2021 IEEE/CVF International Conference on Computer Vision (ICCV) | 978-1-6654-2812-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCV48922.2021.01216

EventHands: Real-Time Neural 3D Hand Pose Estimation from an Event Stream

Viktor Rudnev[1] Vladislav Golyanik[1] Jiayi Wang[1] Hans-Peter Seidel[1]
Franziska Mueller[2] Mohamed Elgharib[1] Christian Theobalt[1]
[1]MPI for Informatics, SIC [2]Google Inc.

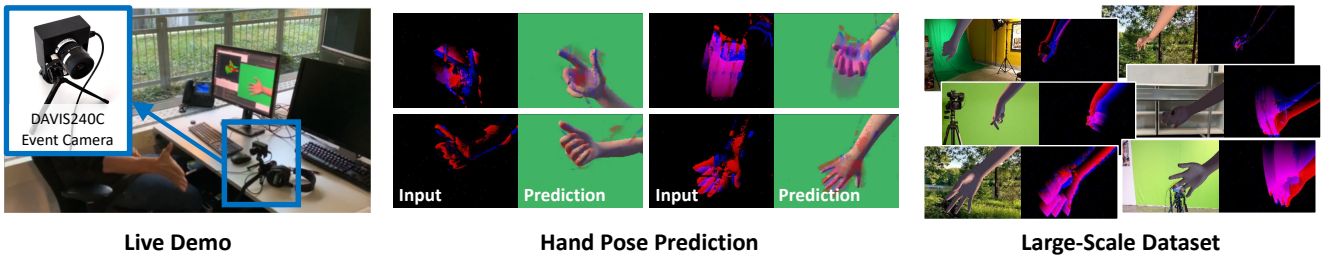


Figure 1. Our *EventHands* approach estimates 3D hand poses from asynchronous event streams in real time (no greyscale or RGB images are used at any step of our method). We built a demo system with a DAVIS240C event camera (left) that runs one order of magnitude faster than image-based prior work on 3D hand reconstruction. *EventHands* leverages our new temporal event representation to reconstruct 3D hands in various challenging poses and moving at previously unseen speed (center). Our approach is trained on a large synthetic dataset (right) created by our new highly efficient GPU-based event camera simulator, but generalises well to real data.

Abstract

3D hand pose estimation from monocular videos is a long-standing and challenging problem, which is now seeing a strong upturn. In this work, we address it for the first time using a single event camera, i.e., an asynchronous vision sensor reacting on brightness changes. Our EventHands *approach has characteristics previously not demonstrated with a single RGB or depth camera such as high temporal resolution at low data throughputs and real-time performance at 1000 Hz. Due to the different data modality of event cameras compared to classical cameras, existing methods cannot be directly applied to and re-trained for event streams. We thus design a new neural approach which accepts a new event stream representation suitable for learning, which is trained on newly-generated synthetic event streams and can generalise to real data. Experiments show that* EventHands *outperforms recent monocular methods using a colour (or depth) camera in terms of accuracy and its ability to capture hand motions of unprecedented speed. Our method, the event stream simulator and the dataset are publicly available (see* https://4dqv.mpi-inf.mpg.de/EventHands/*).*

1. Introduction

Event cameras are vision sensors which respond to events, *i.e.,* local changes in the incoming brightness signals. In contrast to conventional RGB cameras which record images at a pre-defined frequency (*e.g.,* $30-60$ fps), event cameras operate asynchronously, which enables high clock speeds and temporal resolution of up to $1\mu s$ [29]. Due to their unique properties and high-dynamic range, event cameras have already found applications in low-level vision [7, 40, 45, 67], low-latency robotics [54, 15], visual simultaneous localisation and mapping (SLAM) [63, 24], feature and object tracking [2, 34], gesture recognition [3, 60], experimental physics (particle tracking and velocimetry) [9, 61] and astronomy [13, 74], among other fields.

In this work, we are interested in 3D hand pose regression using a single event camera. The high dynamic range, lower latency, and lower throughput of event data are better suited than conventional images for tracking hands, which are often in rapid motion. However, due to the drastically different and less regular data from event cameras, existing RGB- or depth-based methods [10, 68, 31, 36] cannot be directly applied to event streams. A naïve way would be to reconstruct greyscale images from an event stream at arbitrary temporal resolutions first, and then run the same

978-1-6654-2812-5/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCV48922.2021.01216

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:26:53 UTC from IEEE Xplore. Restrictions apply.

monocular methods [6, 10, 68, 71, 41] on those. This policy would, unfortunately, nullify most advantages of event cameras such as low data bandwidth, abstraction from a large variety in textures and illumination conditions and supposedly better generalisation ability. It is also not clear how existing methods would perform on low-resolution greyscale images reconstructed from event streams, as the latter often contain artefacts, and cannot reproduce the exact occurred brightness, due to non-deterministic event thresholds and noise [7, 14, 47]. The primary research question is thus how hands can be reconstructed and tracked in 3D directly from event streams.

We pursue in this paper a learning-based approach and propose the first—to the best of our knowledge—method for 3D reconstruction of a human hand from a single event stream (see Fig. 1). Our neural *EventHands* approach learns to regress 3D hand poses, represented as global rotation and translation as well as pose parameters of a parametric hand model [46], from *locally-normalised event surfaces* (LNES), which is a new way of accumulating events over temporal windows for learning. We generate training data for our neural network using a new high-throughput event stream simulator relying on a parametric hand model [46, 41]. The training data includes variations in hand shape and texture, lighting, and scene background, and accurately mimics the characteristics of a real event camera. Hence, *EventHands* generalises well to real data despite being trained with synthetic data only. Next, *EventHands* runs at 1 KHz, which is significantly faster than any image-based prior works. In summary, our contributions are:

- *EventHands*, *i.e.,* the first approach for 3D hand pose estimation, including rotation and translation in 3D, from a single event stream, running at 1 KHz.
- A new high-throughput event stream simulator supporting a parametric 3D hand model for diverse poses, shapes, and textures, multiple light sources, adjustable event stream properties (*e.g.,* event threshold distributions, noise patterns) and further scene augmentation.
- A live real-time demonstrator of our method running orders of magnitude faster than previous image-based work on a workstation with a single GPU. See our supplementary video for recordings thereof.

