URHand: Universal Relightable Hands

Supplementary Material

1. Demo Video

We provide the supplementary video on our project page (https://frozenburning.github.io/projects/urhand), which includes more visual results and additional discussions of our work. Specifically, it contains:

- Motivation and key features of URHand.
- An animated overview and illustration of the proposed framework.
- Video comparisons with baseline methods.
- Additional qualitative results with diverse identities, including **1)** relighting with monochrome directional light, **2)** relighting with arbitrary environment map, and **3)** quick personalization from a phone scan with corresponding relighting results with environment maps.

2. Limitation and Future Works

Since we learn global light transport with far-field lighting, it does not guarantee correct light transport with near-field lighting. Nevertheless, our work achieves plausible near-field relighting similarly to [1, 3, 16, 17]. Currently the quick personalization requires the complete mean texture of a target hand. Thus, it does not work with a single image. One future work would be inpainting the texture from a single image to enable single-view relightable hand reconstruction. As our hand model is only driven by hand poses, it cannot capture appearance variations due to blood pressure or temperature changes. As recently demonstrated in [10], photorealistic relightable hands can be used to augment training data for image-based pose regression tasks. Using URHand to synthesize large-scale two-hand or hand-to-object interaction images with diverse identities is also fruitful.

3. Personalization from a Phone Scan

In this section, we demonstrate how to quickly adapt URHand to a personalized use case from a phone scan. We followed the hand avatar creation pipeline of UHM [11]. To be specific, we use a single iPhone 12 to scan a hand, which incorporates a depth sensor that can be used to extract better geometry of the user's hand. Our phone scans include hands with neutral finger poses and static 3D global translations with varying 3D global rotations to expose most of the hand surfaces.

We pre-process the phone scan with 1) our in-house 2D hand keypoint detector to obtain 2D hand joint coordinates and 2) RVM [7] to obtain the foreground mask of the phone scan. After the preprocessing, we optimize 3D global rotation, 3D pose, 3D global translation, and ID code of the phone scan. The 3D global rotation, 3D pose, and 3D global translation are optimized for each frame, and a single ID code is shared across all frames as all frames are from a single identity. We optimize them by minimizing 1) $L1$ distance between projected 2D joint coordinates and targets 2) $L1$ distance between differentiably rendered masks and targets with weight 50, and 3) $L1$ distance between differentiably rendered depth maps and targets with weight 100. After the optimization, we unwrap per-frame images to UV space and average intensity values at each texel considering the visibility to get the unwrapped texture map.

To remove shadows from the unwrapped textures, we first obtain the average color of the foreground pixels of captured images. Then, we optimize shadow as a 1-channel difference (*i.e.*, darkness difference) between the averaged color and the captured image in the UV space. To prevent the shadow from dominating local sharp textures (*i.e.*, hairs and tattoos), we apply a total variation regularizer to the shadow. The unwrapped texture without the shadow is simply obtained by dividing the unwrapped texture by the shadow. We empirically observed that such a statistical approach produces better shadow than the physics-based approach of HARP [5], which assumes a single point light, as there is often more than one light source in the scan environment. Then, we take this texture map after shadow removal as the input to URHand for relighting without any finetuning. Besides, for more details about the quick hand avatar creation, please refer to UHM [11].

To demonstrate that our universal relightable prior achieves generalization beyond the training data, we present relighting results of phone scan personalization with tattoos in Figure 1 and Figure 2, where we relight the hand with diverse illuminations and poses without retraining URHand. The subjects were captured only in a neutral pose with an unknown illumination using a single iPhone 12. This is especially challenging as the tattoo is an out-of-distribution appearance from our training data.

Specifically, Figure 1 shows that our relightable hand prior generalizes to clearly novel identities (with tattoos which are never observed during training), diverse natural illuminations, and novel poses (the input phone scans contain only neutral poses) without noticeable artifacts. Moreover, we present relighting results with extreme illuminations as RGB panels in Figure 2 which further demonstrates the generalizability of URHand to any lighting conditions.

