EgoBlur: Blurry Egocentric XR Dataset for Robust Fast Hand Pose Estimation

Category: Research

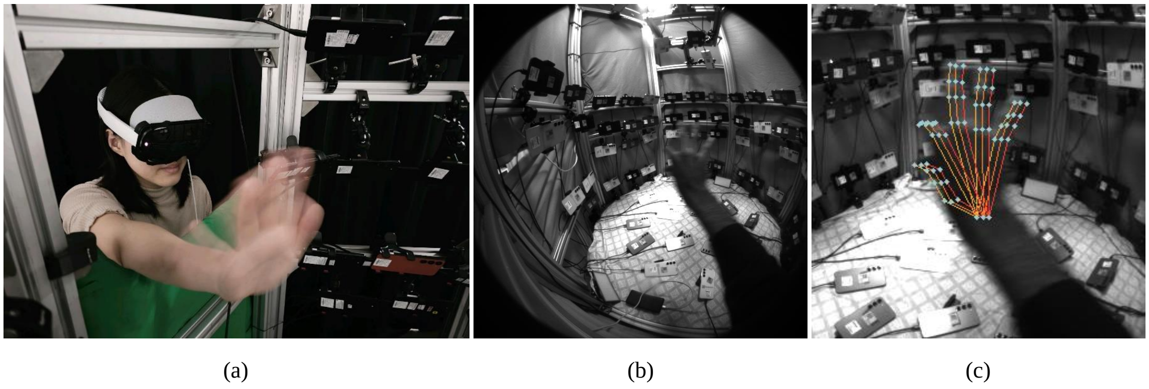


Figure 1: Our EgoBlur dataset captures dynamic hand motions with real motion blur (b) from an egocentric perspective using a prototype HMD device. Corresponding annotations are generated for the blurred images and encode the trajectory of hand motion, providing temporal information within a single frame as shown in (c).

Abstract

Hand tracking in XR serves as a fundamental interaction mechanism, as it allows users to directly interact with virtual content. Accurate 3D hand pose estimation is essential in scenarios involving dynamic hand motions, such as gaming, sports, and virtual musical instruments. These dynamic hand motions often result in motion blur when the hand moves faster than the frame rate of the cameras, making pose estimation challenging. The state of the art methods for 3D hand pose estimation uses deep learning that requires large amounts of data with 3D hand pose ground truth. However, most of the existing publicly available hand pose datasets are captured from static or slowly moving hands that do not contain any explicit motion blur. While techniques such as using short exposure times with higher frame rates have been employed to reduce motion blur, they still pose limitations for developing accurate hand pose estimation algorithms in the presence of fast motion. To address these challenges, firstly, we introduce a new dataset, EgoBlur, consisting of egocentric hand videos with real blur captured from a prototype Head-mounted headset. Our dataset contains $\sim$ 100k images along with accurate and temporally consistent 3D hand pose ground truth. Secondly, we propose EgoBlurNet, a deep learning model capable of estimating 3D hand keypoints from blurry egocentric images by employing a teacher-student paradigm. Experimental results demonstrate that our method provides reliable and accurate 3D hand pose for blurred hand images compared to existing methods, especially in realistic dynamic XR scenarios.

Index Terms: Data capture system, hand pose estimation, motion blur image, extended reality

1 Introduction

The technology of eXtended Reality (XR), which includes Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR), has rapidly developed, enabling various types of immersive experiences. In XR environments, hand-based interaction is considered one of the most natural and intuitive input methods [17]. Hands function as a primary interface for interacting with virtual content, allowing users to manipulate and explore objects in XR applications. To support accurate and stable hand interaction, XR devices are equipped with sensors such as depth cameras [35] and monochrome tracking cameras [11]. Based on the captured image frames from these sensors, a vision-based hand pose estimation model is applied to track and recognize hand gestures.

Recent advances in data-driven approaches based on deep learning have demonstrated significant progress in hand pose estimation [6, 32, 34]. These methods leverage large-scale, high-quality hand datasets to train deep neural networks, thereby enabling natural and precise hand interaction. Various datasets, which were constructed from depth images [33], RGB images [16], and egocentric-view images [20], have been introduced to facilitate this task. However, to obtain accurate ground-truth annotations, most of these datasets have been collected under constrained conditions, such as static [19, 31] or slowly moving hands [12, 23], which limits their applicability in real-world dynamic interaction scenarios.

Hand movements in XR environments are often fast and dynamic, particularly during tasks such as immersive gaming, virtual sports, and musical instrument simulations. In these scenarios, accurate tracking of high-speed hand motion is essential to ensure smooth user interaction. To obtain clear hand images and enable stable tracking with vision-based deep learning models, XR devices typically employ very short exposure times, often below 1ms, and operate at high frame rates. However, these configurations lead to increased power consumption, posing a major challenge for mobile, battery-powered XR headsets [18]. A common alternative is to reduce the frame rate or increase the exposure time to lower the exposure gain. Nevertheless, such adjustments frequently result in motion blur, especially during dynamic hand interactions. Additionally, under low-light conditions when the exposure time of the cameras increases, images are more susceptible to motion blur, which significantly degrades the performance of deep learning models trained exclusively on high-light and blur-free images. The demand for Head-Mounted Display (HMD) devices is increasing with the release of products such as Apple Vision Pro (AVP) and Meta Quest3 (MQ3), while AI-based functionalities are becoming more energy-demanding and are expected to operate in diverse en-

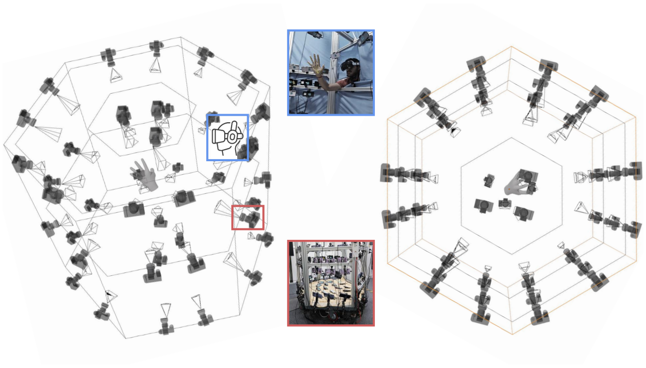
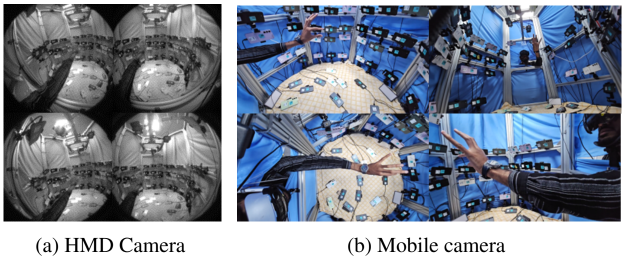