We evaluate the proposed approach on a wide variety of motions with real and synthetic data, and provide numerical evidence for our design choices as well as comparisons to prior work. We show that *EventHands* yields accurate estimates even when existing RGB- and depth-based techniques fail due to fast motion.

2. Related Work

We next review related works on 3D hand reconstruction and event-based vision. Our *EventHands* is the first approach for 3D hand pose estimation operating on event streams and has multiple advantages compared to existing RGB- or depth-based methods highlighted in the following.

3D Hand Reconstruction Methods. The vast majority of existing works for 3D hand reconstruction from depth [56, 38, 58, 18, 35, 28, 16] and monocular RGB [73, 11, 53, 36, 65, 52] regresses sparse hand joints. Several recent works also address dense 3D reconstructions of hands [6, 10, 68, 71, 32, 31, 59, 49, 41], some of which rely on a parametric 3D hand model such as MANO [46] for pose and shape or HTML [41] for textures. Hampali *et al.* [20] introduces a new benchmark for hand-object interaction methods. The dataset is then leveraged for 3D hand pose estimation from RGB images by fitting the MANO model to the predicted 2D hand joints. Taylor *et al.* [55] proposed an approach for hand tracking from a new custom-built depth sensor. Their custom depth camera supports 180 fps which is significantly faster than commodity depth cameras (30–60fps) but still far from the temporal resolution of an event camera.

All the aforementioned methods cannot be directly applied to event streams. Even though intensity images and videos can be reconstructed from event streams [14, 47], the obtained greyscale images considerably differ from the data used by existing hand reconstruction techniques and may exhibit domain-specific artefacts. Bridging this domain gap is not straightforward. On the other hand, direct operation on event streams has the advantage of low data bandwidth and abstraction from appearance variation occurring in RGB images.

Event-Based Vision Techniques. Since dynamic vision sensors or event cameras became available , they have been predominantly used for low-level and mid-level problems such as greyscale image restoration from events [40, 67], optical flow [7, 39], or feature detection and tracking [57, 2, 34]. In the context of our work, noteworthy are SLAM methods [63, 62, 66, 24, 70] which rely on sparse rigid 3D reconstruction as an auxiliary task to localise moving robots, event-driven stereo matching [48, 42, 69], and 2D gesture recognition [3, 60]. To apply learning-based method on event streams, suitable representations for the input have been investigated, for example, event frames [44], event count images [33, 72], surfaces of active events (SAE) [8], time-surfaces [27], hierarchy of time surfaces [27], averaged time surfaces [51], sorted time surfaces [1], and differentiable event spike tensors [19], among others. Our LNES representation relates to SAE and differs from it by representing time stamps in window-normalised time units. For a more detailed discussion of event representations, please refer to the survey by Gallego *et al.* [17].

A related work to ours which tracks a human in 3D from a hybrid input of events and greyscale images is EventCap [64]. It relies on event correspondences between greyscale anchor frames and assumes a known rigged and

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:26:53 UTC from IEEE Xplore. Restrictions apply.

skinned human body template. Compared to human bodies, hands exhibit much more self-occlusions, which makes it difficult to obtain event trajectories or perform image-based model fitting. Nehvi *et al.* [37] propose an unsupervised learning approach for deformable object tracking in 3D, which correlates observed and simulated event streams. However, their method requires an accurate initialisation, supports only simple hand motions, and operates far from real time. Instead, we train a neural network to regress challenging 3D hand poses directly from an event representation suitable for learning (LNES), enabling live applications running five orders of magnitude faster compared to [64, 37]. Although there exist general-purpose event camera simulators [22, 25, 43], we develop a new hands-specific simulator for generating training data. This has the advantage that the parametric hand model is tightly integrated into it, enabling on-the-fly sampling of realistic textures, poses and shapes. Moreover, it is tailored for a high data generation speed with seamless GPU support.

All in all, our *EventHands* approach further advances the underexplored area of 3D reconstruction and tracking of non-rigid objects from events.

3. Event Camera Model

While event cameras obey the pinhole camera model of geometric projections from the 3D space to the 2D image plane, each pixel of an event camera independently and asynchronously reacts to differences in the observed logarithmic brightness $\mathcal{L}(u,t)$. An event $e_i = (u_i, t_i, p_i)$ is a 3-tuple with the pixel identifier u_i, triggering time t_i and the binary polarity flag $p_i \in \{-1, 1\}$ signalising whether the logarithmic brightness has increased or decreased by an absolute threshold $|C|$, *i.e.,* an event is triggered at time t_i as soon as one of the following two conditions is satisfied:

$$\begin{cases} \mathcal{L}(u_i,t_i) - \mathcal{L}(u_i,t_p) = C & (p = 1) \\ \mathcal{L}(u_i,t_i) - \mathcal{L}(u_i,t_p) = -C & (p = -1) \end{cases}, \tag{1}$$

where t_p is the previous triggering time of an event at u_i. The event camera we use in our experiments (DAVIS240C) provides the microsecond precision for t_i. Due to the hardware reasons, C is not a fixed threshold but rather follows an unknown distribution of the thresholds χ. In event camera modelling, it is, however, convenient and sufficient to assume C to be equal to the expected value of χ. Moreover, a capacitor attached to each sensor pixel can suddenly overfill, which leads to a spurious noise event registration.

4. Event Stream Simulator and the Dataset

Due to the lack of event stream dataset for hand pose estimation, and the difficulty of obtaining accurate 3D ground-truth annotations on real data, we build a highly efficient event stream simulator to generate a large-scale synthetic event stream dataset with annotations. In total, we generated 100 hours of simulated event data, which provides $3.6 \cdot 10^8$ discrete time steps with ground-truth annotations for training. This places our data among the most extensive event stream datasets available for research purposes so far. We plan to release both the dataset and the simulator.

4.1. Scene Modelling

Hand and Arm Model. Our simulator models both arm and hand geometry, as modelling the hand alone would generate spurious events generated from the seam at the wrist. Hence we use SMPL+H [46], *i.e.,* a model that combines the hand model MANO [46] and the body model SMPL [30]. To capture events generated by the hand appearance, we use the texture model of Qian *et al.* [41] for the hand. The arm texture is obtained by extending the average hand boundary colour to the rest of the SMPL+H mesh.

