Supplementary Material for

Recovering 3D Hand Mesh Sequence from a Single Blurry Image: A New Dataset and Temporal Unfolding

In this supplementary material, we provide more various visual results, discussions, and other details that could not be included in the main manuscript due to the lack of space. The contents are summarized below:

- S1. Visualization in video format
- S2. Statistics on the BlurHand dataset
- S3. Results from various deblurring methods
- S4. Training details
- S5. Additional qualitative results
- S6. Discussions

S1. Visualization in video format

In the supplementary videos (**`BlurHandNet.mp4`**), we visualize the recovered 3D hand mesh sequence as video clips. In the video, we note that the left part is the input blurry hand image, and the right part is the results from our BlurHandNet. We further note that we adopt linear interpolation to smooth the motion, and motion order is determined in the order of $\mathbf{V}_{\text{E1}}$, $\mathbf{V}_{\text{M}}$, and $\mathbf{V}_{\text{E2}}$.

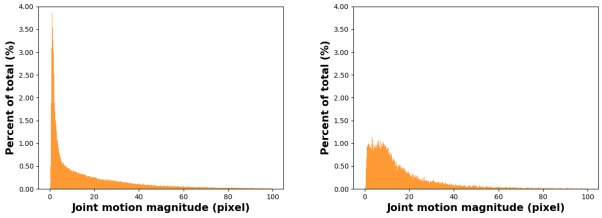
The video shows that our BlurHandNet outputs robust 3D hand meshes from challenging blurry hand. Moreover, our BlurHandNet successfully estimates the motion in the blurry hand by performing temporal unfolding through the proposed Unfolder. Compared to the previous methods, which output the mesh in the static scene, our BlurHandNet gives more accurate and comprehensive results from blurry inputs, including motion information.

S2. Statistics on the BlurHand dataset

In Table S1, we report the detailed number of training samples. We note that the right and left hands are evenly distributed in the BlurHand. In Figure S1, we further report additional measurements, namely joint motion magnitude, to present the statistics on the blur strength of our BlurHand. In detail, we first prepare five sequential sharp frames, which construct a single blurry frame in our BlurHand. Then we calculate the 2D joint distance between two adjacent sharp frames using the GT joint positions. Finally, we add all distances for each joint, which we denote as joint motion magnitude. We note that the large joint motion magnitude means a strong blur exists in hand. Our BlurHand contains samples with various joint motion magnitude in both the train and test sets.

Split	BlurHand	
	Left hand	Right hand
Train	85,380	83,659
Test	17,143	16,914

Table S1. **Number of blurry hand image in our BlurHand.** We count both if the image contains both hands.



(a) Train set (b) Test set

Figure S1. **Statistics on blur strength of the presented BlurHand.** On average, the joint motion magnitude from the train and test set are 16.9 and 17.8, respectively.

S3. Results from various deblurring methods

In Tables 2 and 3 in our main manuscript, we compared our BlurHandNet with the combination of state-of-the-art 3D hand mesh estimation methods [4–6] and off-the-shelf deblurring method [1]. In Table S2, we additionally compare the results from another widely used deblurring method, DeepDeblur [7], as the final mesh estimation results might be dependent on the performance of deblurring methods. Please note that NAFNet [1] is a deblurring method that we used in the main manuscript. Our BlurHandNet still outperforms the case when we use DeepDeblur [7] as the deblurring method. The results again demonstrate that utilizing temporal information is useful rather than simply adopting deblurring methods.

Deblurring methods	Deblur	Unfolder	KTFormer	MPJPE		
				initial	middle	final
DeepDeblur [7]	✓	✗	✗	-	18.03	-
	✓	✓	✓	19.24	18.04	19.27
NAFNet [1]	✓	✗	✗	-	17.28	-
	✓	✓	✓	18.95	17.28	19.10
None	✗	✓	✓	**18.08**	**16.80**	**18.21**

Table S2. **Comparison results on various deblurring methods [1,7].** Instead of adopting deblurring methods, our BlurHandNet (last row) utilizes temporal information from a blurry image.

S4. Training details

BlurHandNet. We use Adam optimizer [2] with a batch size of 48 for training our BlurHandNet. The initial learning rate is set to 1×10^{-4} and reduced by a factor of 10 at the $10th$ and $12th$ epochs. The proposed network is trained for 13 epochs and takes about 5.7 hours using two NVIDIA 2080 Ti GPUs. All other details will be available in our codes.

State-of-the-art models. For training the state-of-the-art 3D hand mesh estimation networks [4–6] and deblurring networks [1] on BlurHand, we follow their official training instruction. In addition, we employ the authors' official pretrained weight in training deblurring methods [1] for easier optimization.

S5. Additional qualitative results

Effectiveness of the BlurHand. In Figure S3, we provide additional qualitative results on YT-3D [3]. We note that training the model on our BlurHand (column (e) in the Figure S3) is significantly helpful in dealing with the in-the-wild blurry hand images compared to the cases using sharp images and deblurred images (column (c) and (d) in the Figure S3). The results justify the necessity of our BlurHand when handling the blurry hand.

Visual comparison on BlurHand. In Figure S4, we present additional comparison results on BlurHand. Compared to the previous state-of-the-art methods [4, 5], our BlurHandNet reconstructs more accurate 3D hand meshes by exploiting temporal information.

