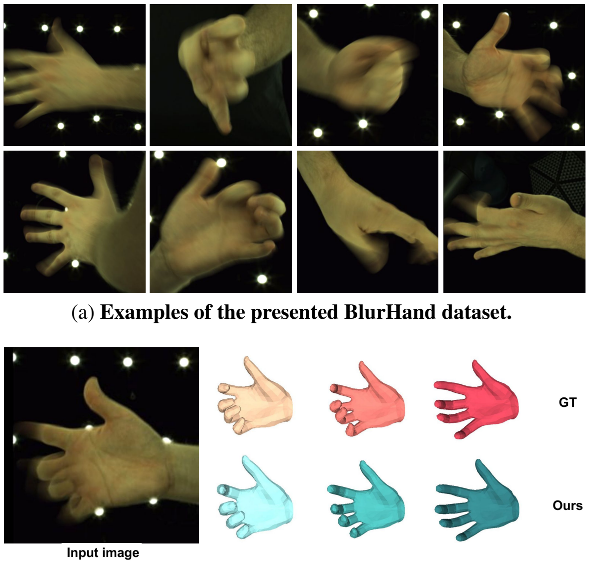
Recovering 3D Hand Mesh Sequence from a Single Blurry Image: A New Dataset and Temporal Unfolding

Yeonguk Oh[1*] JoonKyu Park[1*] Jaeha Kim[1*] Gyeongsik Moon[3] Kyoung Mu Lee[1,2]
[1]Dept. of ECE&ASRI, [2]IPAI, Seoul National University, Korea
[3]Meta Reality Labs Research
{namepllet, jkpark0825}@snu.ac.kr, jhkim97s2@gmail.com, mks0601@meta.com, kyoungmu@snu.ac.kr

Abstract

Hands, one of the most dynamic parts of our body, suffer from blur due to their active movements. However, previous 3D hand mesh recovery methods have mainly focused on sharp hand images rather than considering blur due to the absence of datasets providing blurry hand images. We first present a novel dataset BlurHand, which contains blurry hand images with 3D groundtruths. The BlurHand is constructed by synthesizing motion blur from sequential sharp hand images, imitating realistic and natural motion blurs. In addition to the new dataset, we propose BlurHandNet, a baseline network for accurate 3D hand mesh recovery from a blurry hand image. Our BlurHandNet unfolds a blurry input image to a 3D hand mesh sequence to utilize temporal information in the blurry input image, while previous works output a static single hand mesh. We demonstrate the usefulness of BlurHand for the 3D hand mesh recovery from blurry images in our experiments. The proposed BlurHandNet produces much more robust results on blurry images while generalizing well to in-the-wild images. The training codes and BlurHand dataset are available at https://github.com/JaehaKim97/BlurHand_RELEASE.


(a) **Examples of the presented BlurHand dataset.**

(b) **Illustration of the temporal unfolding.**

Figure 1. **Proposed BlurHand dataset and BlurHandNet.** (a) We present a novel BlurHand dataset, providing natural blurry hand images with accurate 3D annotations. (b) While most previous methods produce a single 3D hand mesh from a sharp image, our BlurHandNet unfolds the blurry hand image into three sequential hand meshes.

1. Introduction

Since hand images frequently contain blur when hands are moving, developing a blur-robust 3D hand mesh estimation framework is necessary. As blur makes the boundary unclear and hard to recognize, it significantly degrades the performance of 3D hand mesh estimation and makes the task challenging. Despite promising results of 3D hand mesh estimation from a single sharp image [5,13,16,17,22], research on blurry hands is barely conducted.

A primary reason for such lack of consideration is the absence of datasets that consist of blurry hand images with accurate 3D groundtruth (GT). Capturing blurry hand datasets is greatly challenging. The standard way of capturing markerless 3D hand datasets [8,23,50] consists of two stages: 1) obtaining multi-view 2D grounds (*e.g.*, 2D joint coordinates and mask) manually [50] or using estimators [14, 15, 44] and 2) triangulating the multi-view 2D grounds to the 3D space. Here, manual annotations or estimators in the first stage are performed from images. Hence, they become unreliable when the input image is blurry, which results in triangulation failure in the second stage.

Contemplating these limitations, we present the BlurHand, whose examples are shown in Figure 1a. Our BlurHand, the first blurry hand dataset, is synthesized from InterHand2.6M [23], which is a widely adopted video-based

[*] Authors contributed equally.

hand dataset with accurate 3D annotations. Following state-of-the-art blur synthesis literature [25, 26, 39], we approximate the blurry images by averaging the sequence of sharp hand frames. As such technique requires high frame rates of videos, we employ a widely used video interpolation method [27] to complement the low frame rate (30 frames per second) of InterHand2.6M. We note that our synthetic blur dataset contains realistic and challenging blurry hands.

For a given blurry hand image, the most straightforward baseline is sequentially applying state-of-the-art deblurring methods [3, 29, 30, 46] on blurry images and 3D hand mesh estimation networks [21, 22, 38] on the deblurred image. However, such a simple baseline suffers from two limitations. First, since hands contain challenging blur caused by complex articulations, even state-of-the-art deblurring methods could not completely deblur the image. Therefore, the performance of the following 3D hand mesh estimation networks severely drops due to remaining blur artifacts. Second, since conventional deblurring approaches only restore the sharp images corresponding to the middle of the motion, it limits the chance to make use of temporal information, which might be useful for 3D mesh estimation. In other words, the deblurring process restricts networks from exploiting the motion information in blurry hand images.

To overcome the limitations, we propose BlurHandNet, which recovers a 3D hand mesh sequence from a single blurry image, as shown in Figure 1b. Our BlurHandNet effectively incorporates useful temporal information from the blurry hand. The main components of BlurHandNet are Unfolder and a kinematic temporal Transformer (KTFormer). Unfolder outputs hand features of three timesteps, *i.e.*, middle and both ends of the motion [12, 28, 32, 36]. The Unfolder brings benefits to our method in two aspects. First, Unfolder enables the proposed BlurHandNet to output not only 3D mesh in the middle of the motion but also 3D meshes at both ends of the motion, providing more informative results related to motion. We note that this property is especially beneficial for the hands, where the motion has high practical value in various hand-related works. For example, understanding hand motion is essential in the domain of sign language [2, 34] and hand gestures [40], where the movement itself represents meaning. Second, extracting features from multiple time steps enables the following modules to employ temporal information effectively. Since hand features in each time step are highly correlated, exploiting temporal information benefits reconstructing more accurate 3D hand mesh estimation.

To effectively incorporate temporal hand features from the Unfolder, we propose KTFormer as the following module. The KTFormer takes temporal hand features as input and leverages self-attention to enhance the temporal hand features. The KTFormer enables the proposed BlurHandNet to implicitly consider both the kinematic structure and temporal relationship between the hands in three timesteps. The KTFormer brings significant performance gain when coupled with Unfolder, demonstrating that employing temporal information plays a key role in accurate 3D hand mesh estimation from blurry hand images.