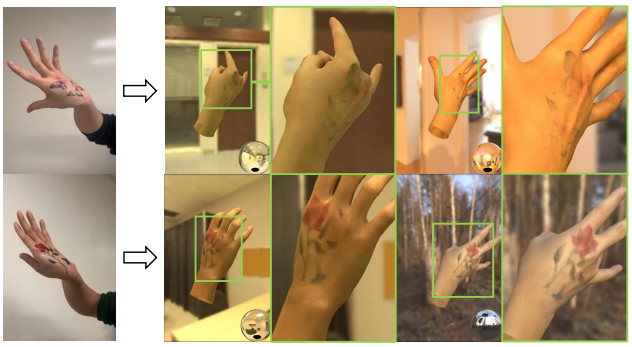

Figure 1. **Relighting results (environment maps) from a phone scan with tattoos.** We capture the hand with tattoos only in a neutral pose with unknown illumination using a single iPhone 12. By applying the universal prior of URHand, we can relight the hand with diverse illuminations and poses without retraining.

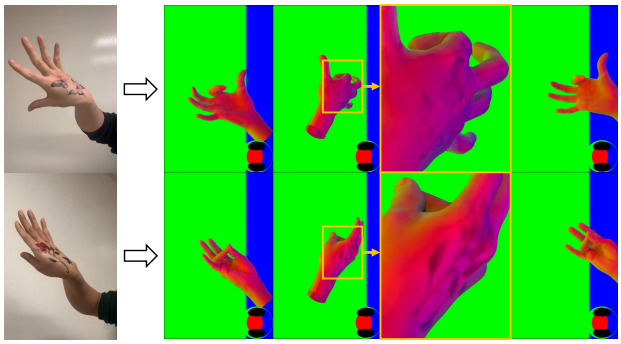

Figure 2. **Relighting results (extreme RGB panel lighting) from a phone scan with tattoos.** We capture the hand with tattoos only in a neutral pose with unknown illumination using a single iPhone 12. By applying the universal prior of URHand, we can even render the hand in extreme out-of-distribution illuminations.

4. Network Architecture

In this section, we provide the details of our network architecture and hyperparameters for our hand geometry model (Sec. 4.1), physical branch (Sec. 4.2), and neural branch (Sec. 4.3), respectively.

4.1. Hand Geometry Autoencoder

We design an autoencoder [11] to obtain accurate hand tracking and geometry from input fully lit frames similar to [9]. The architecture of this autoencoder $\{\mathcal{E}_{\text{id}}, \mathcal{D}_{\text{id}}, \mathcal{E}_\theta, \mathcal{D}_\theta\}$ is illustrated in Figure 3. Specifically, it consists of an identity encoder $\mathcal{E}_{\text{id}}$, an identity decoder $\mathcal{D}_{\text{id}}$, a pose encoder $\mathcal{E}_\theta$, and a pose decoder $\mathcal{D}_\theta$.

The identity encoder $\mathcal{E}_{\text{id}}$ takes as input the depth map in the neutral pose and the coordinates of joints in the neutral pose, which predicts the mean and variance of the distribution of the identity code (*i.e.* ID code). The inputs of the identity encoder have normalized viewpoints by rigidly aligning them to a reference coordinate system; hence both pose and viewpoints are normalized and only identity information is included in them. The identity decoder $\mathcal{D}_{\text{id}}$ learns to decode the identity-dependent offset of joints and vertices from the ID code z. The identity-dependent offset of joints is responsible for adjusting 3D joint coordinates in the template space for each identity, and the offset of vertices are for adjusting 3D vertices in the template space for each identity. The pose encoder $\mathcal{E}_\theta$ directly regresses hand pose θ from the input image and the coordinates of joints. The pose decoder $\mathcal{D}_\theta$ learns to predict the pose-dependent offset of vertices given the pose θ and ID code z. To get posed 3D meshes, we apply the three types of correctives to the template mesh and perform linear blend skinning with the estimated 3D pose from the pose encoder.

We train this autoencoder $\{\mathcal{E}_\theta, \mathcal{D}_\theta, \mathcal{E}_{\text{id}}, \mathcal{D}_{\text{id}}\}$ on fully lit frames with all identities to obtain a general hand tracker. The autoencoder is trained by minimizing 1) $L1$ distance between joint coordinates, 2) point-to-point $L1$ distance from 3D scans with weight 10, 3) KL-divergence of the ID code with weight 0.001, and 4) various regularizers like Moon *et al*. [9]. We freeze it during the training of URHand as well as the quick personalization from phone scans.

