BiTT: Bi-directional Texture Reconstruction of Interacting Two Hands from a Single Image

Supplementary Material

A. Datasets

This section describes details of the datasets we used in the experiments. InterHand2.6M [S5] was mainly used and RGB2Hands [S6] used as an additional dataset. For all datasets, the images are resized to 256 × 256 pixels.

InterHand2.6M [S5]. InterHand2.6M is constructed by capturing sequential frames from multi-view videos of interacting two hands. It has 80 cameras for capturing subjects, which are 19 males and 7 females, in total 26 unique subjects. We selected camera viewpoints to ensure a diverse range of perspectives for the experiment. We used all 26 identities and randomly selected the poses along 17 ranges of motion (ROM) to demonstrate the fidelity of our model in various environments.

RGB2Hands [S6]. We show the results of BiTT using the RGB2Hands dataset. Along with 4 different identities in the RGB2Hands dataset, we select a random image for training and select random other 40 images of different poses of the same identity. Note that RGB2Hands has low-resolution images and both hands generally show the same side of the hand (Tab. 3 in the main paper), thus it is less suited to demonstrate our method using the texture symmetry. The performance gain obtained is relatively less significant compared to that of InterHand2.6M.

B. Baselines

S2Hand [S7]. We expanded the functionality of S2Hand, originally designed for single hand reconstruction, to accommodate both hands. We first increase the feature dimensions of the encoder to double times and use two separate texture regression layers for each hand. Informing symmetric information of both hand, the same mean texture color for each hand are initialized. We used ground truth hand mesh and pose for a fair comparison.

HTML [S8]. We utilize the principal components of the shadow-free version of the left and right hand. We used an HTML network in the coarse stage to estimate each hand texture vector. The texture vector is then multiplied by the corresponding principal components and generates a full hand UV map. For training the HTML network, we use the pixel-wise L1 reconstruction loss.

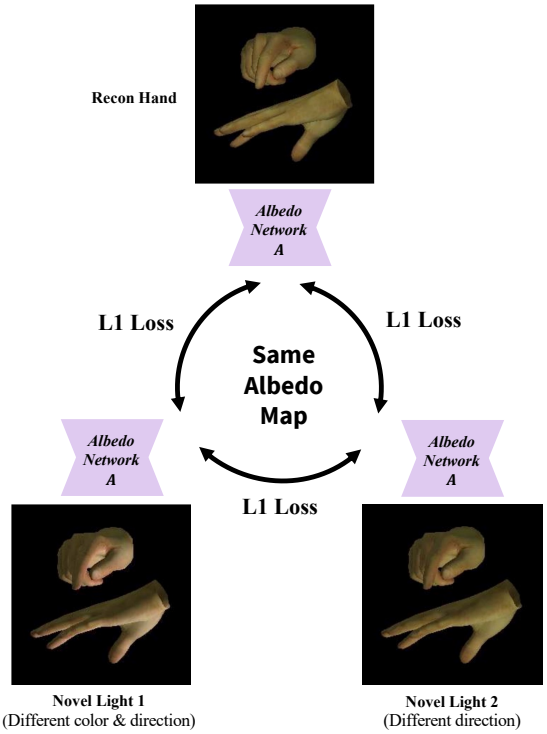


Figure 9. This figure shows the illustration of the albedo consistency loss.

HARP [S9]. HARP optimizes hand texture with visible pixel values from a monocular video. A monocular video including 4 frames of a hand was used to optimize HARP. InterHand2.6M [S5] has more than 300 points of the light source, showing a low rate of shadows. Thus, we remove the shadow rendering part in HARP and optimize the pixel values from the input sequence frames.

C. Implementation Details

In this section, we describe a detailed implementation architecture. We used PyTorch for implementation. Our work is based on HTML [S8] UV texture map template, where the UV map has the size of 1024 × 1024.