With a combination of BlurHand and BlurHandNet, we first tackle 3D hand mesh recovery from blurry hand images. We show that BlurHandNet produces robust results from blurry hands and further demonstrate that BlurHandNet generalizes well on in-the-wild blurry hand images by taking advantage of effective temporal modules and BlurHand. As this problem is barely studied, we hope our work could provide useful insights into the following works. We summarize our contributions as follows:

- We present a novel blurry hand dataset, BlurHand, which contains natural blurry hand images with accurate 3D GTs.
- We propose the BlurHandNet for accurate 3D hand mesh estimation from blurry hand images with novel temporal modules, Unfolder and KTFormer.
- We experimentally demonstrate that the proposed BlurHandNet achieves superior 3D hand mesh estimation performance on blurry hands.

2. Related works

3D hand mesh estimation. Since after the introduction of RGB-based hand benchmark datasets with accurate 3D annotations, *e.g.*, Friehand [50] and InterHand 2.6M [23], various monocular RGB-based 3D hand mesh estimation methods [5, 13, 16, 17, 21, 22, 31] have been proposed. Pose2Mesh [5] proposed a framework that reconstructs 3D mesh from the skeleton pose based on graph convolutional networks. Kulon *et al.* [13] utilized encoder-decoder architecture with a spiral operator to regress the 3D hand mesh. I2L-MeshNet [22] utilized a 1D heatmap for each mesh vertex to model the uncertainty and preserve the spatial structure. I2UV-HandNet [4] proposed UV-based 3D hand shape representation and 3D hand super-resolution module to obtain high-fidelity hand meshes. Pose2Pose [21] introduced joint features and proposed a 3D positional pose-guided 3D rotational pose prediction framework. More recently, LISA [6] captured precise hand shape and appearance while providing dense surface correspondence, allowing for easy animation of the outputs. SeqHAND [45] incorporated synthetic datasets to train a recurrent framework with temporal movement information and consistency constraints, improving general pose estimations. Meng *et al.* [20] decomposed the 3D hand pose estimation task and used the HDR framework to handle occlusion.

After the success of the attention-based mechanism, Transformer [42] has been adopted to recover more accurate

3D hand meshes. METRO [16] and MeshGraphormer [17] proposed Transformer-based architecture, which models vertex-vertex and vertex-joint interactions. Liu *et al.* [19] utilizes spatial-temporal parallel Transformer to model inter-correlation between arm and hand. HandOccNet [31] proposed a Transformer-based feature injection mechanism to robustly reconstruct 3D hand mesh when occlusions are severe. Although the above methods showed promising results for the sharp hand images, none of them carefully considered the hand with blur scenario. As the lack of an appropriate dataset is the main reason for the less consideration, we present BlurHand. Furthermore, we introduce a baseline network, BlurHandNet, which consists of a temporal unfolding module and kinematic temporal Transformer.

Restoring the motion from a single blurry image. Rather than reconstructing only a single sharp image in the middle of the motion, recent deblurring methods [1, 12, 28, 32, 47] have witnessed predicting the sequence of sharp frames from a single blurry image, which constructs the blurry input image. Such a sequence of sharp frames can provide useful temporal information. Jin *et al.* [12] proposed temporal order invariant loss to overcome the temporal order ambiguity problem. Purohit *et al.* [32] proposed an RNN-based solution without constraining the number of frames in sequence. Argaw *et al.* [1] proposed an encoder-decoder-based spatial Transformer network with regularizing terms. Unlike previous methods that proposed to restore a single sharp image, our BlurHandNet aims to recover 3D hand mesh sequences from a single blurry image.

3. BlurHand dataset

Figure 2 shows the overall pipeline for constructing our BlurHand. Our BlurHand dataset is synthesized using 30 frames per second (fps) version of InterHand2.6M [23], which contains large-scale hand videos with diverse poses. We first apply a video interpolation method [27] to increase 30 fps videos to 240 fps ones. Then, a single blurry hand image is synthesized by averaging 33 sequential frames, which are interpolated from 5 sharp sequential frames, following the conventional deblurring dataset manufacture [25,37,49]. We note that video interpolation is necessary when synthesizing blurs, as averaging frames from a low frame rate induces unnatural artifacts such as spikes and steps [24]. For each synthesized blurry image, 3D GTs of *1st*, *3rd*, and *5th* sharp frames from InterHand2.6M 30fps are assigned as 3D GTs of initial, middle, and final, respectively. In the end, the presented BlurHand consists of 121,839 training and 34,057 test samples containing single and interacting blurry hand images. During the synthesis of blurry frames, we skip the frames if two neighboring frames are not available, and further adopt camera view sampling to mitigate the redundancy of samples. We report sample statistics of the BlurHand in the supplementary materials.

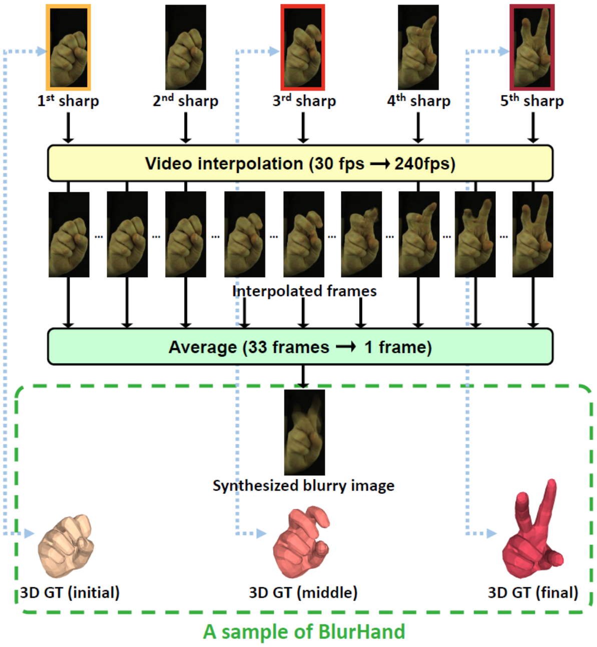


Figure 2. **Pipeline for constructing our BlurHand dataset.** We synthesize the blurry hand from five sequential sharp hand frames by adopting video interpolation and averaging them.

4. BlurHandNet

Figure 3 shows the overall architecture of our BlurHandNet. Our BlurHandNet, which consists of three modules; Unfolder, KTFormer, and Regressor, reconstructs sequential hand meshes from a single blurry hand image. We describe the details of each module in the following sections.