4.2. Physical Branch

The physical branch of URHand consists of a 2D U-Net [15] $\mathcal{F}_{\text{G}}$ and a parametric BRDF [2] $\mathcal{F}_{\text{pb}}$, where only the U-Net $\mathcal{F}_{\text{G}}$ contains optimizable parameters. The U-Net encoder is a 6-layer convolutional neural network (CNN) with channel sizes $(3, 64, 64, 64, 64, 64, 64)$, which takes as input the mean texture $\mathcal{T} \in \mathbb{R}^{1024\times1024\times3}$. To get mean texture, we project the visible pixels of fully lit images onto the UV texture map based on the tracked meshes in neutral poses and take the weighted average based on the surface normals across 5 frames. The hand pose θ is tiled into a UV-aligned 2D feature map θ', concatenated with the output feature from the U-Net encoder as a joint feature $\mathbf{F}_{\theta,\text{id}}$, and passed to the U-Net decoder. The U-Net decoder is a 6-layer CNN with skip connections from the U-Net encoder, with channel sizes $(64, 64, 64, 64, 64, 64, 2)$. We use a transposed convolution layer followed by bilinear interpolation as the upsampling layer in the U-Net decoder. The U-Net decoder predicts the displacement map $\delta d \in \mathbb{R}^{1024\times1024}$ and the roughness map $\beta \in \mathbb{R}^{1024\times1024}$. We unwrap the coarse mesh $\mathcal{M}$ from our hand geometry autoencoder into the UV space to obtain the corresponding coarse normal map $\mathbf{n} \in \mathbb{R}^{1024\times1024\times3}$. The predicted displacement map is applied on top of this coarse normal map to obtain the refined normal map $\hat{\mathbf{n}}$ according to Eq. 2 in the main paper.

The parametric BRDF $\mathcal{F}_{\text{pb}}$ takes as input the refined normal map $\hat{\mathbf{n}}$, the roughness map β, light $\mathcal{L} = \{L_i(\omega_i)\}_i$, and view direction $\mathbf{d}$. The physics-inspired shading features