Albedo Network. The albedo network directly follows the U-Net structure [S1] where the input and output are an image. We use the LeakyReLU [S2] activation function in the encoding layer and the ReLU activation function in the decoding layers. In the albedo network, the encoding

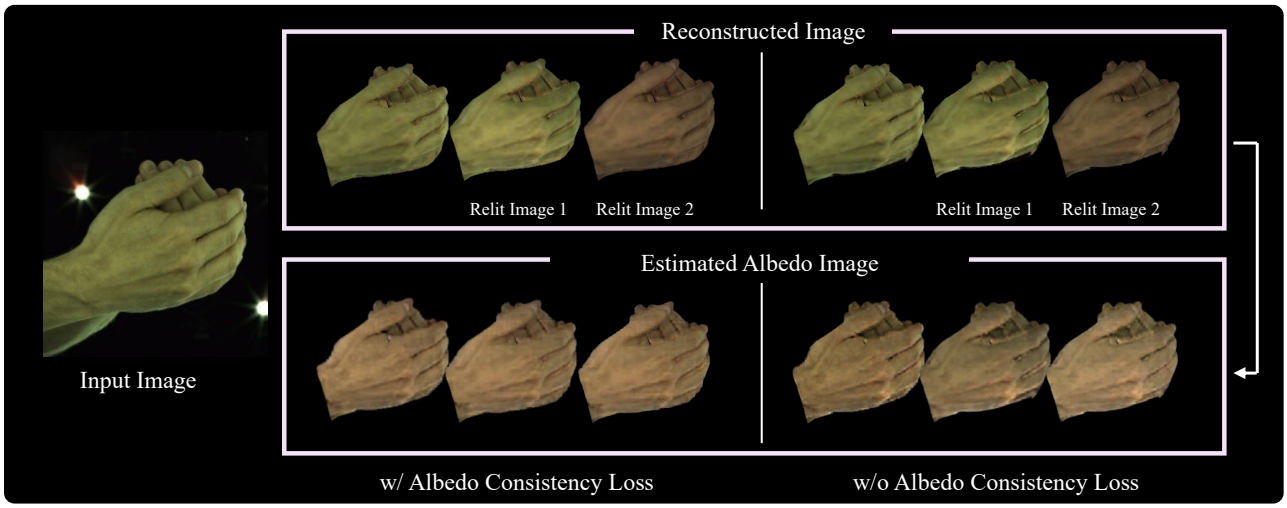


Figure 10. This figure shows the effectiveness of albedo consistency loss. In the process of reconstructing a hand from a given image and subsequently rendering it under two distinct lighting conditions, we generate three reconstructed and relit hand images. With three reconstructed/relightened hands, we estimate the albedo image for each. From the depicted images, we can find that utilizing albedo consistency loss can steadily estimate the albedo image, even when subjected to varying lighting conditions.

layer begins with input channels, which are progressively increased to 64, 128, 256, and 512 channels. After the encoding layers, the decoding layers reduce the feature dimension to 1024, 512, 256, and 128 channels, respectively. The reduction is achieved by concatenating the encoded features to the decoded features, while doubling the channel dimension. We upsample the feature by a scale factor of 2 with the nearest mode in each decoding layer.

Light Network. Light network uses an encoder network that takes an input image. The output of the light network which is denoted as L, is a 12-dimensional vector. This vector consists of different components: L_{color}, L_{diff}, L_{spec}, and $L_{direction}$, each of which is a 3-dimensional vector. The values for L_{color}, L_{diff}, and L_{spec} are adjusted to between 0.2 and 1. We used ReLU for the activation function.

The light network L, starting from the input image, increases the feature dimension to 32, 64, 128, 256. Tanh is applied to the final activation function.

HTML Network. HTML [S8] network also uses an encoder network similar to the light network. The HTML network is used at the coarse stage to estimate HTML vectors for both hands. The HTML network output is a vector of $H(I) \in \mathcal{R}^{202}$, where I is an input image and H is an HTML network. Among the 202-dimensions, the first 101 features represent the left hand HTML vector h_l, while the last 101 features represent the right hand HTML vector h_r. Each vector is then multiplied to the corresponding principal components to generate a full UV map. Similar to the light network, we use the ReLU activation function for each encoding layer.

From the input image, the HTML network H encodes the feature increasing the dimesion to 64, 128, 256, 512 and 202. Tanh is applied to the final activation function.