4.1. Unfolder

Unfolding a blurry hand image. Given a single RGB blurry hand image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, Unfolder outputs feature maps and 3D joint coordinates of the three sequential hands, *i.e.*, temporal unfolding, where each corresponds to the hand from both ends and the middle of the motion. Here, $H = 256$ and $W = 256$ denote the height and width of the input image, respectively. The temporal unfolding could extract useful temporal information from a single blurry image, and we note that effectively utilizing them is one of the core ideas of our methods. To this end, we first feed the blurry hand image $\mathbf{I}$ into ResNet50 [9], pre-trained on ImageNet [7], to extract the *blurry* hand feature map $\mathbf{F}_\text{B} \in \mathbb{R}^{H/32 \times W/32 \times C}$, where $C = 2048$ denotes the channel dimension of $\mathbf{F}_\text{B}$. Then, we predict three temporal features from a blurry hand feature $\mathbf{F}_\text{B}$ through corresponding separate decoders, as shown in Figure 3. As a result, we obtain three sequential hand features $\mathbf{F}_\text{E1}$, $\mathbf{F}_\text{E2}$, and $\mathbf{F}_\text{M}$ with dimension $\mathbb{R}^{h \times w \times c}$, where each corresponds to the hand at both ends and the middle of the motion. Here, $h = {}^{H}\!/4$, $w = {}^{W}\!/4$, and $c = 512$ denote the height, width, and channel dimension of each hand feature, respectively.

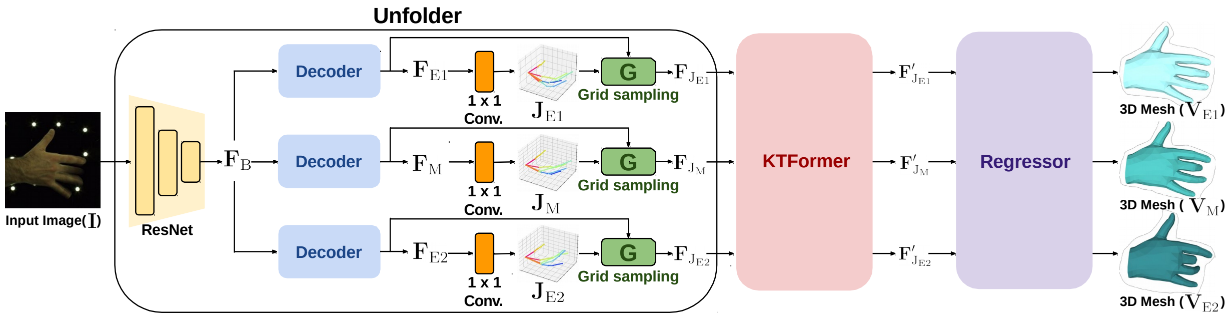

Unfolder
Input Image($\mathbf{I}$)
ResNet
F_{B}
Decoder
Decoder
Decoder
F_{E1}
F_{M}
F_{E2}
1 x 1 Conv.
1 x 1 Conv.
1 x 1 Conv.
J_{E1}
J_{M}
J_{E2}
G
Grid sampling
$\mathrm{F}_{\mathrm{J}_{\mathrm{E1}}}$
$\mathrm{F}_{\mathrm{J}_{\mathrm{M}}}$
$\mathrm{F}_{\mathrm{J}_{\mathrm{E2}}}$
KTFormer
$\mathrm{F}'_{\mathrm{J}_{\mathrm{E1}}}$
$\mathrm{F}'_{\mathrm{J}_{\mathrm{M}}}$
$\mathrm{F}'_{\mathrm{J}_{\mathrm{E2}}}$
Regressor
3D Mesh (V_{E1})
3D Mesh (V_{M})
3D Mesh (V_{E2})

Figure 3. **Overall architecture of BlurHandNet.** BlurHandNet first unfolds input image $\mathbf{I}$ into three temporal joint features $\mathbf{F}_{\mathrm{J}_{\mathrm{E1}}}$, $\mathbf{F}_{\mathrm{J}_{\mathrm{E2}}}$, and $\mathbf{F}_{\mathrm{J}_{\mathrm{M}}}$. The following kinematic temporal Transformer (KTFormer) refines each joint feature by leveraging the attentive correlation between them. Finally, Regressor produces MANO [35] parameters for each time step, resulting in temporal 3D hand meshes.

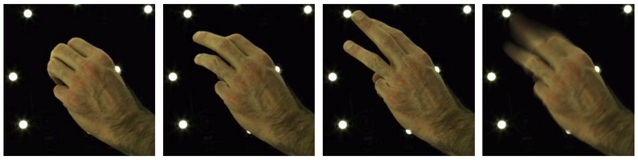

(a) (b) (c) (d)

Figure 4. **The ambiguity on temporal ordering.** Hand image sequences of the extending <(a)→(b)→(c)> and the folding <(c)→(b)→(a)> make the same result blur image (d).

Among the three sequential features, the hand feature at the middle of the motion $\mathbf{F}_{\mathrm{M}}$ can be specified as similar to the conventional deblurring approaches [25, 46]. However, we can not identify whether the hand at each end (*i.e.*, $\mathbf{F}_{\mathrm{E1}}$ or $\mathbf{F}_{\mathrm{E2}}$) is come from the initial or final location of the motion due to the temporal ambiguity [12, 28, 33, 47]. For example, suppose that we obtain the blurry hand image shown in Figure 4d. Then we can not determine whether the blurry hand image comes from the motion of extending or folding. In that regard, Unfolder outputs hand features from both ends of the motion (*i.e.*, $\mathbf{F}_{\mathrm{E1}}$ and $\mathbf{F}_{\mathrm{E2}}$) without considering temporal order. We note that exploiting the temporal information still benefits without explicitly considering the temporal order, and can further be stably optimized with the training loss introduced in Section 4.4.

Extracting temporal joint features. From produced three sequential hand features, we extract the corresponding joint features, which contain essential hand articulation information [21] that helps to recover 3D hand meshes. We first project the sequential hand features $\mathbf{F}_{\mathrm{E1}}$, $\mathbf{F}_{\mathrm{E2}}$, and $\mathbf{F}_{\mathrm{M}}$ into dJ dimensional feature through 1×1 convolution layer, and reshape them into 3D heatmaps with the dimension of $\mathbb{R}^{J\times h\times w\times d}$, where $d = 32$ is a depth discretization size and $J = 21$ is a number of hand joints. Then, we perform a soft-argmax operation [41] on each heatmap to obtain the 3D joint coordinates of three temporal hands, $\mathbf{J}_{\mathrm{E1}}$, $\mathbf{J}_{\mathrm{E2}}$, and $\mathbf{J}_{\mathrm{M}}$ with dimension of $\mathbb{R}^{J\times 3}$. Using 3D joint coordinates in each temporal hand, we perform grid sampling [10, 21] on the corresponding feature map. By doing so, we obtain temporal joint features $\mathbf{F}_{\mathrm{J}_{\mathrm{E1}}}$, $\mathbf{F}_{\mathrm{J}_{\mathrm{E2}}}$, and $\mathbf{F}_{\mathrm{J}_{\mathrm{M}}}$ with a dimension of $\mathbb{R}^{J\times c}$, which enable the following module to exploit temporal information effectively.