Model Animation. To simulate hand articulation, we sample individual model poses using the provided MANO PCA-based parameter spaces to obtain a natural distribution of hand poses. Additional random offsets to the translation and pose parameters of SMPL+H are added to account for rigid body transformations of the hand and to increase variations in arm events. To generate plausible motion, we select a new random pose every single simulated second and smoothly interpolate between those poses using a quadratic Bezier curve. The curve's middle control point is also randomly selected every single simulated second. This ensures that every second, there would be a sharp change in the motion direction (*e.g.,* as in hand waving motions).

Lighting Model. We use a Lambertian lighting model with two lights. Suppose that $n \in \mathbb{R}^3$ is the normal vector to the object surface, $l_1, l_2 \in \mathbb{R}^3$ are light directions and $c_1, c_2, c_{\text{ambient}} \in \mathbb{R}^3$ are light colours. Then,

$$\begin{aligned} &\text{light} = \langle n, l_1 \rangle c_1 + \langle n, l_2 \rangle c_2 + c_{\text{ambient}}, \\ &\text{linear colour} = \text{light} \odot \text{albedo}, \end{aligned} \tag{2}$$

where " $\odot$ " denotes element-wise product.

Image Formation. The scene model and the light model are used to form an RGB image $F_i \in [0, 255]^{W \times H \times 3}$ at time t_i. We convert F_i to a log-brightness image $\mathcal{L}(t_i) \in \mathbb{R}^{W \times H}$ using the estimate

$$\mathcal{L}(t_i) = \log(0.2F_i^r + 0.7F_i^g + 0.1F_i^b + \epsilon), \tag{3}$$

where $\epsilon = 1.0$ is added for numerical stability, and F_i^r, F_i^q and F_i^b are red, green and blue image channels, respectively.

4.2. Event Camera Simulation

Event Stream Generation. To simulate events at a time t_i, at each pixel location u_i, we extract log-brightness $\mathcal{L}(u_i, t_p)$. We additionally maintain a memory frame $M \in$

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:26:53 UTC from IEEE Xplore. Restrictions apply.

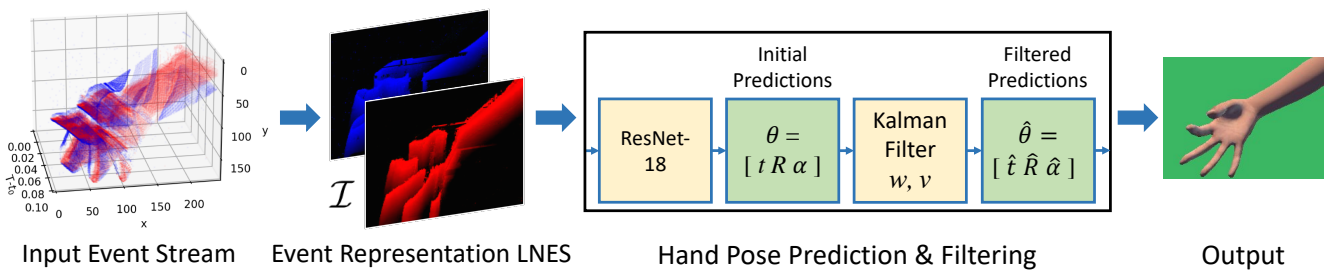


Figure 2. ***EventHands* method overview.** Our approach converts temporal windows of events to the LNES representation with two channels for positive and negative events. The Hand Pose Prediction and Filtering stage uses a neural network (ResNet-18) and a Kalman filter to estimate hand pose as well as the hand translation and rotation. The output shows the rendered hand shape proxy with the estimated parameters. The neural network is trained on our new large-scale dataset for hand pose estimation from event streams.

$\mathbb{R}^{W\times H}$, where $M(u_i) \approx \mathcal{L}(u_i, t_p)$ is the absolute log-brightness of the last generated event at u_i at time t_p. An event tuple (u_i, t_i, p_i) is simulated using the following steps:

1. Noise events: Emit an event tuple with positive or negative polarity with probability p_{positive} and p_{negative}, respectively.
2. Calculate the logarithmic brightness difference $\Delta = \mathcal{L}(u_i, t_i) - M(u_i)$, and
   - 2.1. If $\Delta \geq C$, emit $\lfloor \Delta/C \rfloor$ positive events. Update the memory frame $M(u_i) = M(u_i) + \lfloor \Delta/C \rfloor C$.
   - 2.2. If $\Delta \leq -C$, emit $\lfloor -\Delta/C \rfloor$ negative events. Update $M(u_i) = M(u_i) - \lfloor -\Delta/C \rfloor C$.

The threshold C and the noise event rates $p_{\text{positive}}, p_{\text{negative}}$ are calibrated to match our DAVIS240C event camera. See supplementary material for the details.

Data Augmentation. Augmentation is critical for successful synthetic-to-real domain transfers [5]. Thus, we re-randomise most of the aspects of the simulation every 50 simulated seconds. Those are the hand and body shapes, body position, hand texture, light directions and intensities, background image and its cropped region, as well as C. Please refer to the supplementary document for the randomisation ranges for each variable.

Simulation Runtime. The simulation is capable of rendering and extract events from around 2000 log-brightness images per second. Using temporal resolution of 1000 fps, this allows us to generate ≈100 hours of simulated event data in two days on a single NVIDIA GTX 1070 GPU. Fig. 1 (right) shows sample data synthesised using our simulator.

5. Proposed Approach

In the previous section, we introduced the *EventHands* dataset generated by our new event stream simulator. We now describe our neural approach for 3D hand pose prediction from event streams for which an overview is shown in Fig. 2. We first describe our event stream representation for learning (Sec. 5.1). Next, we elaborate on our method which consists of two stages, *i.e., hand pose prediction* (Sec. 5.2) and *temporal filtering* (Sec. 5.3).