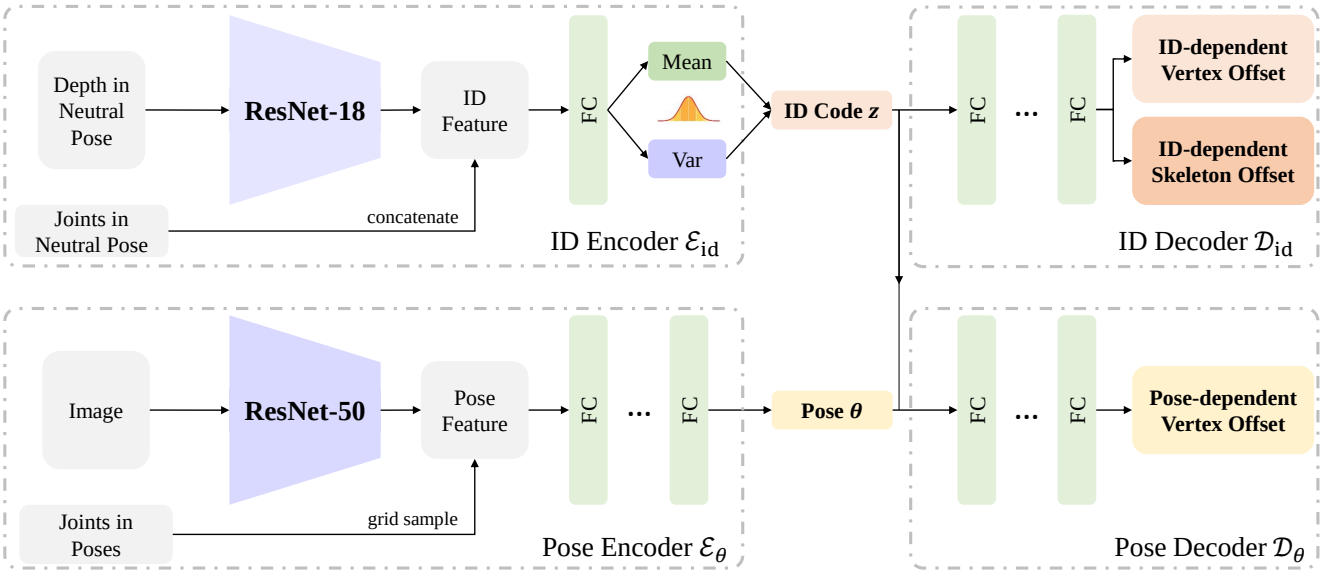

Figure 3. **The architecture of our hand geometry autoencoder.** The identity encoder $\mathcal{E}_{\text{id}}$ takes as input the depth map in the neutral pose and the coordinates of joints in the neutral pose, which predicts the mean and variance of the distribution of the identity code (*i.e.* ID code). The identity decoder $\mathcal{D}_{\text{id}}$ learns to decode the identity-dependent offset of vertices and skeletons from the ID code z. The pose encoder $\mathcal{E}_{\theta}$ directly regresses hand pose θ from the input image and the coordinates of joints. The pose decoder $\mathcal{D}_{\theta}$ learns to predict the pose-dependent offset of vertices given the pose θ and ID code z.

$\mathbf{F}_{\text{pb}} = \{\mathbf{C}^{\mathbf{d}}_{\text{pb}}, \mathbf{C}^{\mathbf{s}}_{\text{pb}}\}$ are computed accordingly. Specifically, the diffuse feature $\mathbf{C}^{\mathbf{d}}_{\text{pb}}$ is computed as:

$$\mathbf{C}^{\mathbf{d}}_{\text{pb}} = \int L_i(\omega_i) \cdot \mathbf{V}_i \cdot (\omega_i \cdot \hat{\mathbf{n}}) d\omega_i, \tag{1}$$

where $L_i(\omega_i)$ is the light intensity from the incident direction ω_i, $\mathbf{V}_i$ is the visibility given the light L_i. We adopt a mesh-based shadow map technique (similar to RelightableHands [3]), which is widely used in Computer Graphics for real-time rendering. Furthermore, the specular feature $\mathbf{C}^{s}_{\text{pb}}$ is computed as:

$$\mathbf{C}^{s}_{\text{pb}} = \int D \cdot F \cdot G \cdot L_i(\omega_i) \cdot \mathbf{V}_i \cdot (\omega_i \cdot \hat{\mathbf{n}}) d\omega_i, \tag{2}$$

$$D = \frac{\beta^4}{\pi[(\mathbf{h} \cdot \hat{\mathbf{n}})^2(\beta^4 - 1) + 1]^2}, \tag{3}$$

$$F = F_0 + (1 - F_0) \cdot 2^{[\lambda_{F1}(\mathbf{d} \cdot \mathbf{h}) + \lambda_{F2}](\mathbf{d} \cdot \mathbf{h})}, \tag{4}$$

$$G = \frac{1}{4[(\hat{\mathbf{n}} \cdot \mathbf{d})(1 - K) + K][(\hat{\mathbf{n}} \cdot \omega_i)(1 - K) + K]}, \tag{5}$$

$$\mathbf{h} = \frac{\omega_i + \mathbf{d}}{||\omega_i + \mathbf{d}||}, \quad K = \frac{(\beta + 1)^2}{8}, \tag{6}$$

where we set Fresnel coefficient $F_0 = 0.04$, $\lambda_{F1} = -5.55473$, and $\lambda_{F2} = -6.98316$, respectively.