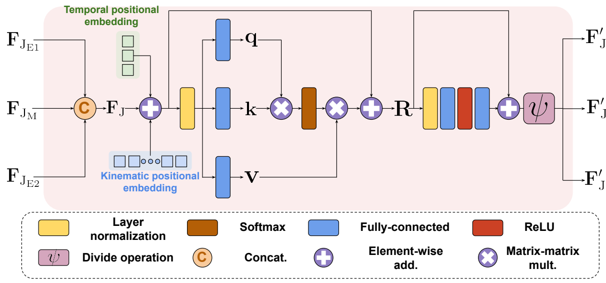

Temporal positional embedding
Kinematic positional embedding
$\mathrm{F}_{\mathrm{J}_{\mathrm{E1}}}$
$\mathrm{F}_{\mathrm{J}_{\mathrm{M}}}$
$\mathrm{F}_{\mathrm{J}_{\mathrm{E2}}}$
C
F_{J}
q
k
v
R
ψ
$\mathrm{F}'_{\mathrm{J}_{\mathrm{E1}}}$
$\mathrm{F}'_{\mathrm{J}_{\mathrm{M}}}$
$\mathrm{F}'_{\mathrm{J}_{\mathrm{E2}}}$
Layer normalization
Softmax
Fully-connected
ReLU
Divide operation
Concat.
Element-wise add.
Matrix-matrix mult.

Figure 5. **Overall architecture of KTFormer.** KTFormer refines temporal joint features $\mathbf{F}_{\mathrm{J}_{\mathrm{E1}}}$, $\mathbf{F}_{\mathrm{J}_{\mathrm{E2}}}$, and $\mathbf{F}_{\mathrm{J}_{\mathrm{M}}}$. First, kinematic and temporal positional embeddings are introduced. Then, the following self-attention mechanism refines joint features by leveraging attentive correlation between them, producing $\mathbf{F}'_{\mathrm{J}_{\mathrm{E1}}}$, $\mathbf{F}'_{\mathrm{J}_{\mathrm{E2}}}$, and $\mathbf{F}'_{\mathrm{J}_{\mathrm{M}}}$.

4.2. KTFormer

Kinematic-temporal positional embedding. The illustration of KTFormer is shown in Figure 5. KTFormer is the Transformer [42]-based module that refines the joint feature $\mathbf{F}_{\mathrm{J}_{\mathrm{E1}}}$, $\mathbf{F}_{\mathrm{J}_{\mathrm{E2}}}$, and $\mathbf{F}_{\mathrm{J}_{\mathrm{M}}}$ by considering the correlation between not only *joints at the same time step* but also *joints at different time steps*. To utilize the temporal joint features as an input of Transformer, we first concatenate the three features along the joint dimension, producing $\mathbf{F}_{\mathrm{J}} \in \mathbb{R}^{3J\times c}$. Then, a learnable positional embedding, namely kinematic and temporal positional embeddings, is applied to $\mathbf{F}_{\mathrm{J}}$. The kinematic positional embedding $\in \mathbb{R}^{J\times c}$ is applied along the joints dimension, while the temporal positional embedding $\in \mathbb{R}^{3\times c}$ is applied along the temporal dimension. The kinematic and temporal positional embedding provide relative positions in kinematic and temporal space, respectively.

Joint feature refinement with self-attention. KTFormer performs self-attention within $\mathbf{F}_{\mathrm{J}}$ by extracting query $\mathbf{q}$, key $\mathbf{k}$, and value $\mathbf{v}$ through three fully-connected layers. Following the formulation of the standard Transformer [42], refined joint features for sequential hands $\mathbf{F}'_{\text{J}_{\text{E1}}}$, $\mathbf{F}'_{\text{J}_{\text{E2}}}$, and $\mathbf{F}'_{\text{J}_{\text{M}}}$ are formulated as follows:

$$\text{Att}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \text{softmax}(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{d_{\mathbf{k}}}})\mathbf{v}, \tag{1}$$

$$\mathbf{R} = \mathbf{F}_{\text{J}} + \text{Att}(\mathbf{q}, \mathbf{k}, \mathbf{v}), \tag{2}$$

$$\mathbf{F}'_{\text{J}_{\text{E1}}}, \mathbf{F}'_{\text{J}_{\text{E2}}}, \mathbf{F}'_{\text{J}_{\text{M}}} = \psi(\mathbf{F}_{\text{J}} + \text{MLP}(\mathbf{R})), \tag{3}$$

where $d_{\mathbf{k}} = 512$ is the feature dimension of the key $\mathbf{k}$, and $\mathbf{R}$ is the residual feature. MLP denotes multi-layer perceptron, and ψ denotes a dividing operation, which separates features in dimension $\mathbb{R}^{3J \times c}$ to three $\mathbb{R}^{J \times c}$. Consequently, three joint features $\mathbf{F}'_{\text{J}_{\text{E1}}}$, $\mathbf{F}'_{\text{J}_{\text{E2}}}$, and $\mathbf{F}'_{\text{J}_{\text{M}}}$ are obtained by attentively utilizing kinematic and temporal information.

4.3. Regressor

The Regressor produces MANO [35] shape (*i.e.*, β_{E1}, β_{E2}, and β_{M}) and pose (*i.e.*, θ_{E1}, θ_{E2}, and θ_{M}) parameters, which correspond to sequential hands. We describe the regression process of the middle hand (*i.e.*, β_{M} and θ_{M}) as a representative procedure, and note that the process at different timesteps can be obtained in the same manner. First, the shape parameter β_{M} is obtained by forwarding the hand feature $\mathbf{F}_{\text{M}}$ to a fully-connected layer after global average pooling [18]. Second, the pose parameter θ_{M} is obtained by considering the kinematic correlation between hand joints. To this end, we first concatenate refined joint feature $\mathbf{F}'_{\text{J}_{\text{M}}}$ with corresponding 3D coordinates $\mathbf{J}_{\text{M}}$. Then, we flatten the concatenated feature into one-dimensional vector $\mathbf{f}_{\text{M}} \in \mathbb{R}^{J(c+3)}$. Instead of regressing poses of entire joints from $\mathbf{f}_{\text{M}}$ at once, the Regressor gradually estimates pose for each joint along the hierarchy of hand kinematic tree, following [43]. In detail, for a specific joint, its ancestral pose parameters and $\mathbf{f}_{\text{M}}$ are concatenated, and forwarded to a fully-connected layer to regress the pose parameters. By adopting the same process for both ends, three MANO parameters are obtained from the Regressor. Then, the MANO parameters are forwarded to the MANO layer to produce 3D hand meshes $\mathbf{V}_{\text{E1}}$, $\mathbf{V}_{\text{E2}}$, and $\mathbf{V}_{\text{M}}$, where each denotes to meshes at both ends and middle, respectively.