Figure 2: Our proposed data capture system. In the multi-camera setup, egocentric blurry images are captured using our prototype XR device at 50 fps, and ground-truth annotations are validated with a high-speed multi-camera system operating at 120 fps.

vironments, including outdoor settings and low-light indoor conditions. Under such circumstances, motion blur in captured images becomes inevitable, and it is essential to develop methods that can robustly handle these blurred hand images.

While some previous studies have explored hand pose estimation using blurred images [22, 25], several limitations prevent their direct application to XR hand interaction scenarios. Firstly, most of these studies do not utilize egocentric views which are essential for Head-mounted devices in XR. They are typically based on the InterHands 2.6M dataset [19] or synthetically generated blur using the YT-3D dataset [16], both of which provide third-person perspectives. Secondly, the images are undistorted and RGB-based, which differs significantly from the imaging characteristics of XR devices that typically produce distorted monochrome images [12]. Apart from containing motion blur, these datasets tend to contain less visual noise compared to real hand movements in XR environments.

To address these limitations, we present EgoBlur, a large-scale dataset of egocentric hand videos with naturally occurring motion blur, captured directly from an XR head-mounted device. The dataset consists of $\sim$ 100K images of real hand motions commonly observed in XR applications, such as dragging and zooming with pinch or false-pinch poses, grabbing, and counting. It provides raw frames from continuous video sequences, each lasting approximately 8 seconds, enabling rich temporal information for model learning. Fast hand motions were captured using an HMD device with an exposure time of 20ms to produce real motion blur. To validate the captured motion and generate accurate ground-truth annotations, we used a high-speed multi-camera system operating at over 120fps. Note that while the HMD camera operated at lower frame rates (50fps) due to its longer exposure time, the high-speed camera system ensured temporally precise annotation. To supplement the amount of blurred data, we also provide synthetically generated blurred images, enabling a large and more balanced dataset for training. Inspired by the BlurHands [22] concept of compressing temporal information into a single motion-blurred image, we provide multiple keypoint annotations per image, enabling the representation of multiple hand trajectories within a single blurred frame. Furthermore, we propose a deep learning model, EgoBlurNet, designed to estimate hand keypoints from motion-blurred images. EgoBlurNet adopts a teacher–student knowledge distillation framework [13], where the teacher model is trained on sharp images and the student model is trained on blurred images. The teacher transfers high-quality pose representations to the student model, enabling the student to learn robust hand pose estimation solely from blurry inputs. This approach allows EgoBlurNet to achieve accurate keypoint estimation even in the presence of severe motion blur.



(a) HMD Camera

(b) Mobile camera

Figure 3: Sample dataset captured using the multi-camera setup. (a) Sample images from the EgoBlur dataset, captured using our XR prototype device. (b) Images captured from high-speed cameras, used for generating ground-truth annotations.

With our EgoBlur dataset and the proposed EgoBlurNet model, we address the challenge of hand pose estimation under motion blur conditions commonly encountered in XR devices. Such blur frequently occurs in scenarios with low frame rates (to reduce power consumption) or under low-light conditions, both of which are typical in real-world XR usage. Our dataset and model offer a practical solution for handling these challenging cases. The main contributions of this paper are summarized as follows:

- We present **EgoBlur**, a new egocentric dataset of blurry hand images containing real hand motions, captured directly from an HMD device. We provide multiple trajectory keypoints annotations embedded within a single blurred frame.

- We propose **EgoBlurNet**, a novel deep learning model for 3D hand keypoint estimation from blurry images, trained using our EgoBlur dataset. The model leverages knowledge distillation from a teacher network that is trained on sharp images.

- We experimentally demonstrate that EgoBlurNet outperforms existing hand pose estimation models under realistic XR conditions, and validate the effectiveness of the proposed dataset.

2 Related Work

Hand interaction in XR device is commonly achieved using vision-based deep learning techniques, particularly 3D hand pose estimation models. In the following sections, we review existing 3D hand pose datasets, which are essential for training accurate models, as well as representative 3D hand estimation methods.

2.1 3D Hand Pose Dataset

Early 3D hand pose datasets often relied on depth cameras with sensor-based [38] or model-based [33] annotations. However, their susceptibility to occlusions and high power consumption limited scalability and diversity, leading to a shift toward RGB-only systems. Simon et al. [30] introduced a multi-camera RGB setup to reduce occlusions and employed an iterative bootstrapping strategy for large-scale annotation. We also adopt a similar multi-camera setup and annotation method to obtain ground-truth labels using high-speed cameras. A number of large-scale hand pose datasets have been collected using multi-camera systems. FreiHAND [41] and InterHands 2.6M [19] provide large-scale RGB datasets from third-person views. GaneratedHands [20] presents a synthetic dataset with egocentric views, while YT-3D [16] contains real-world third-person hand images collected from YouTube videos. UmeTrack [12] and AssemblyHands [23] offer egocentric-view datasets captured directly from HMD devices. However, almost all of these datasets were collected under static or slow-motion conditions, and thus do not include motion-blurred hand images.

More recently, BlurHand [22] and EBH [25] introduced a dataset containing motion-blurred hand images. However the blur in BlurHand [22] is synthetically generated, making it less realistic compared to actual motion blur. EBH [25] captures real motion blur, but it is not captured from an egocentric view using an HMD, which limits its relevance to XR device environments.