4.3. Neural Branch

The neural branch of URHand consists of a non-linear network $\mathcal{F}_{\text{nl}}$ and a linear network $\mathcal{F}_{\text{l}}$ (*i.e.* linear lighting model). We illustrate the detailed architecture of the neural branch in Figure 4. Specifically, the non-linear network $\mathcal{F}_{\text{nl}}$ is a 7-layer CNN with channel sizes $(128, 256, 128, 128, 64, 32, 16, 4)$, which takes as input the pose- and identity-dependent joint feature $\mathbf{F}_{\theta,\text{id}}$. The linear network $\mathcal{F}_{\text{l}}$, namely the linear lighting model, consists of an encoder $\mathcal{F}_{\text{l}-\text{enc}}$ and a decoder $\mathcal{F}_{\text{l}-\text{dec}}$. The linear encoder $\mathcal{F}_{\text{enc}}$ consists of unbiased convolutional layers which takes as input the concatenated physics-inspired shading features $\{\mathbf{C}^{d}_{\text{pb}}, \mathbf{C}^{s}_{\text{pb}}\}$. The linear decoder is a 7-layer unbiased CNN with channel sizes $(128, 256, 128, 128, 64, 32, 16, 4)$. We fuse the linear features from $\mathcal{F}_{\text{l}-\text{enc}}$ and the non-linear features from $\mathcal{F}_{\text{nl}}$ as layer-wise modulation at each layer of the linear decoder $\mathcal{F}_{\text{l}-\text{dec}}$ according to Eq. 5 in the main paper. The predicted gain map $g \in \mathbb{R}^{1024 \times 1024 \times 3}$ and bias map $b \in \mathbb{R}^{1024 \times 1024}$ contributes to the final rendering according to Eq. 6 in the main paper.

5. Implementation of Baselines

In this section, we present implementation details of our baseline methods for comparisons in the main paper. Specifically, we introduce our modifications to RelightableHands [3] and Handy [12] in Sec. 5.1. Moreover, we introduce our implementations of all baselines for ablation stud-

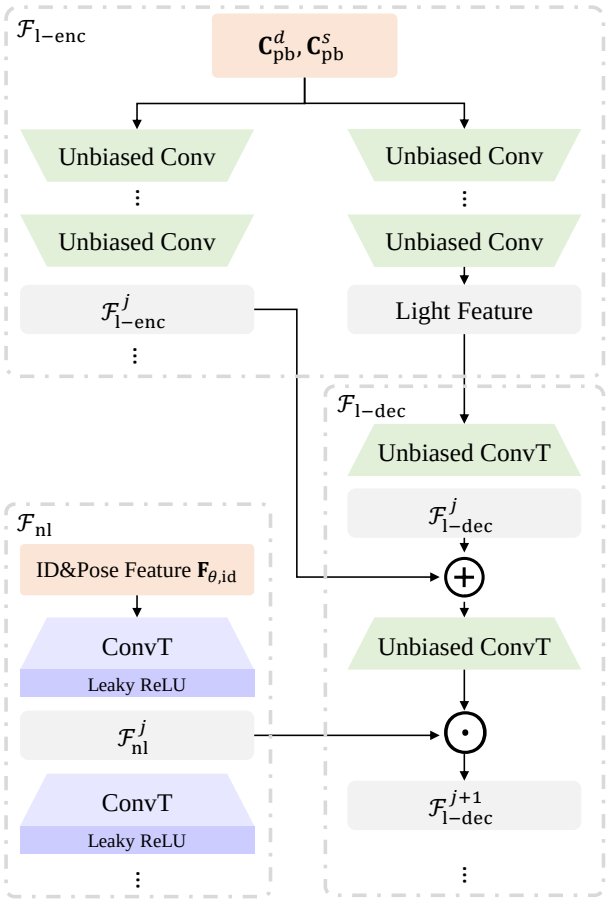

Figure 4. **The architecture of the neural branch of URHand.** The neural branch of our model consists of a non-linear network $\mathcal{F}_{\text{nl}}$ and a linear network $\mathcal{F}_{\text{l}}$ (*i.e.* linear lighting model). Notably, we remove all non-linear activation and use the convolutional layer without bias in the linear network so that the linearity of the output is explicitly kept w.r.t. the input physics-inspired shading feature $\{\mathbf{C}^d_{\text{pb}}, \mathbf{C}^s_{\text{pb}}\}$. This figure also illustrates how Eq. 5 in the main paper is implemented within our network.

ies in Sec. 5.2.