4.4. Training loss

During the training, a prediction on the middle of the motion can be simply supervised with GT of the middle frame. On the other hand, it is ambiguous to supervise both ends of motion as the temporal order is not uniquely determined, as shown in Figure 4. To resolve such temporal ambiguity during the loss calculation, we propose two items. First, we employ *temporal order-invariant loss*, which is invariant to GT temporal order [36]. To be specific, the temporal order in our loss function is determined in the direction that minimizes loss functions, not by the GT temporal order. Second, we propose to use a *Unfolder-driven temporal ordering*. It determines the temporal order based on the output of Unfolder, then uses the determined temporal order to supervise the outputs of Regressor rather than determining the temporal order of two modules separately. The effectiveness of the two items is demonstrated in the experimental section.

The overall loss function $\mathcal{L}$ is defined as follows:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{\text{U}} + \mathcal{L}_{\text{R}} \\ &= \mathcal{L}_{\text{U,M}} + \mathcal{L}_{\text{U,E}} + \mathcal{L}_{\text{R,M}} + \mathcal{L}_{\text{R,E}}, \end{aligned} \tag{4}$$

where $\mathcal{L}_{\text{U}}$ and $\mathcal{L}_{\text{R}}$ are loss functions applied to outputs of the Unfolder and the Regressor, respectively. The subscripts M and E stand for prediction of the middle and both ends.

To supervise outputs of Unfolder, we define $\mathcal{L}_{\text{U,M}}$ and $\mathcal{L}_{\text{U,E}}$ as follows:

$$\mathcal{L}_{\text{U,M}} = \mathcal{L}_{\text{joint}}(\mathbf{J}_{\text{M}}, \mathbf{J}^*_{\text{middle}}), \tag{5}$$

$$\begin{aligned} \mathcal{L}_{\text{U,E}} = \min(\; &\mathcal{L}_{\text{joint}}(\mathbf{J}_{\text{E1}}, \mathbf{J}^*_{\text{initial}}) + \mathcal{L}_{\text{joint}}(\mathbf{J}_{\text{E2}}, \mathbf{J}^*_{\text{final}}), \\ &\mathcal{L}_{\text{joint}}(\mathbf{J}_{\text{E1}}, \mathbf{J}^*_{\text{final}}) + \mathcal{L}_{\text{joint}}(\mathbf{J}_{\text{E2}}, \mathbf{J}^*_{\text{initial}}) \;), \end{aligned} \tag{6}$$

where $\mathbf{J}^*_{\text{middle}}$, $\mathbf{J}^*_{\text{initial}}$, and $\mathbf{J}^*_{\text{final}}$ are GT 3D joint coordinates of the middle, initial and final frame, respectively. $\mathcal{L}_{\text{joint}}$ is $L1$ distance between predicted and GT joint coordinates. The temporal order is determined to *forward* if the first term in min of Eq. 6 is selected as minimum and *backward* otherwise. Therefore, our loss function is *invariant to the temporal order of GT*.

To supervise the outputs of the Regressor, we define the loss function $\mathcal{L}_{\text{R,M}}$ and $\mathcal{L}_{\text{R,E}}$ as follows:

$$\mathcal{L}_{\text{R,M}} = \mathcal{L}_{\text{mesh}}(\Theta_{\text{M}}, \Theta^*_{\text{middle}}), \tag{7}$$

$$\begin{aligned} \mathcal{L}_{\text{R,E}} = &\mathbb{1}_{\mathbf{f}} \; (\; \mathcal{L}_{\text{mesh}}(\Theta_{\text{E1}}, \Theta^*_{\text{initial}}) + \mathcal{L}_{\text{mesh}}(\Theta_{\text{E2}}, \Theta^*_{\text{final}}) \;) \\ &+ \mathbb{1}_{\mathbf{b}} \; (\; \mathcal{L}_{\text{mesh}}(\Theta_{\text{E1}}, \Theta^*_{\text{final}}) + \mathcal{L}_{\text{mesh}}(\Theta_{\text{E2}}, \Theta^*_{\text{initial}}) \;), \end{aligned} \tag{8}$$

where $\mathbb{1}_{\mathbf{f}} = 1$ when the temporal order is determined to forward in Eq. 6 otherwise 0, and $\mathbb{1}_{\mathbf{b}} = 1 - \mathbb{1}_{\mathbf{f}}$. In other words, the temporal order of Eq. 8 follows that of Eq. 6, which we call *Unfolder-driven temporal ordering*. $\Theta_{\bullet} = \{\theta_{\bullet}, \beta_{\bullet}\}$ is GT or predicted MANO parameters, where the superscript * denotes GT. $\mathcal{L}_{\text{mesh}}$ is the summation of three $L1$ distances between GT and prediction of: 1) MANO parameters 2) 3D joint coordinates obtained by multiplying joint regression matrix to hand mesh 3) 2D joint coordinates projected from 3D joint coordinates.

5. Experiments

5.1. Datasets and evaluation metrics

BlurHand. The BlurHand (BH) is our newly presented 3D hand pose dataset containing realistic blurry hand images as

Methods	Train set	Test set	
		IH2.6M	BH
I2L-MeshNet [22]	IH2.6M	22.27	29.16
	BH	24.30	24.32
METRO [16]	IH2.6M	18.44	35.43
	BH	20.19	20.54
Pose2Pose [21]	IH2.6M	16.85	25.36
	BH	18.40	18.80
BlurHandNet (Ours)	IH2.6M	**15.33**	24.57
	BH	16.12	**16.80**

Table 1. **Effectiveness of BlurHand on handling blurry hand.** We calculate MPJPE (mm) on hand meshes located in the middle of the motion.

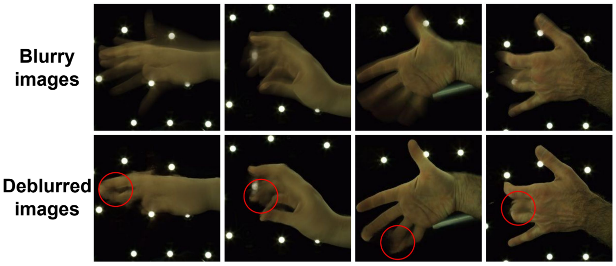


Figure 6. **Examples of deblurred images.** Since the blurry hand undergoes challenging blur from complex articulation, even the state-of-the-art deblurring method [3] cannot fully restore and blur artifact remains, highlighted with red circles.

Methods	Train set	Test set	MPJPE (mm)	MPVPE (mm)
I2L-MeshNet [22]	IH2.6M	BH+Deblur	26.56	25.23
	BH+Deblur	BH+Deblur	26.13	25.00
	BH	BH	24.32	23.08
METRO [16]	IH2.6M	BH+Deblur	26.07	32.05
	BH+Deblur	BH+Deblur	20.11	26.55
	BH	BH	20.54	27.03
Pose2Pose [21]	IH2.6M	BH+Deblur	22.43	21.04
	BH+Deblur	BH+Deblur	18.81	17.43
	BH	BH	18.80	17.42
BlurHandNet (Ours)	IH2.6M	BH+Deblur	21.37	19.93
	BH+Deblur	BH+Deblur	17.28	15.82
	BH	BH	**16.80**	**15.30**

Table 2. **Effectiveness of BlurHand compared to deblurring baseline.** MPJPE and MPVPE are calculated at hand meshes located in the middle of the motion.