2.2 3D Hand Pose Estimation

Extensive research has been conducted on 3D hand pose estimation using depth images [8], single RGB images [9, 24], and multi-view RGB or grayscale images [4, 6]. These approaches include skeleton-based methods [14, 20, 40], which estimate 3D hand keypoints, and mesh-based methods [1,2], which estimate mesh parameters from hand models such as MANO [27]. Most of these methods adopt convolutional neural networks (CNNs) as their backbone, performing either heatmap regression [14] or direct regression [21] of hand poses or meshes. More recently, Transformer-based models [15, 36] have improved estimation accuracy by leveraging attention mechanisms. In addition, real-time hand pose estimation methods [7, 39] have been introduced to meet the memory and latency constraints of XR environments. For practical use cases, several models also address hand–hand [37] and hand–object interactions [3]. However, only a few existing models focus on hand pose estimation from motion-blurred images. Even recent Transformer-based methods [22,25] are primarily designed for RGB third-person view inputs, limiting their applicability in XR scenarios with egocentric and blurred inputs.

3 EgoBlur Dataset

With the growing popularity of XR devices in recent years, there is an increasing demand for high-quality hand pose estimation datasets. Many XR applications, such as immersive gaming, virtual musical instruments, and even the device interactions such as pinching, swiping, etc that involve fast hand movements make precise hand tracking challenging. To address this issue, we propose the first real egocentric hand dataset with naturally occurring motion blur, captured directly from an HMD device. The dataset contains $\sim$ 100k temporally continuous, egocentric, multi-view images collected from an HMD, along with accurate 2D and 3D hand keypoint annotations. EgoBlur consists of two different types of blurry datasets, namely, EgoBlur (SB) and EgoBlur (Real). EgoBlur (SB), where SB stands for Synthetic Blur, has HMD images with synthetically created blur by performing a weighted average of consecutive frames. We have $\sim$ 75k blur images in this dataset. EgoBlur (Real) contains $\sim$ 25k HMD images with Real blur captured from fast moving hands using the strategy explained in Section 3.1. In addition, we provide keypoint annotations that represent the trajectories embedded in motion-blurred images. This facilitates training models capable of recovering temporally consistent motion information from blurred inputs. EgoBlur (Real) dataset comprises of methodically chosen hand poses that typically introduce motion blur in XR interaction scenarios such as pinch and swipe, grab, etc. We provide pose-wise details of the real dataset in Figure 6.

3.1 Multi-View Data Capture System

Our proposed data capture system is illustrated in Figure 2. We constructed a custom high-speed multi-camera system using 70 Samsung S23 Plus smartphones, each capable of recording video at 120 frames per second (fps) with an exposure time of $\sim$ 8ms, following the setup described in [28]. This multi-camera setup [31] enables the generation of precise ground-truth annotations with/without hand occlusions, as the smartphones are arranged in a multi-tier hexagonal configuration, all oriented toward the central capture area to cover the hands from all possible viewpoints. Our prototype HMD device is used for acquiring egocentric data. During data acquisition, a subject stands right behind the HMD device to obtain a real egocentric view of their hands. To ensure head stability during recording, the HMD is mounted to the capture system itself. Fast hand gestures are performed by the subject, as listed in Figure 6. The setup also incorporates a highly responsive, wirelessly controlled lighting system capable of dynamically adjusting illumination levels from 0.01 lux to 1000 lux. The dataset for creating EgoBlur (SB) is captured in a setting such that smartphones captures hand motion with an exposure time of $\sim$ 8ms and a dump rate of 120fps, along with the HMD which captures frames with an exposure time of $\sim$ 2ms and a dump rate of 60fps. Because the captures are at a lower exposure time, both the set of frames are sharp. On the contrary, for creating EgoBlur (Real) dataset, we use similar settings for the smartphones, while the HMD is configured to capture with a higher exposure time of $\sim$ 20ms and a dump rate of 50fps. This configuration for the EgoBlur (Real) dataset ensures that natural motion blur occurs in the HMD captured images, while the smartphone captures have minimal amount of motion blur. Figure 3 showcase representative images captured from the HMD and the smartphone devices.

3.2 Synchronization and GT generation

We capture videos of fast hand motions using both smartphones and the HMD device. The video recording is triggered at the same time on all these devices. All the devices are connected over Wi-Fi and are controlled using a system called Jhammer [28], which transmits the capture signals via socket communication at almost similar time. To synchronize the frames captured from smartphones and HMD, we propose a strategy of intensity based frames synchronization. Before starting to capture the sequences, we initially dim the lighting system to 0.01 lux. The capture is then triggered on HMD and 70 mobile devices simultaneously. After almost 2 seconds of captures, the lighting system is again switched back to 500 lux instantly. Then, we continue recording for an additional $\sim$8 seconds. Synchronization is achieved by identifying the frame at which a sudden change in mean pixel intensity crosses a predefined threshold across all devices. For each video sequence capture, let I_n^m, denote device (smartphone/HMD) camera frame where n denotes device index and m denotes frame index in video sequence. We compute the mean intensity for all frames $1...m$, and that for all the devices $1...n$, as explained in Equation 1. Frame captured for each smartphone which has $\overline{I_n^m}$ greater than a intensity threshold are labelled as starting frame of the synchronized video sequence.

$$\overline{I_n^m} = \frac{1}{N}\sum_{i=1}^{N} Intensity(I_n^m(i)) \tag{1}$$

where N is total number of pixels in the frame.

This ensures that the initial frames are temporally aligned and that the alignment is maintained throughout the sequence. We save the timestamps for all the frames of smartphones as well as HMD after we get the first corresponding synced frame. The synchronization latency between high-speed cameras is measured at approximately 8 ms, and the latency between the cameras and the XR headset is approximately 20 ms.