Bi-directional Texture Reconstructor (BTR). Bi-directional Texture reconstructor (BTR) is based on the ResNet [S3] structure. In each layer, it has 2 resblocks. After the 2 sequence res blocks, we downsample the feature size into half. After each resblock, the feature dimension is increased to 16, 64, 128, and 256. The encoding network for each hand shares the same parameter weights.

The decoding part consists of a bidirectional decoding block. We have a total 3 decoding blocks with input feature dimensions of 48, 192, 384, and 768 (3 $\times$ (encoding channels)). We finally use the activation function sigmoid to ensure texture pixel colors between 0 and 1, preventing odd colors when rendered to 2D images.

D. Albedo Consistency Loss

The albedo consistency loss aims to ensure the consistency of estimated albedo maps across images rendered under different lighting conditions. Fig. 9 shows an illustration of the albedo consistency loss concept. In this work, we have reconstructed the hand image and two other images that have been relit. Albedo network A estimates the albedo map of each image. In our experiment, we constructed two different novel light conditions. Novel light 1 is characterized by a light source positioned at the bottom and has

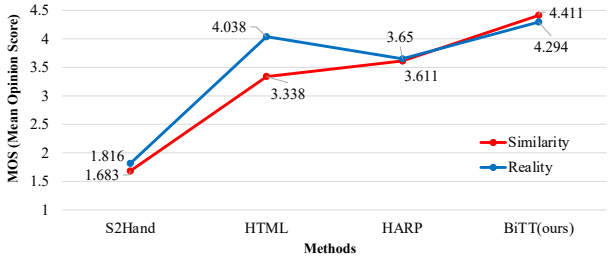
half the brightness of white light. On the other hand, novel light 2 differs from novel light 1 in terms of color, which is the same as the reconstructed light color. By applying the albedo consistency loss, we ensure that the albedo maps obtained from these three different light conditions remain consistent with each other. This helps to maintain the accurate representation of the object's reflectance properties regardless of the lighting variations.

Fig. 10 illustrates the efficacy of the albedo consistency loss. It is shown that the albedo consistency loss allows a more precise and consistent estimation of the albedo image compared to not using it.

Figure 11. Qualitative results of BiTT on Re:InterHand [S5] dataset. We present a relightened image, and a novel pose of hands in the same lighting environment of input image. The identical novel pose images effectively show the differences of estimated lighting condition.

E. Subjective Tests

We conducted a subjective test involving 27 users to answer 72 questions, earning a total of 1,944 responses. Each question presented a randomly generated image of each method with its corresponding GT image to evaluate the similarity and texture realism. Responses were collected in a 5-point discrete scale, ranging from "bad" (1) to "excellent" (5).



	Correlation value of L1 to MOS	Correlation value of PSNR to MOS	Correlation value of SSIM to MOS
Similarity	-0.119	0.177	0.303
Realism	-0.202	0.151	0.358

Figure 12. The subjective test results.

Our method achieved the best scores compared to baselines. S2Hand [S7] showed lower scores due to blurred texture as a limitation of vertex rendering. HTML [S8] showed strength in realism, while showing weakness in similarity. HARP [S9] concluded with scores ranging between "fair" (3) and "good" (4), showing difficulty in the few shot learning. In Fig. 12, we report subjective results with Pearson correlation value of the L1, PSNR, and SSIM metrics.

F. More Qualitative Results

In this section, we present additional qualitative results of our reconstructed images of two hands, in Fig. 14. Even if the ground truth mesh does not perfectly align with the input hand image, our proposed method BiTT can present realistic hand textures. As a hand texture parametric model (HTML) [S8] is robust to the noise and lacks its ability to represent background color, BiTT rendered image does not include background pixel colors on the hand texture due to minor geometric misalignments. Each column, from left to right, represents the input image, reconstructed image, novel pose, novel viewpoint with ground truth images, and relightened hands.