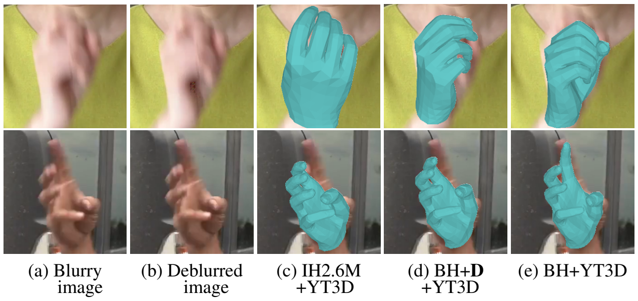

(a) Blurry image (b) Deblurred image (c) IH2.6M +YT3D (d) BH+**D** +YT3D (e) BH+YT3D

Figure 7. **Qualitative comparison on real-world blurry hand images.** The captions below figures describe datasets used to train 3D hand mesh estimation networks. A setting with **D** represents that the network is trained on deblurred BH and tested on (b).

introduced in Section 3. We train and test the 3D hand mesh estimation networks on the train and test splits of the BH.

InterHand2.6M. InterHand2.6M [23] (IH2.6M) is a recently presented large-scale 3D hand dataset. It is captured under highly calibrated camera settings and provides accurate 3D annotations for hand images. We employ IH2.6M as a representative of sharp hand frames, as the hand images in IH2.6M do not contain blur. We use a subset of IH2.6M for training and testing purposes, where the subset is a set of the third sharp frame in Figure 2 for each image of BH.

YT-3D. YouTube-3D-hands (YT-3D) [13] is a 3D hand dataset with diverse and non-laboratory videos collected from youtube. We utilize the YT-3D as an additional training dataset when testing on YT-3D. Since YT-3D does not provide 3D GTs, we only provide a qualitative comparison of this dataset without quantitative evaluations.

Evaluation metrics. We use mean per joint position error (MPJPE) and mean per vertex position error (MPVPE) as our evaluation metrics. The metrics measure Euclidean distance (mm) between estimated coordinates and groundtruth coordinates. Before calculating the metrics, we align the translation of the root joint (*i.e.*, wrist).

5.2. Ablation study

Benefit of BlurHand dataset. Directly measuring how much synthesized blur is close to the real one is still an open research problem in the deblurring community [11, 48]. Hence, we justify the usefulness of the presented BlurHand using indirect commonly used protocols [11], *i.e.*, train the model with the presented dataset and test it on unseen blurry images. Table 1 shows that all 3D hand mesh recovery networks trained on InterHand2.6M suffer severe performance drops when they are tested on BlurHand, while networks trained on BlurHand perform well. These experimental results validate that training on BlurHand is necessary when handling the blurry hand. In addition, networks trained on BlurHand also perform well on InterHand2.6M, which consists of sharp images. This shows the generalizability of our dataset to sharp images.

Table 2 shows that utilizing presented BlurHand is more valuable than applying deblurring methods [3]. The deblurring method [3], trained on our BlurHand as a pre-trained deblurring network, performs poorly on our BlurHand. Moreover, the networks trained on sharp images (IH2.6M) or deblurred images and tested on deblurred images perform worse than those trained and tested on our BlurHand. Such comparisons demonstrate the usefulness of our BlurHand dataset compared to applying deblurring methods. One reason is that, as Figure 6 shows, deblurring methods often fail to restore sharp hand images due to complicated hand motions. Another reason is that deblurring removes temporal information from the blurry image, which is helpful for reconstructing accurate 3D hand mesh sequences.

Deblur	Unfolder	KTFormer	MPJPE		
			initial	middle	final
✗	✗	✗	-	17.89	-
✗	✗	✓	-	17.41	-
✗	✓	✗	18.94	17.55	19.05
✗	✓	✓	**18.08**	**16.80**	**18.21**
✓	✗	✗	-	17.28	-
✓	✓	✓	18.95	17.28	19.10

Table 3. **Ablation study on proposed Unfolder and KTFormer.** (✓ in Deblur): The experiments with the Deblur item checked are trained and tested on deblurred BlurHand. (Second row): We only employ features from a single time step when applying KTFormer, as temporal information does not exist without Unfolder.

# of unfolding	MPJPE				
	initial	initial*	middle	final*	final
1	-	-	17.41	-	-
3 (BlurHandNet)	18.08	-	16.80	-	18.21
5	18.06	17.18	16.78	17.36	18.18

Table 4. **Ablation study on the number of unfolded hands.** The initial* and final* denote hands between the initial and middle, and the middle and final, respectively.

Figure 7 provides a qualitative comparison of real-world blurry images in YT-3D, which further demonstrates the usefulness of our BlurHand. We train BlurHandNet on three different combinations of datasets: 1) InterHand2.6M and YT-3D (7b), 2) deblurred BlurHand and YT-3D (7c), and 3) BlurHand and YT-3D (7d). The comparison shows that networks trained on BlurHand produce the most robust 3D meshes, demonstrating the generalizability of BlurHand.

Effectiveness of Unfolder and KTFormer. Table 3 shows that using both Unfolder and KTFormer improves 3D mesh estimation accuracy by a large margin. As the proposed Unfolder allows a single image to be regarded as three sequential hands, we evaluate hands in both ends and middle of the motion. Since the temporal order of hand meshes in both ends (*i.e.*, $\mathbf{V}_{\text{E1}}$ and $\mathbf{V}_{\text{E2}}$) is not determined, we report better MPJPE among the initial-final and final-initial pairs following [1, 36]. Solely employing one of Unfolder or KTFormer (the second and third rows) shows a slight improvement over the baseline network, which is designed without any of the proposed modules (the first row). On the other hand, our BlurHandNet (the fourth row) results in great performance boosts, by benefiting from the combination of two modules that effectively complement each other. In particular, KTFormer benefits from temporal information which is provided by Unfolder. Consequently, introducing both Unfolder and KTFormer, which have strong synergy, consistently improves the 3D errors in all time steps.

In the point of the baseline (the first row), using our two proposed modules, Unfolder and KTFormer, leads to more performance gain (the fourth row) than training and testing a network on deblurred BlurHand (the fifth row). This comparison shows that utilizing proposed modules is more effective than using deblurring methods. Interestingly, using our two modules does not bring performance gain when training and testing on deblurred BlurHand (the last row) compared to the deblur baseline (the fifth row). This validates our statement that deblurring prohibits networks from utilizing temporal information.