Once the frames are synchronized, we perform calibration between the smartphone images and the HMD images to compute the camera calibration parameters. Since the HMD has been mounted on the capture system and smartphones are also static for a 10 seconds sequence, we compute the calibration parameters for the first synchronized frames from all the devices using COLMAP [29]. The same camera parameters are then used for the subsequent frames of the capture sequence. Due to the complexity of hand articulations, manual annotation of all keypoints across the entire dataset is highly challenging. Previous methods, such as those proposed in [31] and [19], utilize RANSAC-based approaches for automated hand pose labeling. However, these methods rely on

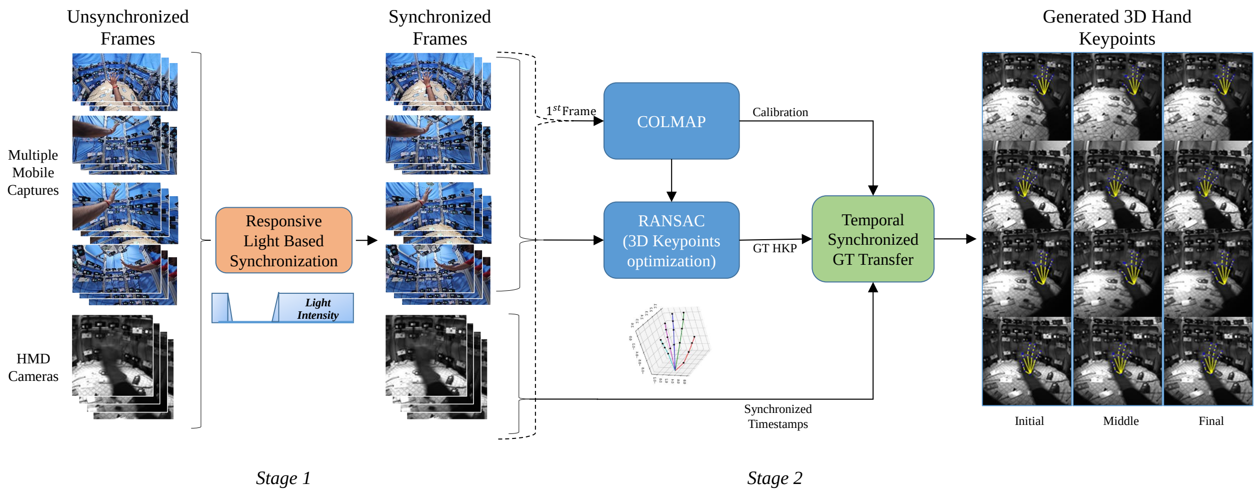


Figure 4: Overall pipeline of our data capture system. It consists of two major stages. Stage 1 is the Synchronization stage, where all the frames captured are synchronized using an intensity-based synchronization. Stage 2 consists of 2D/3D ground truth generation and transform to HMD motion-blurred images.

highly accurate camera calibration. Since our dataset is calibrated using COLMAP, certain cameras may exhibit sub-optimal calibration, which can adversely affect RANSAC-based inlier detection if these methods are applied directly. To address this, motivated by the work in [28] which introduced a COLMAP-based RANSAC optimization framework for annotating static multi-view hand data, which accounts for cameras with poor calibration parameters, we extend this method to support multi-view video sequence capture, incorporating an additional HMD device. Following describes the proposed method in [28] simplified for our capture setup.

To obtain ground truth 3D hand keypoint annotations from multi-view images, we employ an iterative multi-view refinement approach. Let I_n denote the image from the n^{th} camera view. For each image I_n, we use the pre-trained HaMeR [26] hand pose estimator to obtain an initial estimate of the 2D keypoints, denoted as $P^{'}_{2d}$. We then refine these estimates through an iterative process that identifies reliable camera views with accurate calibration and pose annotations. Let C_i represent the set of camera views used in iteration i. Initially, C_0 includes all available camera views in our multi-camera capture setup. At each iteration i, we apply RANSAC to identify the inlier views for each keypoint based on reprojection error. Specifically, for keypoint k, we define the inlier set $C_i^{\text{in},k}$ as the subset of views from C_i whose reprojection error is less than 10 pixels. Using these inlier sets, we estimate the pseudo 3D location of each keypoint k by minimizing the reprojection error via the BFGS optimization algorithm as defined in Equation 2.

$$P_{3d}^{k,i} = \arg\min_{\mathbf{x}} \sum_{v \in C_i^{\text{in},k}} \left\| \mathscr{P}_v(\mathbf{x}) - P_{2d}^{\prime(k,v)} \right\|_2 \tag{2}$$

where $P_{3d}^{k,i}$, is the required 3D location of the k^{th} keypoint after i^{th} iteration, $P_{2d}^{'(k,v)}$ denotes sub-optimal label of k^{th} keypoint for view v and $\mathscr{P}$ denotes projection function to view v. Subsequently, we compute the average reprojection error $E_{(i,v)}$ for each view v across all keypoints as follows:

$$E_{(i,v)} = \sum_{k=1}^{21} \left\| \mathscr{P}_v\left(P_{3d}^{k,i}\right) - P_{2d}^{\prime(k,v)} \right\|_2 \tag{3}$$

Views with an average error $E_{(i,v)}$ exceeding 30 pixels are considered to have inaccurate calibration and are excluded from C_{i+1}, the view set for the next iteration. Empirically, we find that two iterations of this procedure are sufficient to yield accurate 3D keypoint reconstructions and consistent 2D projections across the camera views.

GT Generation for EgoBlur (SB) dataset. The corresponding sets containing both smartphones and HMD images are processed using the pipeline above and the 2D/3D GT keypoints are obtained for all these views. Once we get the GT keypoints for HMD views, we perform a weighted average on window of consecutive HMD frames to obtain blur synthetically. The blur image contains a set of three different keypoints called initial, middle and final keypoints, similar to [22]. Initial keypoint comes from the first frame of the window, middle keypoint is obtained from center frame, and final keypoint comes from the last frame of the window. Also, for each blur frame in this dataset, we have a sharp frame which is the center frame of the window. Let us denote this dataset as EgoBlur (Sharp).

GT Generation for EgoBlur (Real) dataset. Unlike EgoBlur (SB), we use RANSAC pipeline described above to compute 2D/3D GT keypoints for only corresponding frames from smartphones. Once the GT keypoints is generated for all the synchronized smartphone frames sets, we transform these GT keypoints onto the blur hands frames captured from HMD using the following technique. For every HMD frame, we identify the smartphone frames set with the closest timestamps and the same 3D hand keypoints are transferred to the HMD viewpoints to obtain accurate 3D/2D ground truth (GT) hand keypoints despite the frames being blurred. The sample of our EgoBlur (Real) dataset can be seen in Figure 5.