5.1. Our Representation of Events for Learning

The original event stream output of an event camera is an asynchronous and 1D. At the same time, most recent advances in visual machine learning have explored models that work on spatial 2D images, 3D voxel grids or graphs. The straightforward way to convert the 1D event stream to a 2D representation is to accumulate and collapse all events in a time interval, which leads to the loss of temporal resolution within the interval [33]. Hence, we propose a 2D representation called Locally-Normalised Event Surfaces (LNES), which encodes all events within a fixed time window as an image $\mathcal{I} \in \mathbb{R}^{W\times H\times 2}$ (see Fig. 2, left). Using separate channels for positive and negative events preserves the polarities and reduces the number of overridden events. In contrast to existing representations, (*e.g.,* [8]), LNES operates with window-normalised time stamps.

Consider the k-th time window in an event stream of size L. We can create the LNES representation of this window, $\mathcal{I}_k$, by first initializing it with zeros, and collect events $\mathcal{E} = \{(t_i, x_i, y_i, p_i)\}_{i=1}^{N_k}$ which have timestamps t_i within this window. $\mathcal{I}_k$ is updated by iterating through $\mathcal{E}$ from the oldest to the newest event and assigning

$$\mathcal{I}(x_i, y_i, p_i) = \frac{t_i - t_0}{L}. \tag{4}$$

Thus, $\mathcal{I}(x_i, y_i, p_i) \neq 0$ is a window-normalised timestamp of the event which preserves the relative temporal correlation of the events within the window. Note that due to the iteration order $\mathcal{I}(x_i, y_i, p_i)$ can be overridden when a new event with the same polarity is occurring at the same pixel. For our experiments, we used a fixed time length window of 100ms with a 99ms overlap between consecutive windows. Hence, our representation has an effective temporal resolution of 1 ms to match the inference speed of our network.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:26:53 UTC from IEEE Xplore. Restrictions apply.

The proposed event stream representation allows for several augmentations. For example, different contrasts between skin tone and background colour can be simulated by switching the polarity of some events. This can be easily performed in LNES by swapping the content of the two channels at a subset of pixels. We also augment the speed of the motion during training by changing the window length without having to re-generate a dataset with new settings.

Note that in contrast to naïve event accumulation in a time window [44, 33], where the temporal ordering of the events within the window is lost, LNES preserves temporal information of the events which leads to a more expressive input for learning. In addition, this enables our method to run with large window sizes without losing temporal resolution and hence prediction quality. In Sec. 6.2, we provide experimental evidence for the merits of our representation.

5.2. Hand Pose Prediction

We represent the hand pose with $\theta = [t, R, \alpha] \in \mathbb{R}^{12}$, where $\alpha \in \mathbb{R}^6$ are the coefficients of the MANO PCA pose space, and $t, R \in \mathbb{R}^3$ encode the rigid translation in meter and the rotation in axis-angle formulation, respectively. Note that we assume constant lighting and static background relative to the event camera, *i.e.,* all events are due to the hand and arm, up to noise.

We train a ResNet-18 [21] on our event input representation $\mathcal{I}$ to regress the pose representation θ. This architecture allows us to predict $\approx 750-1550$ poses per second depending on the GPU (GTX 1070 *vs* RTX 2080 Ti) to fully take advantage of the millisecond temporal resolution LNES. Please refer to the supplementary document for a more detailed discussion of network architecture choice. During training, we minimise the following loss function $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_\alpha + \lambda_t \mathcal{L}_t + \lambda_R \mathcal{L}_R, \tag{5}$$

with the MANO loss $\mathcal{L}_\alpha$, translation loss $\mathcal{L}_t$, rotation loss $\mathcal{L}_R$ (all ℓ_2 losses) along with the weights $\lambda_t = 500.0$ and $\lambda_R = \frac{1}{3}$. These weights are chosen empirically for normalisation to account for parameters of different magnitudes. We use the most significant MANO components corresponding to the six largest eigenvalues during training and inference. We use 45 hours of synthetically-generated event streams to train our neural network.

5.3. Temporal Filtering

Although our novel input representation explicitly models relative temporal information of the events within the event window, and we use overlapping event windows for pose prediction, sequences of raw network predictions still exhibit temporal jitter due to missing longer-term correlation across event windows. This is especially relevant on real test data where jitter is more severe due to the domain gap (see Sec. 6.2). Hence, we apply additional temporal filtering by using a constant-velocity Kalman filter [23] on the raw network outputs. We set the process noise $W = \omega(0.1)$ and the observation noise $v = 5.0$ for low-speed movements and $W = \omega(3.0)$ and $v = 1.0$ for high-speed movements, with $\omega(\cdot)$ being discrete white noise covariance matrix operator [26]. See our supplement for the exact form of $\omega(\cdot)$.

		synthetic		real
		2D-AUCp	**3D-AUC**	**2D-AUCp**
	no filtering	**0.89**	***0.85***	***0.75***
	no aug.	***0.88***	**0.86**	0.70
EOI	33ms	0.86	***0.85***	0.70
	100ms	0.78	0.80	0.56
ECI-S	33ms	0.83	0.81	0.66
	100ms	0.69	0.76	0.56
ECI	33ms	0.86	0.83	0.69
	100ms	0.76	0.79	0.52
LNES	33ms	***0.88***	***0.85***	0.72
	300ms	0.87	0.84	0.72
	proposed	***0.88***	***0.85***	**0.77**

Table 1. Ablation study on synthetic and real test data. We report the 2D-AUCp and the 3D-AUC (higher values are better, bold/bold italic font denotes best/second-best numbers).

6. Results

We perform experiments on multiple sequences and demonstrate the ability of our approach in capturing a wide variety of motions, including translation, rotation and articulations. *EventHands* is able to accurately reconstruct hands that are moving at speeds previously unseen in the literature. We first introduce our evaluation metrics and test data (Sec. 6.1). Then we present evaluations of our design choices (Sec. 6.2), compare against related techniques [71, 36, 10, 35] (Sec. 6.3) and provide additional results of our method (Sec. 6.4). For more visual results and comparisons, please refer to the supplemental video.

We visualise results on a mean hand shape [46] with a mean texture [41]. Note that our work focuses on predicting the motion of only the hand, and not the arm. For visualisation purposes, we render the arm using the predicted parameters to ensure it attaches to the predicted hand. Nevertheless, this can produce arm movement different from the ground truth. We kindly ask readers to ignore the predicted motion of the arm as it is outside the scope of our work.

