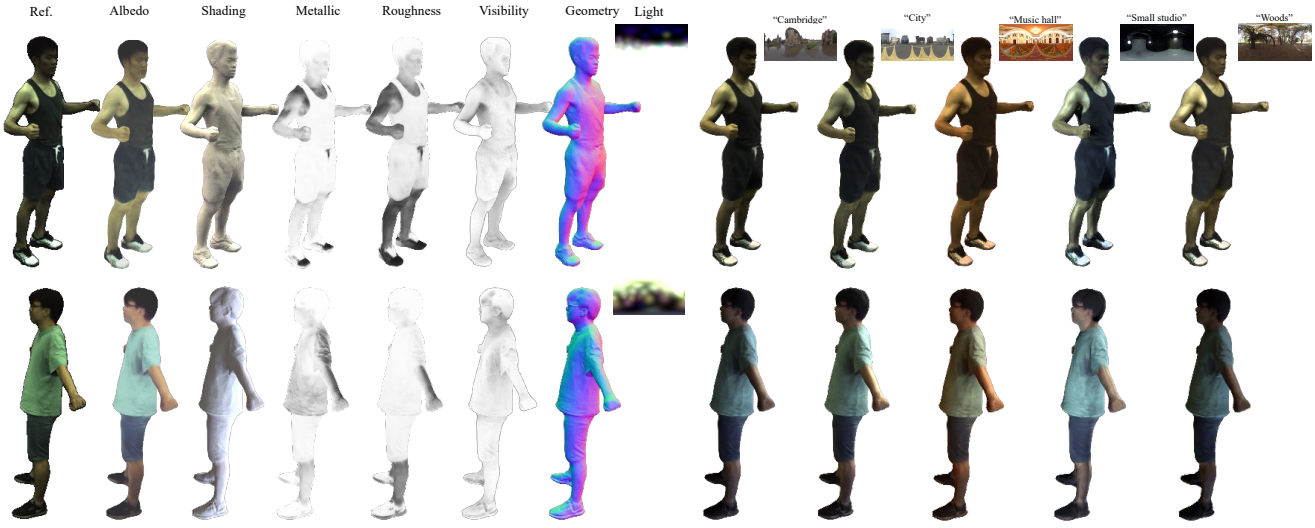

Figure 7. **Results on the ZJU-MoCap dataset.** We show results on monocular input (top row) and 4-view input (bottom row). Note that the environment lighting in the ZJU-MoCap dataset is relatively dark, resulting in darker albedo in the monocular setup.