Results on Re:InterHand [S5] Dataset. Re:InterHand [S5] used the RelightableHands [S4] method and generated a large dataset of two interacting hands relightened in several different environments. RelightableHands presents the neural rendering approach to create relightable hand

avatars. This process requires a specialized capturing studio having numerous cameras and light sources. Note our method only requires a single image input, end users in AR/VR systems can easily generate their personalized two-hand avatar realistically.

As Re:InterHand has been passed through the learning process, the generated images exhibit less clear details of hands, such as wrinkles, hairs, and veins, compared to InterHand2.6M [S5] where the images are taken directly from cameras. We present several qualitative results of BiTT in the Re:InterHand dataset at Fig. 11. The figure demonstrates that BiTT accurately reconstructs a relightable two-hand avatar from an input image, even in diverse settings of environments.

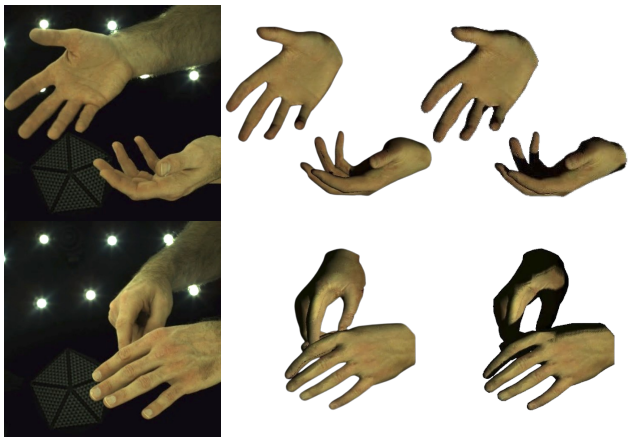


Figure 13. Visualization of the results with shadow rendering applied in BiTT method.

Applying Shadow Construction. Our method relies on mesh-based rendering, which makes it easily compatible with traditional concepts in computer graphics. We have applied differentiable self-shadow rendering directly from the methodology presented in [S9]. Given that our method involves two hands, it faces significant challenges in occlusion from each hand, along with self-occlusion toward the light source. Despite these complexities, it effectively captures the shadow appearance occured by self-occlusion and interhand-occlusion. The results are illustrated in Fig. 13. Since InterHand2.6M [S5] was captured in an environment with numerous light sources, the application of shadow rendering tends to deviate from the input image. Nevertheless, in a scene illuminated by a single point light, it can enhance the realism of the rendered hand.

References

[S1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI, 2015. 1

[S2] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In ICML, 2013. 1

[S3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 2

[S4] Shun Iwase, Shunsuke Saito, Tomas Simon, Stephen Lombardi, Timur Bagautdinov, Rohan Joshi, Fabian Prada, Takaaki Shiratori, Yaser Sheikh, Jason Saragih. RelightableHands: Efficient Neural Relighting of Articulated Hand Models. In CVPR, 2023. 3

[S5] Gyeongsik Moon, Shunsuke Saito, Weipeng Xu, Rohan Joshi, Julia Buffalini, Harley Bellan, Nicholas Rosen, Jesse Richardson, Mallorie Mize, Philippe de Bree, Tomas Simon, Bo Peng, Shubham Garg, Kevyn McPhail, Takaaki Shiratori. A Dataset of Relighted 3D Interacting Hands. In NIPS, 2023. 3

[S5] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In ECCV, 2020 1, 4, 5

[S6] Jiayi Wang, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A.Otaduy, Dan Casas, and Christian Theobalt. Rgb2hands: Real-time tracking of 3d hand interactions from monocular rgb video. ACM TOG, 2020 1

[S7] Yujin Chen, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. Model-based 3d hand reconstruction via self-supervised learning. In CVPR, 2021. 1, 3

[S8] Neng Qian, Jiayi Wang, Franziska Mueller, Florian Bernard, Vladislav Golyanik, and Christian Theobalt. HTML: A Parametric Hand Texture Model for 3D Hand Reconstruction and Personalization. In ECCV, 2020 1, 2, 3

[S9] Korrawe Karunratanakul, Sergey Prokudin, Otmar Hilliges, and Siyu Tang. Harp: Personalized hand reconstruction from a monocular rgb video. In CVPR, 2023. 1, 3, 4