5.1. Methods for Main Comparisons

RelightableHands [3] is originally proposed for per-identity relightable appearance reconstruction tailored with volumetric representation [8]. For a fair comparison, we re-implement it with our mesh-based representation. Specifically, we leverage a U-Net as the texture decoder $\mathcal{A}_{\text{tex}}$ that takes as input the UV-aligned view direction, light direction, and visibility. The pose parameter is tiled into a UV-aligned feature map and concatenated with the bottleneck representation of the U-Net. This texture decoder $\mathcal{A}_{\text{tex}}$ predicts texture map $\mathbf{T} \in \mathbb{R}^{1024\times1024\times3}$ and shadow map $\mathbf{S} \in \mathbb{R}^{1024\times1024}$ in the UV space. The final texture $\mathbf{C}$ for rendering is obtained as:

$$\mathbf{C} = \sigma(\mathbf{S})(\text{ReLU}(\lambda_s \mathbf{T}) + \lambda_b), \tag{7}$$

where $\sigma(\cdot)$ is the sigmoid function, $\text{ReLU}(x) = \max(0, x)$, λ_s is a scale factor, and λ_b is a bias parameter. In our experiments, we set $\lambda_s = 25$, and $\lambda_s = 100$, respectively.

Handy [12] leverages the parametric hand model [14] as the shape representation and StyleGAN3 [4] as the texture model. As the shape and latent code regressor are not publicly available, we cannot infer the latent code w or shape parameters from input images as in the original paper. Instead, we fit hand shape parameters using our multiview fully lit frames on the training segments. Then, we do the StyleGAN inversion following [13]. Specifically, we randomly initialize the latent code w and optimize it given the reconstruction loss between the ground truth fully lit images and the images rendered with the current predicted texture map. In our experiments, we optimize 50,000 iterations for each identity. We use the Adam [6] optimizer with the initial learning rate as 1×10^{-3}. Once the inversion is done, we take the latent code w and feed it into the pretrained texture model of Handy to get the unwrapped texture map. We treat the unwrapped texture map as the albedo map for physically based relighting evaluation.

5.2. Baselines for Ablation Studies

Non-linear model is based on our full model but we add LeakyReLU function to all layers in the original linear network $\mathcal{F}_{\text{l}}$ which breaks the linearity.

Linear consistency model is based on the aforementioned non-linear model. We additionally constrain the linearity of this non-linear network by applying linearity consistency loss during training. Specifically, for every n iterations, we augment two physics-inspired shading features with two random scalars, *i.e.* $a_1\mathbf{F}^1_{\text{pb}} + a_2\mathbf{F}^2_{\text{pb}}$, where $a_1, a_2 \in (0, 1)$. The linearity consistency loss is defined as:

$$\mathcal{L}_{\text{lc}} = ||a_1\mathcal{F}_{\text{l}}(\mathbf{F}^1_{\text{pb}}) + a_2\mathcal{F}_{\text{l}}(\mathbf{F}^2_{\text{pb}}) - \mathcal{F}_{\text{l}}(a_1\mathbf{F}^1_{\text{pb}} + a_2\mathbf{F}^2_{\text{pb}})||_2 \tag{8}$$

MLP-based linear model [18] is a variant of the linear lighting model with no spatially varying lighting feature. We replace the encoder of linear network $\mathcal{F}_{\text{l-enc}}$ as a one-layer MLP without bias. It takes as input the environment map with a resolution of $3 \times 16 \times 32$, and predicts the lighting feature. Then we reshape the prediction into a UV-aligned feature map with a resolution of $128 \times 16 \times 16$ and feed into the decoder of linear network to predict the final gain and bias map for neural rendering.

Phong based model is implemented by replacing our physics-inspired shading feature $\mathbf{F}_{\text{pb}} = \{\mathbf{C}^d_{\text{pb}}, \mathbf{C}^s_{\text{pb}}\}$ with simple diffuse and specular feature from the Phong reflectance model. This neural relighting model is similar to [1, 3] with no learnable material parameter.