6.1. Metrics and Test Data

Synthetic Data. For the synthetic test set, we simulate a total of 1240 seconds of hand motions with $2.64 \cdot 10^8$ events. Ground-truth annotations on all 21 keypoints are available at 1ms intervals.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:26:53 UTC from IEEE Xplore. Restrictions apply.

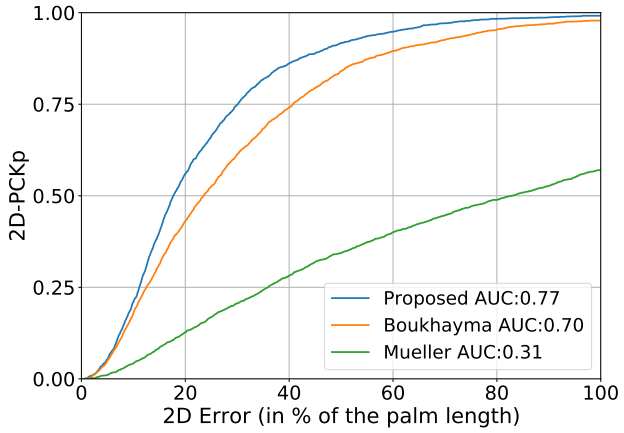


Figure 3. Quantitative results of the RGB-based hand pose estimation methods by Mueller *et al.* [36] and Boukhayma *et al.* [10].

Real Data. We recorded four real event sequences totalling 12 600 milliseconds and $5.93 \cdot 10^6$ events using the DAVIS240C event camera and a single synchronised high-speed RGB camera Sony RX0. Considering each LNES event window sampled at 1ms temporal resolution as a frame, the sequences are annotated evenly at 30 frames per second for a total of 357 frames.

To obtain 2D annotations, we first use OpenPose [12, 50] on the 500 fps high-speed RGB footage to generate initial reference keypoints. The keypoints for fingertips, middle MCP, and wrist are then manually inspected and corrected by multiple annotators to obtain ground-truth annotations. Similarly, hand keypoints are manually annotated on event images from the real event streams and in a second step inspected and corrected by a second set of annotators. In total, we obtained 2499 keypoints over 357 frames. Please refer to the supplemental video for visualizations of the annotation quality.

Evaluation Metrics. When 3D keypoints are available, we evaluate the root-aligned percentage of correct 3D keypoints (3D-PCK) [36] and the area under the PCK curve (3D-AUC) with thresholds ranging from 0 to 100mm.

For real data, we cannot calculate the 3D-PCK since we do not have access to ground-truth 3D annotations and obtaining them manually is challenging. Instead, we report the 2D-PCK and the corresponding area under the curve (2D-AUC). To make the 2D-PCK comparable across different data modalities when comparing to existing RGB methods, we use the wrist and middle finger MCP annotations to calculate the average palm length in pixels for each sequence and normalise the 2D errors by it. Analogously to the 2D body pose estimation literature [4], we refer to the palm-normalised 2D-PCK as *2D-PCKp* and the corresponding AUC as *2D-AUCp*. Here, we use a threshold ranging from 0 to 100% of the relative palm length.

6.2. Ablation Study

We quantitatively evaluate the different design choices of our technique on synthetic and real test data (Table 1) Note that the synthetic test data is an easier setting for our method since it was trained on synthetic event streams. Hence, different versions achieve similar results. The tests on real sequences which shows the benefits of our design choices for generalisation purposes.

Influence of Data Augmentation. As discussed in Sec. 4.2, we use several data augmentation schemes to diversify our event stream data used for training. Augmentation does not help on synthetic data since there is no domain gap to be bridged. On real data, using data augmentation significantly improves the quality of the predictions.

Influence of Temporal Filtering. We use a Kalman filter to improve the longer-term temporal smoothness of our predictions (see Sec. 5.3). On synthetic data, both versions perform similar (Table 1). On real data, however, where temporal jitter is larger due to the domain gap, the proposed filtering improves the results. Since temporal smoothness is best examined in videos, we refer to our supplemental video for visual results of this ablation study.

Influence of Input Event Representation. We compare to three different event representations as baselines: event occurence images (EOI), single-channel event count images (ECI-S) [44], and two-channel event count images (ECI) [33]. EOI and ECI consist of two channels, one for each polarity. EOI contain binary event occurence flags for each pixel whereas ECI contain the accumulated number of all events occured for each pixel in the time window. ECI-S is a simpler version of ECI where all events are accumulated in a single channel irrespective of their polarity. For more details about the baselines, please refer to the supplemental document. In contrast to our LNES, these other event representations do not consider the temporal information of the events. The best window size is task-specific. For an event representation it is hence advantageous to support a wide range of window sizes. Our evaluation shows that LNES captures meaningful and precise information without degrading for longer windows (where more events are condensed). *EventHands* uses an LNES window of 100ms. Using one of the other representations with the same window length performs significantly worse. We also observe that using a shorter temporal window of 33ms leads to similar performance of the baselines and the LNES window. This is expected since less temporal information is lost in shorter baseline windows. However, our LNES representation supports very long windows (300 ms) while the accuracy degrades gracefully. In contrast, the performance of the baselines decays rapidly with increasing window size due to more events being accumulated without any temporal ordering information.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:26:53 UTC from IEEE Xplore. Restrictions apply.