Kinematic	Temporal	MPJPE		
		initial	middle	final
✗	✗	18.99	17.79	19.06
✓	✗	18.28	16.92	18.34
✗	✓	18.79	17.41	18.92
✓	✓	**18.08**	**16.80**	**18.21**

Table 5. **Ablation study on the kinematic and temporal positional embeddings.**

Temporal order invariant loss	Unfolder-driven temporal ordering	MPJPE		
		initial	middle	final
✗	✗	18.72	16.98	18.86
✓	✗	18.44	17.14	18.55
✓	✓	**18.08**	**16.80**	**18.21**

Table 6. **Ablation study on proposed loss functions.**

Effect of the number of unfolded hands. Table 4 shows that unfolding more sequential hands further improves the 3D errors. As our KTFormer utilizes temporal information to enhance the joint feature, the number of hand sequences can affect the overall performance. Although unfolding a blurry hand into five sequential hands shows the best results, the performance is nearly saturated when a blurry hand is unfolded into three sequential hands. Considering the increased computational costs of producing additional hands and the temporal input size of KTFormer, we design our Unfolder to produce three sequential hands. We note that our BlurHandNet can be easily extended if more number of unfolding is needed for some applications.

Effect of the kinematic and temporal positional embeddings. Table 5 shows that our positional embedding setting, which uses both kinematic and temporal positional embedding, achieves the best performance. We design four variants with different positional embedding settings. The second and third rows, where either one of kinematic and temporal positional embedding is applied, achieve better results than a baseline without any positional embedding (the first row), but worse results than ours (the last row). This indicates that positional information of both kinematic and temporal dimensions is necessary for KTFormer.

Effectiveness of the proposed loss functions. Table 6 shows that two proposed items in our loss function, *temporal order-invariant loss* and *Unfolder-driven temporal ordering* as introduced in Section 4.4, are necessary for the high performance. We compare three variants for loss design at both ends of the motion, while keeping the loss function for the middle of the motion the same. In detail, the settings without *temporal order-invariant loss* are supervised with 3D meshes following the GT temporal order instead of determining the order based on Eq. 6. On the other hand,

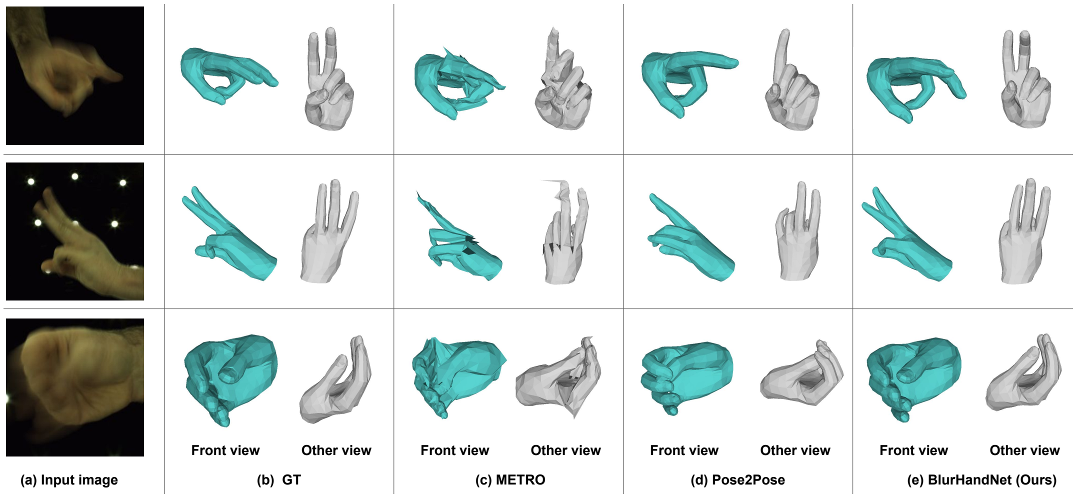
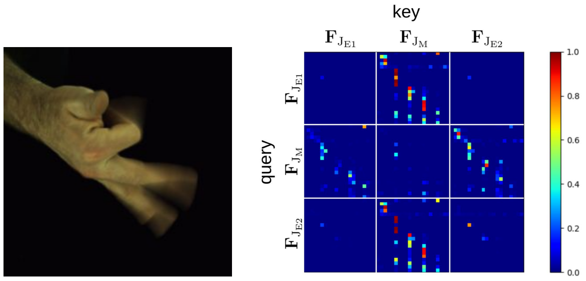


Figure 8. **Visual comparison of the BlurHandNet and state-of-the-art 3D hand mesh estimation methods [16, 21] on BlurHand.**



(a) Input image (b) Attention map

Figure 9. **Visualization of attention map from KTFormer.** The pixel at location (i,j) represents the correlation between *i-th* joint feature and *j-th* joint feature. The attention map is obtained after performing a softmax operation across each row (query).

the setting without *Unfolder-driven temporal ordering* supervises Regressor with Regressor-driven temporal ordering, which indicates that Unfolder and Regressor can be supervised with different temporal ordering. Since inferring the temporal order of GT is ambiguous, the setting without *temporal order-invariant loss* degrades the performance as it forces to strictly follows the temporal order of GT. Utilizing the proposed *Unfolder-driven temporal ordering* performs the best, as it provides consistent temporal order to both Unfolder and Regressor, making the training stable.

Visualization of attention map from KTFormer. Figure 9 shows a visualized attention map, obtained by the self-attention operation in KTFormer. Here, query and key are obtained from a combination of temporal joint features $\mathbf{F}_{\mathrm{J}_{\mathrm{E1}}}$, $\mathbf{F}_{\mathrm{J}_{\mathrm{C}}}$, and $\mathbf{F}_{\mathrm{J}_{\mathrm{E1}}}$, as described in Section 4.2. The figure shows that our attention map produces two diagonal lines, representing a strong correlation between the corresponding query and key. Specifically, features from both ends of motion, $\mathbf{F}_{\mathrm{J}_{\mathrm{E1}}}$ and $\mathbf{F}_{\mathrm{J}_{\mathrm{E2}}}$ (the first and third queries), show high correlation with the middle hand feature $\mathbf{F}_{\mathrm{J}_{\mathrm{M}}}$ (the second key), and $\mathbf{F}_{\mathrm{J}_{\mathrm{M}}}$ (the second query) shows high correlation with $\mathbf{F}_{\mathrm{J}_{\mathrm{E1}}}$ and $\mathbf{F}_{\mathrm{J}_{\mathrm{E2}}}$ (the first and third keys). This indicates that temporal information is highly preferred to compensate for insufficient joint information in a certain time step. This is also consistent with the result in Table 3. In the second row of Table 3, solely employing KTFormer without Unfolder shows slight performance improvement over the baseline due to the lack of opportunity to exploit temporal information from both ends.