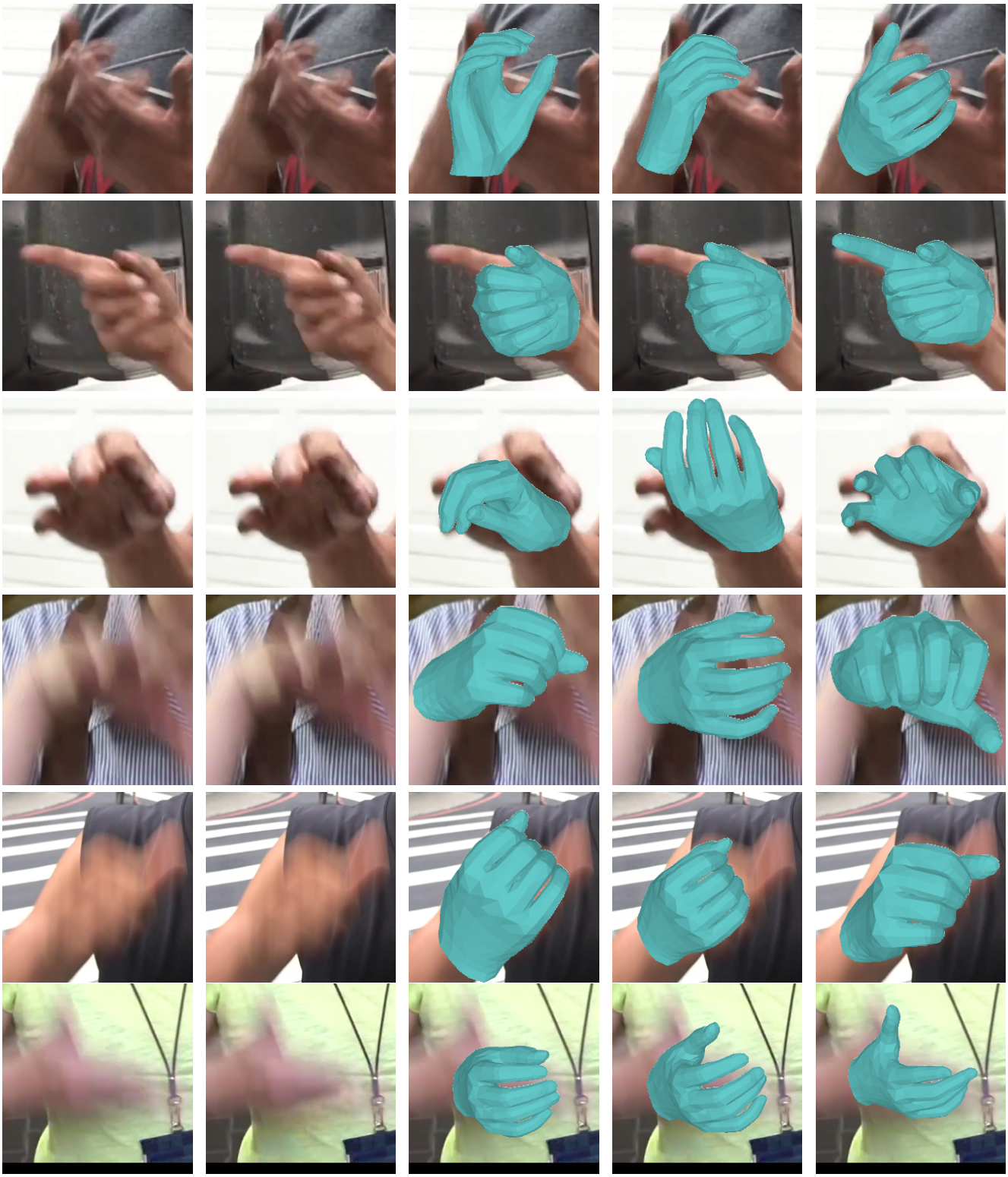
S6. Discussion

Limitations and future works. While various types of image degradations are prevalent in real-world hand images, *e.g.*, low-resolution, noise, and low illumination, we especially focus on hands with blur artifacts. Figure S2 shows that BlurHandNet produces not robust results when the input image is low-resolution with blur, which should be considered in our future works.

Societal impacts. Our BlurHand and BlurHandNet suggest a new and necessary research direction toward real-world applications, the robust 3D hand mesh estimation from blurry images. In particular, our method can be useful for AR/VR as people often move hands fast, which causes motion blur of hands.

(a) Input image (blurry and low-resolution) (b) BlurHandNet (Ours)

Figure S2. **Failure cases.** Our BlurHandNet produces less accurate results when inputs contain multiple complex degradations.



(a) Blurry image (b) Deblurred image (c) IH2.6M + YT3D (d) BH+**D** + YT3D (e) BH + YT3D

Figure S3. **Effectiveness of the presented BlurHand.** The captions below figures describe training sets used to train 3D hand mesh estimation networks. The notation **D** represents that the network is trained on deblurred BH and tested on (b).

Figure S4. **Visual comparison of the proposed BlurHandNet and state-of-the-art 3D hand mesh estimation methods [4, 5] on BlurHand.** We note that all the methods are trained on BlurHand.

References

[1] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *arXiv preprint arXiv:2204.04676*, 2022. 1, 2

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

[3] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020. 2

[4] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 1, 2, 4

[5] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Accurate 3D hand pose estimation for whole-body 3D human mesh estimation. In *CVPRW*, 2022. 1, 2, 4

[6] Gyeongsik Moon and Kyoung Mu Lee. I2L-MeshNet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image. In *ECCV*, 2020. 1, 2

[7] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 1, 2