4 EgoBlurNet

Estimating 3D hand pose becomes particularly challenging in the presence of motion blur. Neural networks often struggle to learn meaningful features under such conditions when trained solely with direct supervision from ground-truth labels. To address this issue, we propose EgoBlurNet, a deep learning model specifically designed to estimate 3D hand keypoints from motion-blurred images. Our approach leverages a teacher-student knowledge distillation framework, as illustrated in Figure 7. We sample pairs of

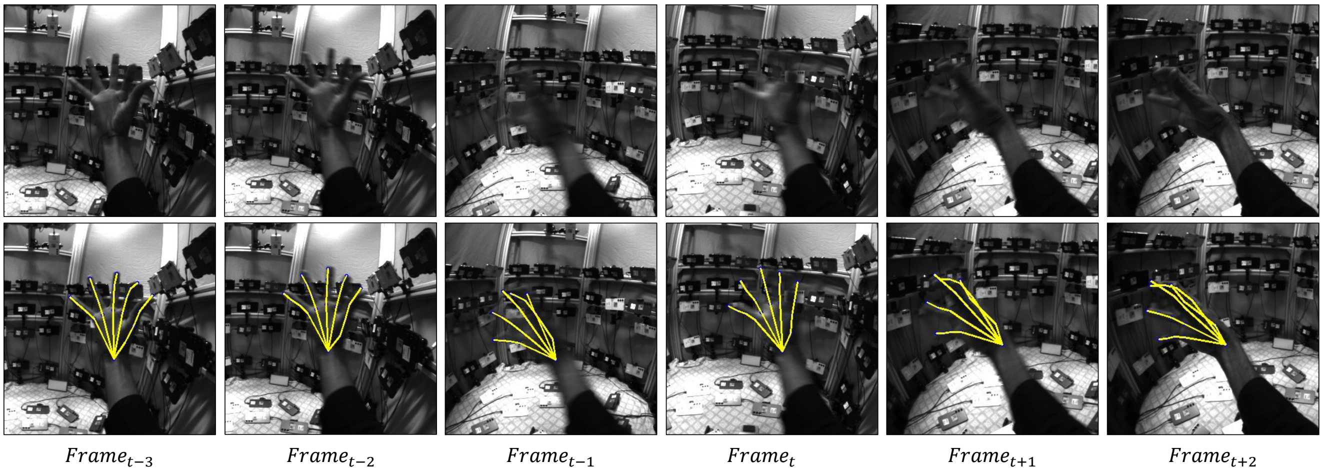


Figure 5: EgoBlur Dataset samples captured directly from a HMD and their annotations. Sample frames from 6 consecutive timestamps. Frames from timestamps t and $t-1$ depicts real motion blur captured from HMD of a fast moving hand.

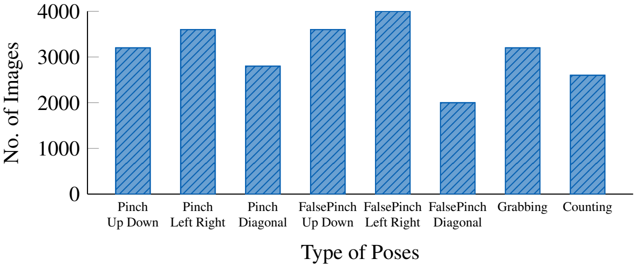


Figure 6: Distribution of different poses in EgoBlur (Real) dataset

sharp images(I^s) and corresponding blurry images(I^b) , which are processed by the sharp-pose estimator (N^s) and the blurry-pose estimator (N^b), respectively. During inference, only the blurry estimator N^b is used as a standalone model to predict hand poses from a single blurry image. Both N^s and N^b utilize the Cascaded Pyramid Network (CPN) [5] as the backbone architecture, as depicted in Figure 7. Our network incorporates a ResNet-50-based feature extractor, followed by the GlobalNet and RefineNet modules [5] to generate hand keypoint predictions. To enable accurate 3D pose estimation, the RefineNet module is extended with two parallel prediction heads: one predicts the 3D location of each hand keypoint relative to the wrist, while the other estimates the absolute depth of the wrist keypoint. The final 3D positions of all hand keypoints are obtained by combining these two outputs. To mitigate the challenge posed by the limited availability of labeled real-world data, we adopt a three-stage training strategy, which is detailed below.

4.1 Synthetically Generated Motion-Blur Training

In the first stage, we pre-train both the sharp-pose estimator N_s and the blurry-pose estimator N_b using the synthetically generated motion-blur subset of the EgoBlur dataset. This pre-training phase enables the models to learn generalizable feature representations for hand pose estimation. Notably, the intermediate feature maps produced by the trained N_s contain rich semantic information that is essential for accurately predicting hand keypoints. We denote Mean Squared Error (MSE) loss that is minimized for fine-tuning with synthetic data as $\mathscr{L}_{\text{SYN}}$.

$$\mathscr{L}_{\text{SYN}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\widehat{y^i_{3D}} - y^i_{3D}\right\|^2 \qquad (4)$$

where $\widehat{y^i_{3D}}, y^i_{3D}$ denote the predicted and ground-truth hand keypoint labels respectively.