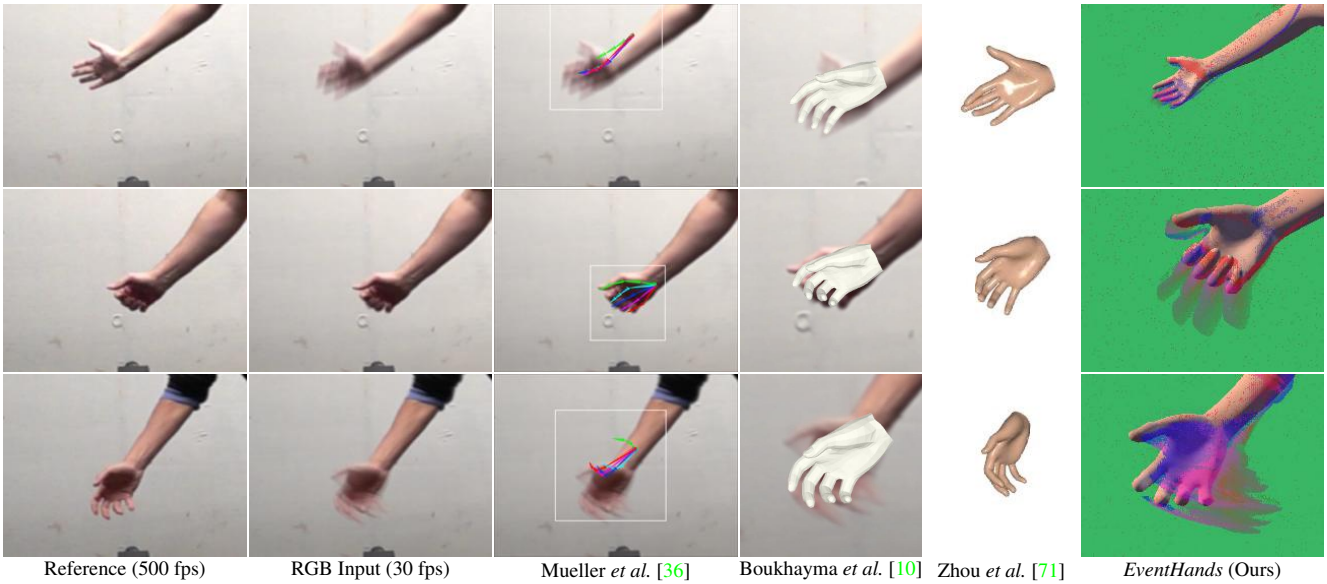


Figure 4. Comparison against state-of-the-art RGB hand pose estimation techniques. We show the high-frame-rate footage (first column) only for reference, while the RGB techniques process a version downsampled to 30fps (second column). Mueller *et al.* [36] estimates wrong bounding boxes and hence produces erroneous network predictions which are propagated to their final IK skeleton fit (shown here). Boukhayma *et al.* [10] estimates wrong rigid rotation on blurry input and often resorts to approximately the MANO mean pose (first and last row). Zhou *et al.* [71] do not estimate any hand translation and hence cannot handle translational motion (first rows). Furthermore, their method struggles with fast blurry motion (last row). Our approach produces accurate 3D hand poses including global translation and rotation, also for challenging articulations like fists (second row) and clearly outperforms state of the art, especially on fast motions.

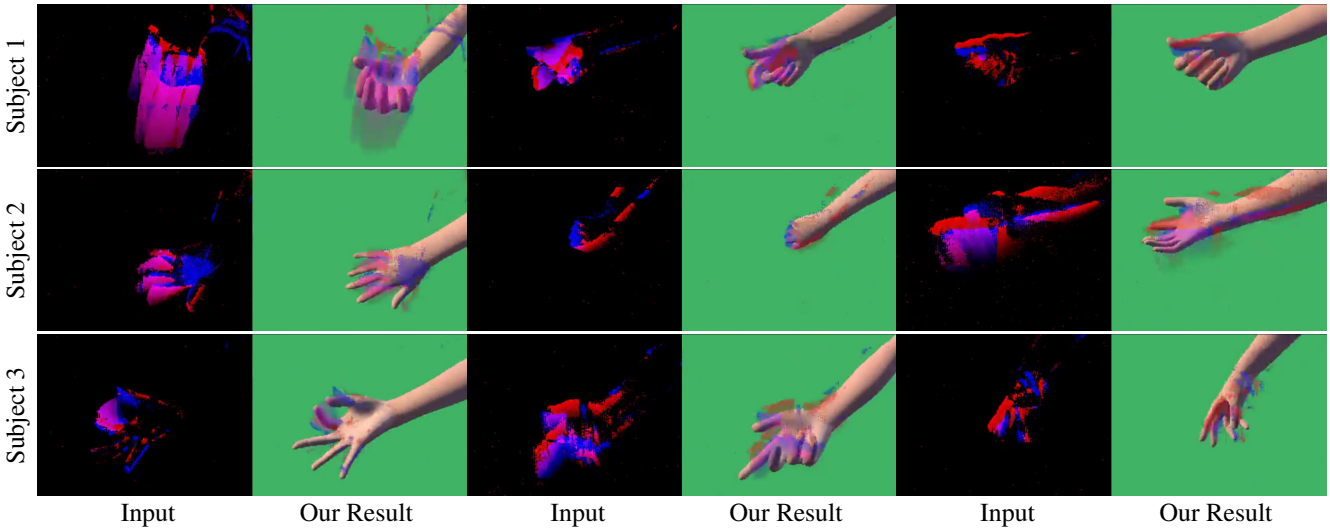


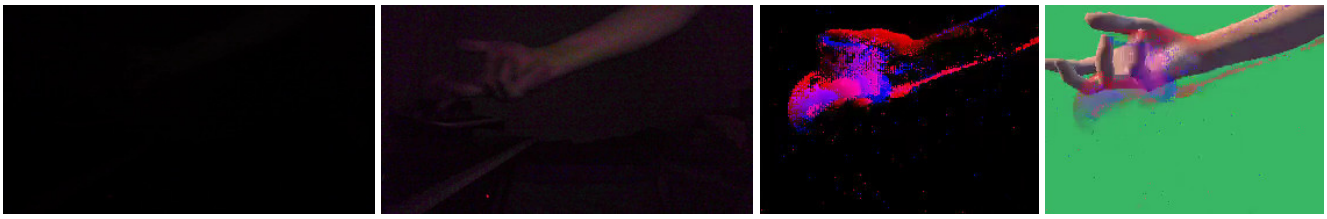
Figure 5. Results of *EventHands* on real data of different subjects. Our technique predicts a wide variety of hand poses under fast motion.

6.3. Comparisons to the State of the Art

We compare *EventHands* to a variety of RGB techniques [71, 36, 10]. Note that data corruptions due to fast motion similarly exist in images produced by commodity depth cameras. Additionally, we show severe failures of a depth-based state-of-the-art method [35] in the supplement.