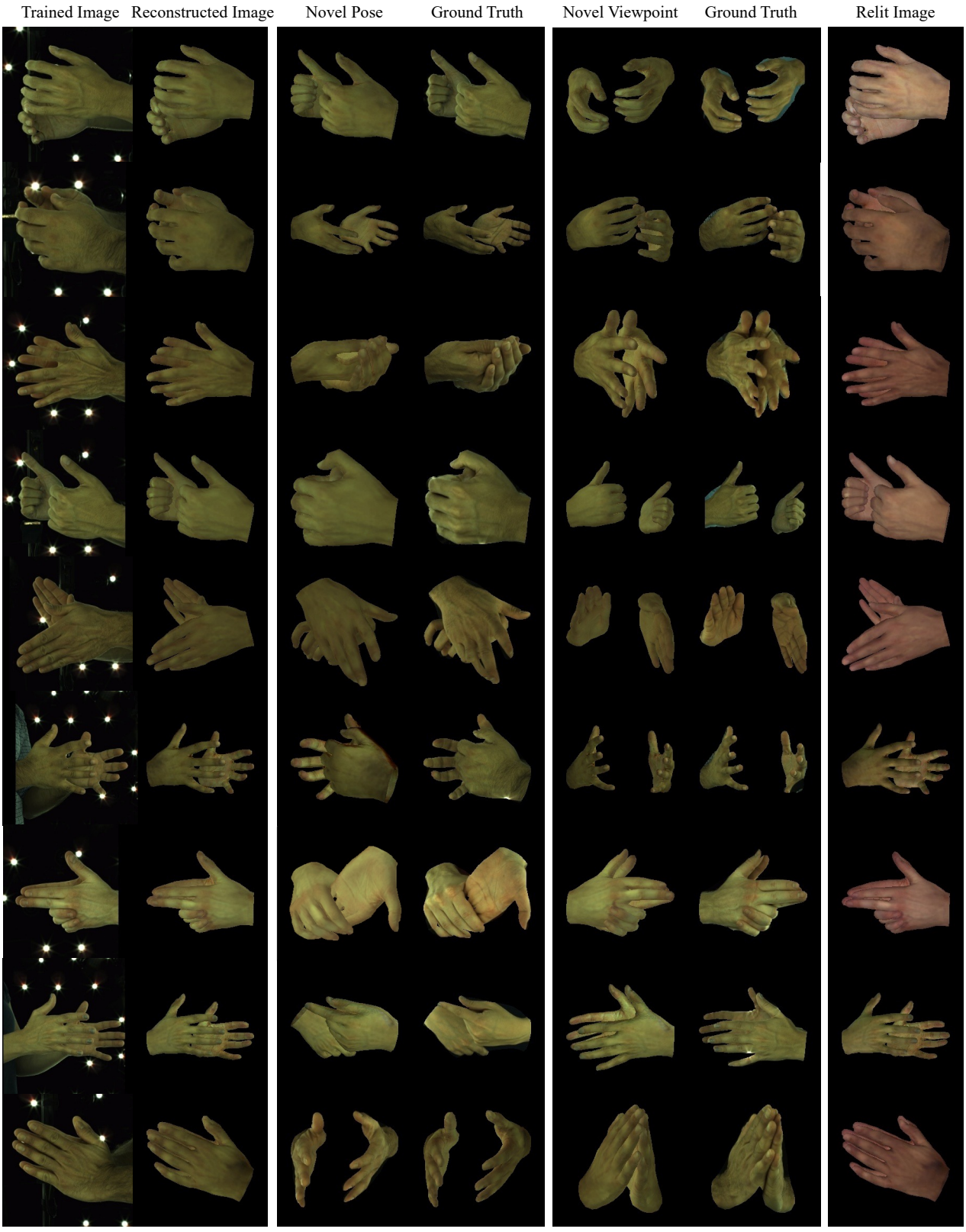


Figure 14. More qualitative results of reconstructed hands rendered on the novel pose, novel viewpoint, and relightened image in the dataset InterHand2.6M [S5].