w/o Specular is the baseline where we dropout the specular feature $\mathbf{C}^s_{\text{pb}}$ during training.
w/o Visibility is the baseline where we do not incorporate visibility $\mathbf{V}_i$ when compute the physics-inspired shading feature in Eq. 1 and Eq. 2.
w/o Refiner is the baseline where we only use the normal map $\mathbf{n}$ from the coarse geometry without further refinement during training.
w/o $\mathcal{L}_{\text{GAN}}$ is the baseline trained with the reconstruction loss $\mathcal{L}_{\text{img}}$ and L1 regularization $\mathcal{L}_{\text{reg}}$ only.
w/o Light-aware $\mathcal{L}_{\text{GAN}}$ is the baseline trained with the vanilla adversarial loss without conditional discriminator. Specifically, the adversarial loss of Eq. 8 in the main paper degrades to $\mathcal{L}_{\text{GAN}} = \log \mathcal{F}_{\text{D}}(I) + \log[1 - \mathcal{F}_{\text{D}}(\hat{I})]$, where I is the ground truth and $\hat{I}$ is the rendered image.
w/o L1 Reg is the baseline trained with the reconstruction loss $\mathcal{L}_{\text{img}}$ and lighting-aware adversarial loss $\mathcal{L}_{\text{GAN}}$ only.

References

[1] Sai Bi, Stephen Lombardi, Shunsuke Saito, Tomas Simon, Shih-En Wei, Kevyn Mcphail, Ravi Ramamoorthi, Yaser Sheikh, and Jason Saragih. Deep relightable appearance models for animatable faces. *ACM Transactions on Graphics (TOG)*, 40(4):1–15, 2021. 1, 4

[2] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *Acm Siggraph*, pages 1–7. vol. 2012, 2012. 2

[3] Shun Iwase, Shunsuke Saito, Tomas Simon, Stephen Lombardi, Bagautdinov Timur, Rohan Joshi, Fabian Prada, Takaaki Shiratori, Yaser Sheikh, and Jason Saragih. Relightablehands: Efficient neural relighting of articulated hand models. In *CVPR*, 2023. 1, 3, 4

[4] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021. 4

[5] Korrawe Karunratanakul, Sergey Prokudin, Otmar Hilliges, and Siyu Tang. Harp: Personalized hand reconstruction from a monocular rgb video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12802–12813, 2023. 1

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[7] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *WACV*, 2022. 1

[8] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021. 4

[9] Gyeongsik Moon, Takaaki Shiratori, and Kyoung Mu Lee. Deephandmesh: A weakly-supervised deep encoder-decoder framework for high-fidelity hand mesh modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 440–455. Springer, 2020. 2

[10] Gyeongsik Moon, Shunsuke Saito, Weipeng Xu, Rohan Joshi, Julia Buffalini, Harley Bellan, Nicholas Rosen, Jesse Richardson, Mallorie Mize, Philippe De Bree, et al. A dataset of relighted 3d interacting hands. *arXiv preprint arXiv:2310.17768*, 2023. 1

[11] Gyeongsik Moon, Weipeng Xu, Rohan Joshi, Chenglei Wu, and Takaaki Shiratori. Authentic hand avatar from a phone scan via universal hand model. In *CVPR*, 2024. 1, 2

[12] Rolandos Alexandros Potamias, Stylianos Ploumpis, Stylianos Moschoglou, Vasileios Triantafyllou, and Stefanos Zafeiriou. Handy: Towards a high fidelity 3d hand shape and appearance model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4670–4680, 2023. 3, 4

[13] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021. 4

[14] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG)*, 36(6):1–17, 2017. 4

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2

[16] Kripasindhu Sarkar, Marcel C. Buehler, Gengyan Li, Daoye Wang, Delio Vicini, Jérémy Riviere, Yinda Zhang, Sergio Orts-Escolano, Paulo Gotardo, Thabo Beeler, and Abhimitra Meka. Litnerf: Intrinsic radiance decomposition for high-quality view synthesis and relighting of faces. In *ACM SIGGRAPH Asia 2023*, 2023. 1

[17] Yingyan Xu, Gaspard Zoss, Prashanth Chandran, Markus Gross, Derek Bradley, and Paulo Gotardo. Renerf: Relightable neural radiance fields with nearfield lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22581–22591, 2023. 1

[18] Haotian Yang, Mingwu Zheng, Wanquan Feng, Haibin Huang, Yu-Kun Lai, Pengfei Wan, Zhongyuan Wang, and Chongyang Ma. Towards practical capture of high-fidelity relightable avatars. In *SIGGRAPH Asia 2023 Conference Proceedings*, 2023. 4