5.3. Comparison with state-of-the-art methods

Table 1 and 2 show that our BlurHandNet outperforms the previous state-of-the-art 3D hand mesh estimation methods in all settings. As the previous works [16, 21, 22] do not have a special module to address blurs, they fail to produce accurate 3D meshes from blurry hand images. On the contrary, by effectively handling the blur using temporal information, our BlurHandNet robustly estimates the 3D hand mesh, even under abrupt motion. Figure 8 further shows that our BlurHandNet produces much better results than previous methods on BlurHand.

6. Conclusion

We present the BlurHand dataset, containing realistic blurry hand images with 3D GTs, and the baseline network BlurHandNet. Our BlurHandNet regards a single blurry hand image as sequential hands to utilize the temporal information from sequential hands, which makes the network robust to the blurriness. Experimental results show that BlurHandNet achieves state-of-the-art performance in estimating 3D meshes from the newly proposed BlurHand and real-world test sets.

Acknowledgments. This work was supported in part by the IITP grants [No.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University), No.2022-0-00156, No. 2021-0-02068, and No.2022-0-00156], and the NRF grant [No. 2021M3A9E4080782] funded by the Korea government (MSIT).

References

[1] D. M. Argaw, J. Kim, F. Rameau, C. Zhang, and I. S. Kweon. Restoration of video frames from a single blurred image with motion understanding. In *CVPR*, 2019. 3, 7

[2] Mehrez Boulares and Mohamed Jemni. Automatic hand motion analysis for the sign language space management. In *Pattern Analysis and Applications*, 2019. 2

[3] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *arXiv preprint arXiv:2204.04676*, 2022. 2, 6

[4] Ping Chen, Yujin Chen, Dong Yang, Fangyin Wu, Qin Li, Qingpei Xia, and Yong Tan. I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3D hand mesh modeling. In *ICCV*, 2021. 2

[5] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2Mesh: Graph convolutional network for 3D human pose and mesh recovery from a 2D human pose. In *ECCV*, 2020. 1, 2

[6] Enric Corona, Tomas Hodan, Minh Vo, Francesc Moreno-Noguer, Chris Sweeney, Richard Newcombe, and Lingni Ma. Lisa: Learning implicit shape and appearance of hands. In *CVPR*, 2022. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 3

[8] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3D annotation of hand and object poses. In *CVPR*, 2020. 1

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NeurIPS*, 2015. 4

[11] Geonung Kim Jaesung Rim, Junyong Lee Jungeon Kim, and Sunghyun Cho Seungyong Lee. Realistic blur synthesis for learning image deblurring. In *ECCV*, 2022. 6

[12] Meiguang Jin, Givi Meishvili, and Paolo Favaro. Learning to extract a video sequence from a single motion-blurred image. In *CVPR*, 2018. 2, 3, 4

[13] 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. 1, 2, 6

[14] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019. 1

[15] Chen Liang-Chieh, Papandreou George, Schroff Florian, and Adam Hartwig. Rethinking atrous convolution for semantic image segmentation. In *CoRR*, 2017. 1

[16] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 1, 2, 3, 6, 8

[17] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021. 1, 2, 3

[18] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 5

[19] Shuying Liu, Wenbin Wu, Jiaxian Wu, and Yue Lin. Spatial-temporal parallel transformer for arm-hand dynamic estimation. In *CVPR*, 2022. 3

[20] Hao Meng, Sheng Jin, Wentao Liu, Chen Qian, Mengxiang Lin, Wanli Ouyang, and Ping Luo. 3d interacting hand pose estimation by hand de-occlusion and removal. In *ECCV*, 2022. 2

[21] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Accurate 3D hand pose estimation for whole-body 3D human mesh estimation. In *CVPRW*, 2022. 2, 4, 6, 8

[22] 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, 6, 8

[23] 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, 2, 3, 6

[24] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPRW*, 2019. 3

[25] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 2, 3, 4

[26] Seungjun Nah, Sanghyun Son, Radu Timofte, Kyoung Mu Lee, et al. AIM 2019 challenge on video temporal super-resolution: Methods and results. In *ICCVW*, 2019. 2

[27] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, 2017. 2, 3

[28] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *CVPR*, 2019. 2, 3, 4

[29] JoonKyu Park, Seungjun Nah, and Kyoung Mu Lee. Pay attention to hidden states for video deblurring: Ping-pong recurrent neural networks and selective non-local attention. *arXiv preprint arXiv:2203.16063*, 2022. 2

[30] Joonkyu Park, Seungjun Nah, and Kyoung Mu Lee. Recurrence-in-recurrence networks for video deblurring. In *BMVC*, 2022. 2

[31] JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Handoccnet: Occlusion-robust 3D hand mesh estimation network. In *CVPR*, 2022. 2, 3

[32] Kuldeep Purohit, Anshul Shah, and AN Rajagopalan. Bringing alive blurred moments. In *CVPR*, 2019. 2, 3

[33] Kuldeep Purohit, Anshul Shah, and AN Rajagopalan. Bringing alive blurred moments. In *CVPR*, 2019. 4

[34] Jefferson Rodríguez, Juan Chacón, Edgar Rangel, Luis Guayacán, Claudia Hernández, Luisa Hernández, Fabio Martínez, Hiroshi Ishikawa, Cheng-Lin Liu, Tomas Pajdla, and Jianbo Shi. Understanding motion in sign language: A new structured translation dataset. In *ACCV*, 2020. 2

[35] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *SIGGRAPH Asia*, 2017. 4, 5

[36] Denys Rozumnyi, Martin R. Oswald, Vittorio Ferrari, Jiri Matas, and Marc Pollefeys. Defmo: Deblurring and shape recovery of fast moving objects. In *CVPR*, 2021. 2, 5, 7

[37] Ziyi Shen, Wenguan Wang, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *ICCV*, 2019. 3

[38] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018. 2

[39] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, 2017. 2

[40] Daisuke Sugimura, Yusuke Yasukawa, and Takayuki Hamamoto. Using motion blur to recognize hand gestures in low-light scenes. 2016. 2

[41] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018. 4

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 2, 4, 5

[43] Ziniu Wan, Zhengjia Li, Maoqing Tian, Jianbo Liu, Shuai Yi, and Hongsheng Li. Encoder-decoder with multi-level attention for 3D human shape and pose estimation. In *ICCV*, 2021. 5

[44] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 1

[45] John Yang, Hyung Jin Chang, Seungeui Lee, and Nojun Kwak. Seqhand: Rgb-sequence-based 3d hand pose and shape estimation. In *ECCV*, 2020. 2

[46] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. 2, 4

[47] K. Zhang, W. Luo, B. Stenger, W. Ren, L. Ma, and H. Li. Every moment matters: Detail-aware networks to bring a blurry image alive. In *ACMMM*, 2020. 3, 4

[48] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *CVPR*, 2020. 6

[49] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, Haozhe Xie, Jinshan Pan, and Jimmy S. Ren. Davanet: Stereo deblurring with view aggregation. In *CVPR*, 2019. 3

[50] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single RGB images. In *ICCV*, 2019. 1, 2