4.2 Sharp Image Guidance

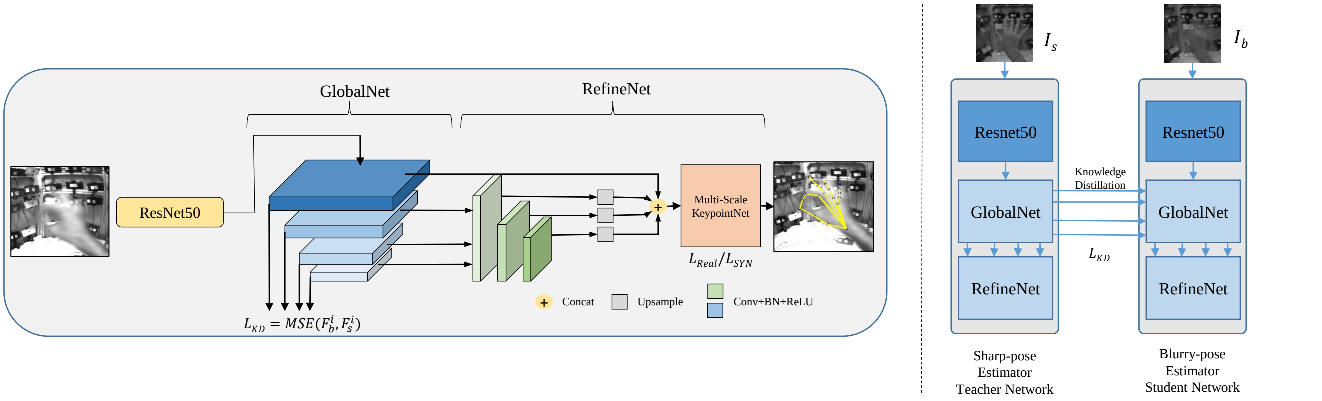
In the second stage as shown in Figure 7(b), we distill the rich hand pose representations learned by N_s to guide the training of N_b. During each training iteration, a sharp image is provided as input to N_s, while the corresponding motion-blurred image I_b is fed to N_b. The weights of N_s are kept frozen throughout this stage to serve as a fixed teacher network. We extract t intermediate feature maps, denoted as $F_s^{1\cdots t}$ and $F_b^{1\cdots t}$, from the sharp and blurry image streams, respectively. Our objective is to minimize the feature distance between corresponding pairs (F_s^i, F_b^i) for $i = 1\cdots t$, thereby transferring the rich semantic information from N_s to N_b. In addition to feature distillation, we also employ ground-truth 3D hand pose annotations in camera coordinates to directly supervise the training of N_b. We minimize the knowledge-transfer loss $\mathscr{L}_{\text{KT}}$ as follows:

$$\mathscr{L}_{\text{KT}} = \frac{1}{T}\sum_{i=1}^{t}\left\|F_s^i - F_b^i\right\|^2 \qquad (5)$$

4.3 Real Motion-Blur Fine Tuning

In the final stage, we fine-tune the blurry-pose estimator N_b using real-world motion-blurred images from the EgoBlur dataset. While synthetic data provides a controlled environment to pre-train and transfer high-level representations, it often lacks the complexity and variability present in real-world scenarios, such as lighting variations, sensor noise, and diverse hand shapes or poses. Fine-tuning on real blurred images enables the model to bridge the domain gap between synthetic and real data, allowing N_b to adapt to subtle cues and noise patterns that are difficult to simulate. As a result, this stage significantly enhances the model's robustness and performance. We denote MSE loss that is minimized for fine-tuning with real data as $\mathscr{L}_{\text{Real}}$.

$$\mathscr{L}_{\text{Real}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\widehat{y^i_{3D}} - y^i_{3D}\right\|^2 \qquad (6)$$



(a) *(b)*

Figure 7: Proposed EgoBlurNet Architecture. (a) depicts the model architecture. (b) depicts the process of knowledge distillation from sharp hand pose estimator to Blurry hand pose estimator.

where $\widehat{y_{3D}^i}, y_{3D}^i$ denote the predicted and ground-truth hand keypoint labels respectively.

5 Experiments & Results

5.1 Comparison of different State-of-the-art methods

We conduct experiments to evaluate the effectiveness of the proposed EgoBlur dataset and the EgoBlurNet model. For the experimental setup, we compare several state-of-the-art 3D hand pose estimation approaches, including A2J Transformer [15], and Keypoint Transformer [10], as well as BlurHand [22], which is trained specifically on blurry hand images. For the evaluation datasets, we use UmeTrack [12], an egocentric dataset captured from an HMD, which shares the same domain characteristics as EgoBlur. To simulate blurry conditions, we generate a synthetically blurred version of UmeTrack by averaging consecutive frames (referred to as UmeTrack (SB)). We also include the BlurHand [22] dataset, which provides synthetic blurry images in the InterHands 2.6M [19] format, along with both synthetic and real blur versions of our EgoBlur dataset. For evaluation, we use the 2D Mean Per Joint Position Error (MPJPE), computed by projecting local 3D keypoints onto the image plane and measuring the pixel-wise differences from the ground truth. We prepare testing sets for EgoBlur (SB) and EgoBlur (Real) consisting of 7.5k and 2.5k images respectively.

We perform another experiment in which we fine-tune A2J Transformer [15] and Keypoint Transformer [10] using our EgoBlur (SB) and EgoBlur (Real) datasets and evaluate on our testing sets. Table 2 shows that our EgoBlurNet outperforms both A2J Transformer [15] and Keypoint Transformer [10] by $\sim 6\times$ on EgoBlur (SB) and $\sim 4\times$ on EgoBlur (Real) respectively.

We also provide the qualitative comparisons of the above SOTA methods and EgoBlurNet on our EgoBlur (Real) datasets in Figure 8. Our method is able to generate accurate hand-pose labels even for severely blurred hands.

Table 1: Quantitative comparisons (MPJPE) for different state-of-the-art hand pose estimation methods on blurry hands datasets including our EgoBlur (SB) and EgoBlur (Real) datasets. Best in **bold**, second best underlined.

Methods	MPJPE			
	UmeTrack (SB)	BlurHands	EgoBlur (SB)	EgoBlur (Real)
BlurHands [22]	**19.30**	**6.30**	**10.15**	**13.58**
A2J Transformer [15]	58.52	17.21	58.22	64.86
Kypt transformer [10]	44.81	18.33	68.35	61.18

Table 2: Quantitative comparisons (MPJPE) against A2J Transformer [15] and Kypt transformer [10] fine-tuned on EgoBlur (SB) and EgoBlur (Real) datasets and EgoBlurNet (Ours). Best in **bold**, second best underlined