To obtain the input to the RGB techniques at 30 fps, we apply a moving average filter to the 500 fps images with a window size of 16 frames. Fig. 4 shows qualitative comparisons against different monocular RGB-based hand pose estimation methods. The bounding box estimation of Mueller *et al.* [36] is severely impacted by fast motion since they use simple temporal propagation. Even if the bounding box

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:26:53 UTC from IEEE Xplore. Restrictions apply.



Original RGB (ISO 320) | Brightened RGB | LNES Input | Result

Figure 6. *EventHands* can reconstruct accurate hand poses in the dark, whereas RGB cameras output starkly under-exposed images.

step succeeds, the blurred input image often leads to erroneous predictions which are propagated to the inverse kinematics skeleton fit. The method of Boukhayma *et al.* [10] also struggles on our test sequences. In the presence of motion blur, the estimated rigid rotation is inaccurate, and the articulation often defaults to the MANO mean pose (first and last row). Zhou *et al.* [71] does not predict any hand translation and hence fails in capturing translational movements (first two rows). It also fails to capture fast articulations due to motion blur. Our approach, however, clearly outperforms the RGB-based methods for fast motions. Furthermore, *EventHands* also captures fists (third row) where the other methods fail although there is no blur in the input.

For the quantitative comparison, we use the palm-length-normalised 2D-PCKp to ensure a fair comparison across different data modalities. For Zhou *et al.* [71], we could not calculate any 2D error since they do not provide translation estimates. In Fig. 3, we show that our proposed method performs significantly better than the existing RGB-based methods by Mueller *et al.* [36] and Boukhayma *et al.* [10].

6.4. Additional Results

We evaluate *EventHands* on several real videos and show results in Fig. 5. For each time segment, we show the input event stream and our predicted hand pose overlaid with the input. Our approach handles different subjects performing a wide variety of poses and articulations, under fast motion.

EventHands can also handle slow motions without modifications in the network architecture or retraining. For that, we detect whether our LNES contains sufficient meaningful (*i.e.*, non-noisy) events and fallback to previous predictions in case of insufficient input events. See our supplement for technical details and the video for visualisations.

In contrast to image-based sensors, event cameras monitor relative brightness changes and are able to record reasonable data in dark environments. We show one such example in Fig. 6 and more in our supplemental video. We achieve 0.77 2D-PCKp AUC on 236 frames with 1645 annotations. For more details, please refer to the supplementary document.

7. Discussion

Our *EventHands* assumes that the scene background is approximately static, *i.e.,* although being robust to a certain degree of noise events, there should not be events in the input that are generated from other moving objects in the scene or due to camera movement. Although this means that our method is not explicitly designed to handle hand-object and hand-hand interactions, we observe that it is robust to interactions with small objects (see supplementary video). Future work could investigate how to filter out background events or how to best train a predictor with event data from fully-dynamic scenes. Another interesting avenue for future research would be to combine both RGB and event data in a way to preserve the low-latency and throughput nature of events, while incorporating the information-rich images where it is easier to detect occlusions and interactions.

Our method would additionally benefit from a learned motion prior to integrate the per-frame predictions better. Such statistical temporal model could replace the white-noise assumption of the Kalman filter.

8. Conclusion

We presented *EventHands*, the first method for 3D hand pose estimation from event streams. Our method runs at milestone 1000 Hz and can reconstruct significantly faster hand motions than any previous work, which is shown in our thorough experiments. We believe that the proposed method is also a step forward in general non-rigid 3D reconstruction from event streams, and the presented ideas can be applied in related scenarios and for other types of objects.

Our supplementary material provides additional results of the proposed approach and further details on the Kalman filter, architecture choice and our event stream simulator.

Acknowledgements. This work was supported by the ERC Consolidator Grant 4DRepLy (770784). We thank Jalees Nehvi and Navami Kairanda for help with comparisons.

References

[1] Ignacio Alzugaray and Margarita Chli. Ace: An efficient asynchronous corner tracker for event cameras. In *International Conference on 3D Vision (3DV)*, 2018. 2

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:26:53 UTC from IEEE Xplore. Restrictions apply.

[2] Ignacio Alzugaray and Margarita Chli. Asynchronous corner detection and tracking for event cameras in real time. *In Robotics and Automation Letters (RA-L)*, 3(4):3177–3184, 2018. 1, 2

[3] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 6

[5] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 4

[6] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[7] Patrick Bardow, Andrew J. Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[8] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):407–417, 2014. 2, 4

[9] David Borer, Tobi Delbruck, and T. Rösgen. Three-dimensional particle tracking velocimetry using dynamic vision sensors. *Experiments in Fluids*, 58, 12 2017. 1

[10] Adnane Boukhayma, Rodrigo de Bem, and Philip H.S. Torr. 3d hand shape and pose from images in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5, 6, 7, 8

[11] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[12] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019. 6

[13] Tat-Jun Chin, Samya Bagchi, Anders Eriksson, and Andre van Schaik. Star tracking using an event camera. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 1

[14] Gottfried Graber Christian Reinbacher and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. In *British Machine Vision Conference (BMVC)*, 2016. 2

[15] Davide Falanga, Kevin Kleber, and Davide Scaramuzza. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40), 2020. 1

[16] Linpu Fang, Xingyan Liu, Li Liu, Hang Xu, and Wenxiong Kang. Jgr-p2o: Joint graph reasoning based pixel-to-offset prediction network for 3d hand pose estimation from a single depth image. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[17] Guillermo Gallego, Tobi Delbruck, Garrick Michael Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jorg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[18] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[19] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[20] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[22] Jacques Kaiser, J Camilo Vasquez Tieck, Christian Hubschneider, Peter Wolf, Michael Weber, Michael Hoff, Alexander Friedrich, Konrad Wojtasik, Arne Roennau, Ralf Kohlhaas, et al. Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 127–134, 2016. 3

[23] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960. 5

[24] Hanme Kim, Stefan Leutenegger, and Andrew Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 2

[25] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2149–2154, 2004. 3

[26] Roger Labbe. Kalman and bayesian filters in python. https://elec3004.uqcloud.net/2015/tutes/Kalman_and_Bayesian_Filters_in_Python.pdf, 2014. online, accessed on the 11 Dec. 2020. 5

[27] Xavier Lagorce, G. Orchard, F. Galluppi, B. Shi, and R. Benosman. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39:1346–1359, 2017. 2

[28] Shile Li and Dongheui Lee. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:26:53 UTC from IEEE Xplore. Restrictions apply.