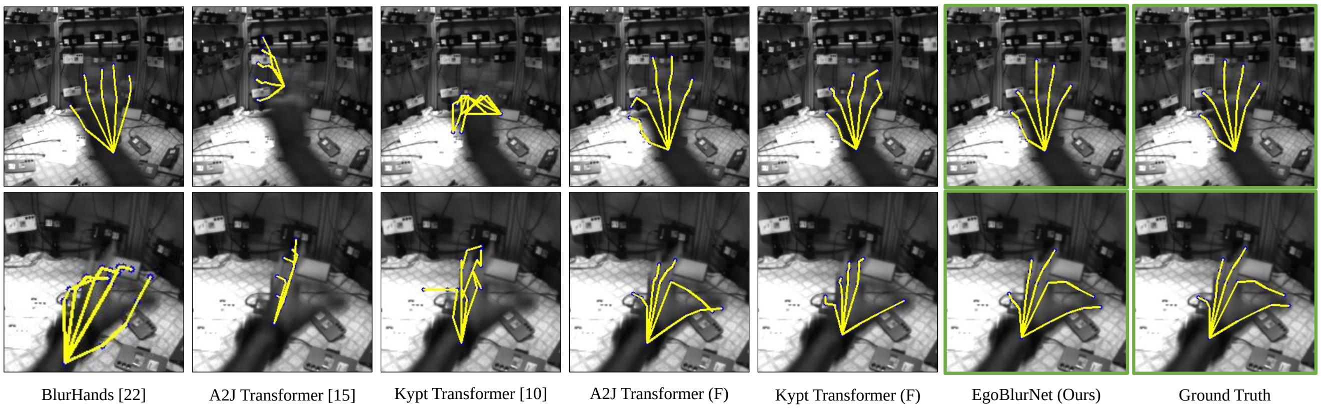
Methods	MPJPE	
	EgoBlur (SB)	EgoBlur (Real)
A2J Transformer (F) [15]	6.45	11.78
Kypt transformer (F) [10]	7.08	16.79
EgoBlurNet (Ours)	**1.41**	**3.98**

5.2 Ablation Study

Impact of different training strategies on the performance of EgoBlurNet. We conduct a comprehensive ablation study to evaluate the effectiveness of different training configurations and learning strategies for handling motion blur on our proposed EgoBlur dataset. Specifically we analyse how the choice of training data - EgoBlur (Sharp), EgoBlur (SB) and EgoBlur (Real) affects the performance of our proposed EgoBlurNet. Furthermore, we explore the role of Knowledge distillation (KD) in transferring pose estimation knowledge from sharp images to blurred domain. The performance is reported in terms of MPJPE on both EgoBlur (SB) and EgoBlur (Real) test sets in Table 3.

These ablation results confirm the effectiveness of our key strategy of Knowledge distillation to transfer robust hand pose features from sharp to blurred images. Our final model which is trained using KD and fine-tuned on EgoBlur (Real) sets a new benchmark on the EgoBlur dataset.

Impact of different frame rates and exposure times on XR device performance. Table 4 presents the effect of frame rate and exposure time on the performance of XR devices. We measured power consumption, current drawn, and CPU load as performance metrics. Each measurement was conducted three times over a one-minute period, and the average was used for analysis. Measurements were taken under different system settings, and the results in the table represent the difference from the default configuration, calculated as (default value - value under current setting). Reducing the frame rate leads to a significant decrease in all performance metrics, including power, current and CPU load. This suggests


BlurHands [22]

A2J Transformer [15]

Kypt Transformer [10]

A2J Transformer (F)

Kypt Transformer (F)

EgoBlurNet (Ours)

Ground Truth

Figure 8: Qualitative comparison of different state-of-the-art hand pose estimation models and EgoBlurNet (Ours) on EgoBlur (Real) dataset. Row1 and Row2 are different samples from EgoBlur (Real) dataset. (F) represents Fine-tuning using EgoBlur dataset. Visual results clearly show that our method estimates the hand-pose most accurately for images having significant motion blur.

Table 3: Table representing various combination and strategies for training EgoBlurNet and the corresponding MPJPE on EgoBlur (SB) and EgoBlur (Real) test sets.

Methods	Training Dataset			MPJPE	
	EgoBlur (Sharp)	EgoBlur (SB)	EgoBlur (Real)	EgoBlur (SB)	EgoBlur (Real)
EgoBlurNet	✓	-	-	31.96	40.12
	-	✓	-	2.47	28.86
	-	✓	✓	1.67	9.43
EgoBlurNet + KD	✓	✓	-	1.89	14.87
	✓	✓	✓	**1.41**	**3.98**

Table 4: Impact of different frame rates and exposure times on XR device performance. We report the difference of values from the default setting (60 fps, exposure time 0.8ms)

FPS	Exposure Time	Performance Measure		
		Power (W) ↓	Current (A)↓	CPU load (%) ↓
50 fps	20 ms	0.087	0.007	2.4
30 fps	0.8 ms	0.184	0.013	13.7
30 fps	20 ms	0.245	0.017	13.9

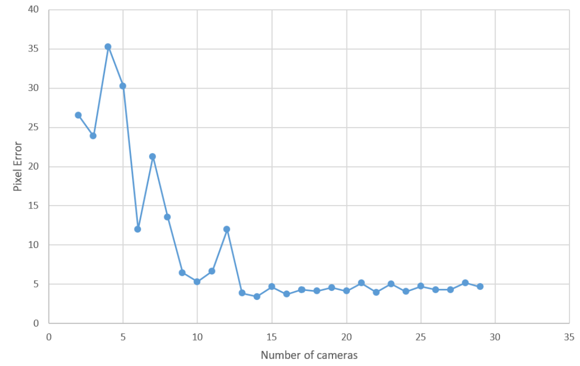


Figure 9: Effectiveness of our annotation pipeline. We compute the pixel error between manually labeled hand keypoints and the ground-truth annotations generated by our method.

that maintaining a high frame rate imposes a considerable computational and energy cost on XR devices. In terms of exposure time, there is only a small difference across all metrics, but a consistent decrease is still observed. This indicates that longer exposure times can reduce both power consumption and CPU load. This is likely because longer exposure times require less exposure gain, which in turn reduces the computational load on the CPU during image processing. These results indicate that if hand keypoints can be estimated accurately under lower frame rates and longer exposure time, the saved computational and power resource can be reallocated to other XR applications, an important consideration for resource-constrained XR systems.

Effectiveness of our annotation pipeline. Figure 9 illustrates the accuracy of our annotation pipeline with respect to the number of cameras used. We evaluated the pixel error between the estimated keypoints and the ground-truth keypoints by varying the number of camera views. Since our dataset does not include ground-truth annotations independent from our own pipeline, we employed the publicly available multi-view CMU Panoptic dataset [31] to validate the annotation accuracy. We randomly selected subsets of camera views (up to a maximum of 30) and measured the pixel error against the ground-truth keypoints provided by the CMU dataset. The results indicate that the annotation error stabilizes when using more than 13 cameras, achieving an average pixel error of approximately 3–5 pixels. Given that our system uses 70 cameras, we expect the annotation error in our dataset to be even lower.