[29] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. 1

[30] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 3

[31] Jameel Malik, Ibrahim Abdelaziz, Ahmed Elhayek, Soshi Shimada, Sk Aziz Ali, Vladislav Golyanik, Christian Theobalt, and Didier Stricker. Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[32] Jameel Malik, Ahmed Elhayek, Fabrizio Nunnari, Kiran Varanasi, Kiarash Tamaddon, Alexis Héloir, and Didier Stricker. Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In *International Conference on 3D Vision (3DV)*, 2018. 2

[33] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 4, 5, 6

[34] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. Event-based moving object detection and tracking. In *International Conference on Intelligent Robots and Systems (IROS)*, 2018. 1, 2

[35] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 5, 7

[36] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 5, 6, 7, 8

[37] Jalees Nehvi, Vladislav Golyanik, Franziska Mueller, Hans-Peter Seidel, Mohamed Elgharib, and Christian Theobalt. Differentiable event stream simulator for non-rigid 3d tracking. In *CVPR Workshop on Event-based Vision*, 2021. 3

[38] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[39] Liyuan Pan, Miaomiao Liu, and Richard Hartley. Single image optical flow estimation with an event camera. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[40] 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 *Computer Vision and Pattern Recognition(CVPR)*, 2019. 1, 2

[41] Neng Qian, Jiayi Wang, Franziska Mueller, Florian Bernard, Vladislav Golyanik, and Christian Theobalt. HTML: A Parametric Hand Texture Model for 3D Hand Reconstruction and Personalization. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 3, 5

[42] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. Emvs: Event-based multi-view stereo—3d reconstruction with an event camera in real-time. *International Journal of Computer Vision (IJCV)*, 126:394–1414, 2018. 2

[43] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on Robot Learning (CoRL)*, pages 969–982, 2018. 3

[44] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In *British Machine Vision Conference (BMVC)*, 2017. 2, 5, 6

[45] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1

[46] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 2, 3, 5

[47] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *Asian Conference on Compututer Vision (ACCV)*, 2018. 2

[48] Stephan Schraml, Ahmed Nabil Belbachir, and Horst Bischof. Event-driven stereo matching for real-time 3d panoramic vision. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[49] Jingjing Shen, Tom Cashman, Qi Ye, Tim Hutton, Toby Sharp, Federica Bogo, Andrew Fitzgibbon, and Jamie Shotton. The phong surface: Efficient 3d model fitting using lifted optimization. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[50] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[51] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[52] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[53] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[54] Rika Sugimoto, Mathias Gehrig, Dario Brescianini, and Davide Scaramuzza. Towards Low-Latency High-Bandwidth Control of Quadrotors using Event Cameras. In *International Conference on Robotics and Automation (ICRA)*, 2020. 1

[55] Jonathan Taylor, Vladimir Tankovich, Danhang Tang, Cem Keskin, David Kim, Philip Davidson, Adarsh Kowdle, and Shahram Izadi. Articulated distance fields for ultra-fast tracking of hands interacting. *ACM Transactions on Graphics (TOG)*, 36(6), 2017. 2

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:26:53 UTC from IEEE Xplore. Restrictions apply.

[56] Jonathan Tompson, Murphy Stein, Yann LeCun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)*, 33, 2014. 2

[57] Valentina Vasco, Arren Glover, and Chiara Bartolozzi. Fast event-based harris corner detection exploiting the advantages of event-driven cameras. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 4144–4149, 2016. 2

[58] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[59] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dual grid net: Hand mesh vertex regression from single depth maps. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[60] Qinyi Wang, Yexin Zhang, Junsong Yuan, and Yilong Lu. Space-time event clouds for gesture recognition: From rgb cameras to event cameras. In *Winter Conference on Applications of Computer Vision (WACV)*, 2019. 1, 2

[61] Yuanhao Wang, Ramzi Idoughi, and Wolfgang Heidrich. Stereo event-based particle tracking velocimetry for 3d fluid flow reconstruction. In *European Conference on Computer Vision (ECCV)*, 2020. 1

[62] David Weikersdorfer, David B. Adrian, Daniel Cremers, and Jorg Conradt. Event-based 3d slam with a depth-augmented dynamic vision sensor. In *International Conference on Robotics and Automation (ICRA)*, 2014. 2

[63] David Weikersdorfer, Raoul Hoffmann, and Jörg Conradt. Simultaneous localization and mapping for event-based vision systems. In *Computer Vision Systems*, pages 133–142, 2013. 1, 2

[64] Lan Xu, Weipeng Xu, Vladislav Golyanik, Marc Habermann, Lu Fang, and Christian Theobalt. Eventcap: Monocular 3d capture of high-speed human motions using an event camera. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3

[65] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. Aligning latent spaces for 3d hand pose estimation. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[66] Wenzhen Yuan and Srikumar Ramalingam. Fast localization and tracking using event sensors. In *International Conference on Robotics and Automation (ICRA)*, 2016. 2

[67] Song Zhang, Yu Zhang, Zhe Jiang, Dongqing Zou, Jimmy Ren, and Bin Zhou. Learning to see in the dark with events. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[68] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[69] Yi Zhou, Guillermo Gallego, Henri Rebecq, Laurent Kneip, Hongdong Li, and Davide Scaramuzza. Semi-dense 3d reconstruction with a stereo event camera. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[70] Yi Zhou, Guillermo Gallego, and Shaojie Shen. Event-based stereo visual odometry. *IEEE Transactions on Robotics (T-RO)*, 2021. 2

[71] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5, 7, 8

[72] Alex Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems (RSS)*, 2018. 2

[73] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[74] Michał Żołnowski, Rafal Reszelewski, Diederik Paul Moeys, Tobi Delbruck, and Krzysztof Kamiński. Observational evaluation of event cameras performance in optical space surveillance. In *ESA NEO and Debris Detection Conference (2019)*, 2019. 1

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:26:53 UTC from IEEE Xplore. Restrictions apply.