6 Conclusion

In this paper, we proposed EgoBlur and EgoBlurNet for 3D hand pose estimation from motion-blurred images in XR environments. Firstly, we introduced EgoBlur, a new first-ever dataset consisting of egocentric, real-world blurred images captured from a prototye HMD XR device during dynamic hand movements. Unlike previous approaches, which mostly relied on synthetically blurred datasets that do not generalize well to real-world blur scenarios, our dataset captures naturally occurring motion blur while providing accurate keypoint annotations with temporal continuity. Secondly, we presented EgoBlurNet, a hand pose estimation model that leverages guidance from sharp representations to improve robustness and accuracy under motion blur. Experimental results demonstrates that our model achieve state-of-the-art performance on our EgoBlur dataset which consists of egocentric, real-world blurred hand images captured using XR device. This highlights the robustness of our approach to real motion blur and its potential impact on real-world XR applications.

References

[1] S. Baek, K. I. Kim, and T.-K. Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1067–1076, 2019. 3

[2] A. Boukhayma, R. d. Bem, and P. H. Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10843–10852, 2019. 3

[3] Z. Cao, I. Radosavovic, A. Kanazawa, and J. Malik. Reconstructing hand-object interactions in the wild. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12417–12426, 2021. 3

[4] L. Chen, S.-Y. Lin, Y. Xie, Y.-Y. Lin, and X. Xie. Mvhm: A large-scale multi-view hand mesh benchmark for accurate 3d hand pose estimation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 836–845, 2021. 3

[5] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. Cascaded pyramid network for multi-person pose estimation. *CoRR*, abs/1711.07319, 2017. 5

[6] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3593–3601, 2016. 1, 3

[7] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Real-time 3d hand pose estimation with 3d convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):956–970, 2018. 3

[8] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Robust 3d hand pose estimation from single depth images using multi-view cnns. *IEEE Transactions on Image Processing*, 27(9):4422–4436, 2018. 3

[9] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10833–10842, 2019. 3

[10] S. Hampali, S. D. Sarkar, M. Rad, and V. Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11090–11100, 2022. 6

[11] S. Han, B. Liu, R. Cabezas, C. D. Twigg, P. Zhang, J. Petkau, T.-H. Yu, C.-J. Tai, M. Akbay, Z. Wang, A. Nitzan, G. Dong, Y. Ye, L. Tao, C. Wan, and R. Wang. Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Trans. Graph.*, 39(4), Aug. 2020. doi: 10.1145/3386569.3392452 1

[12] S. Han, P.-c. Wu, Y. Zhang, B. Liu, L. Zhang, Z. Wang, W. Si, P. Zhang, Y. Cai, T. Hodan, et al. Umetrack: Unified multi-view end-to-end hand tracking for vr. In *SIGGRAPH Asia 2022 conference papers*, pp. 1–9, 2022. 1, 2, 6

[13] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[14] U. Iqbal, P. Molchanov, T. B. J. Gall, and J. Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 118–134, 2018. 3

[15] C. Jiang, Y. Xiao, C. Wu, M. Zhang, J. Zheng, Z. Cao, and J. T. Zhou. A2j-transformer: Anchor-to-joint transformer network for 3d interacting hand pose estimation from a single rgb image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8846–8855, 2023. 3, 6

[16] D. Kulon, R. A. Guler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4990–5000, 2020. 1, 2

[17] Y. LI, J. HUANG, F. TIAN, H.-A. WANG, and G.-Z. DAI. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware*, 1(1):84–112, 2019. doi: 10.3724/SP.J.2096-5796.2018.0006 1

[18] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl. Energy characterization and optimization of image sensing toward continuous mobile vision. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pp. 69–82, 2013. 1

[19] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pp. 548–564. Springer, 2020. 1, 2, 3, 6

[20] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 49–59, 2018. 1, 2, 3

[21] M. Oberweger and V. Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *Proceedings of the IEEE international conference on computer vision Workshops*, pp. 585–594, 2017. 3

[22] Y. Oh, J. Park, J. Kim, G. Moon, and K. M. Lee. Recovering 3d hand mesh sequence from a single blurry image: A new dataset and temporal unfolding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 554–563, 2023. 2, 3, 4, 6

[23] T. Ohkawa, K. He, F. Sener, T. Hodan, L. Tran, and C. Keskin. Assemblyhands: Towards egocentric activity understanding via 3d hand pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12999–13008, 2023. 1, 2

[24] P. Panteleris, I. Oikonomidis, and A. Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 436–445. IEEE, 2018. 3

[25] J. Park, G. Moon, W. Xu, E. Kaseman, T. Shiratori, and K. M. Lee. 3d hand sequence recovery from real blurry images and event stream. In *European Conference on Computer Vision*, pp. 343–359. Springer, 2024. 2, 3

[26] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024. 4

[27] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022. 3

[28] G. Rosh, M. Shankar, P. Kukreja, A. Namdev, and P. P. B. H. Xpose: Towards extreme low light hand pose estimation. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 2838–2848, February 2025. 3, 4

[29] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[30] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1145–1153, 2017. 2

[31] T. Simon, H. Joo, I. A. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. *CoRR*, abs/1704.07809, 2017. 1, 3, 7

[32] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 648–656, 2015. 1

[33] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33, August 2014. 1, 2

[34] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *Advances in neural information processing systems*, 27, 2014. 1

[35] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393, 2013. doi: 10.3390/s130506380 1

[36] Y. Wen, H. Pan, L. Yang, J. Pan, T. Komura, and W. Wang. Hierarchical temporal transformer for 3d hand pose estimation and ac-

tion recognition from egocentric rgb videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21243–21253, 2023. 3

[37] H. Xu, T. Wang, X. Tang, and C.-W. Fu. H2onet: Hand-occlusion-and-orientation-aware network for real-time 3d hand mesh reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17048–17058, 2023. 3

[38] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. 2

[39] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020. 3

[40] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pp. 4903–4911, 2017. 3

[41] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 813–822, 2019. 2