This paper has been accepted for publication at the
IEEE International Conference on 3D Vision (3DV), Fukuoka, 2020. ©IEEE

Learning Monocular Dense Depth from Events

Javier Hidalgo-Carrió, Daniel Gehrig and Davide Scaramuzza
Robotics and Perception Group, University of Zurich, Switzerland

Abstract

Event cameras are novel sensors that output brightness changes in the form of a stream of asynchronous "events" instead of intensity frames. Compared to conventional image sensors, they offer significant advantages: high temporal resolution, high dynamic range, no motion blur, and much lower bandwidth. Recently, learning-based approaches have been applied to event-based data, thus unlocking their potential and making significant progress in a variety of tasks, such as monocular depth prediction. Most existing approaches use standard feed-forward architectures to generate network predictions, which do not leverage the temporal consistency presents in the event stream. We propose a recurrent architecture to solve this task and show significant improvement over standard feed-forward methods. In particular, our method generates dense depth predictions using a monocular setup, which has not been shown previously. We pretrain our model using a new dataset containing events and depth maps recorded in the CARLA simulator. We test our method on the Multi Vehicle Stereo Event Camera Dataset (MVSEC). Quantitative experiments show up to 50% improvement in average depth error with respect to previous event-based methods.

Code and dataset are available at:
http://rpg.ifi.uzh.ch/e2depth

1. Introduction

Event cameras, such as the Dynamic Vision Sensor (DVS) [19] or the ATIS [21], are bio-inspired vision sensors with radically different working principles compared to conventional cameras. While standard cameras capture intensity images at a fixed rate, event cameras only report changes of intensity at the pixel level and do this asynchronously at the time they occur. The resulting stream of events encodes the time, location, and sign of the change in brightness. Event cameras possess outstanding properties when compared to standard cameras. They have a very high dynamic range (140 dB versus 60 dB), no motion blur, and high temporal resolution (in the order of microseconds). Event cameras are thus sensors that can provide high-quality visual information even in challenging high-speed scenarios and high dynamic range environments, enabling new application domains for vision-based algorithms. Recently, these sensors have received great interest in various computer vision fields, ranging from computational photography [27, 26, 30, 31][1] to visual odometry [29, 25, 24, 37, 40, 14] and depth prediction [15, 25, 22, 36, 38, 33, 40]. The survey in [7] gives a good overview of the applications for event cameras.

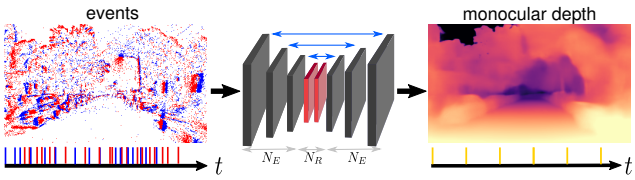


Figure 1: Method overview, the network receives asynchronous events inputs and predicts normalized log depth $\hat{\mathcal{D}}_k$. Our method uses N_R recurrent blocks to leverage the temporal consistency in the events input.

Monocular depth prediction has focused primarily on on standard cameras, which work synchronously, i.e., at a fixed frame rate. State-of-the-art approaches are usually trained and evaluated in common datasets such as KITTI [9], Make3D [3] and NYUv2 [20].

Depth prediction using event cameras has experienced a surge in popularity in recent years [29, 22, 36, 25, 24, 15, 40, 36, 38, 33], due to its potential in robotics and the automotive industry. Event-based depth prediction is the task of predicting the depth of the scene at each pixel in the image plane, and is important for a wide range of applications, such as robotic grasping [17] and autonomous driving, with low-latency obstacle avoidance and high-speed path planning.

However, while event-cameras have appealing properties they also present unique challenges. Due to the working principles of the event camera, they respond predominantly to edges in the scene, making event-based data inherently

[1] https://youtu.be/eomALySSGVU

Method	Density	Monocular	Metric depth	Learning based
[24]	sparse	yes	yes	no
[37]	sparse	yes	yes	no
[15]	semi-dense	yes	no	no
[25]	semi-dense	yes	yes	no
[36]	semi-dense	no	yes	no
[38]	semi-dense	no	yes	no
[40]	semi-dense	yes	yes	yes
[33]	dense	no	yes	yes
Ours	dense	yes	yes	yes

Table 1: A literature review on event-based depth: model-based methods are listed top, learning-based methods are listed bottom. The type of output density is denoted with "sparse" (depth at pixels when only events occurred), "semi-dense" (depth at the reconstructed edges on the image), and "dense" (depth prediction at all pixels). Note than only [33] addresses dense per-pixel depth, but their work uses a stereo setup.

sparse and asynchronous. This makes dense depth estimation with an event camera challenging, especially in low contrast regions, which do not trigger events and, thus, need to be filled in. Prior work in event-based depth estimation has made significant progress in this direction, especially since the advent of deep learning. However, most existing works are limited: they can reliably only predict sparse or semi-dense depth maps [29, 22, 36, 25, 24, 15, 40, 36, 38] or rely on a stereo setup to generate dense depth predictions [33].

In this work, we focus on dense, monocular, and metric depth estimation using an event camera, which addresses the aforementioned limitations. To the best of our knowledge, this is the first time that dense monocular depth is predicted using only events (see Fig. 1). We show that our approach reliably generates dense depth maps overcoming the sparsity in a stream of events. Our methodology is based on learning methods and gives reliable results, setting a baseline for dense depth estimation using events. We release DENSE, a dataset recorded in CARLA, which comprises events, intensity frames, semantic labels, and depth maps. Our contributions are the following:

- A recurrent network that predicts dense per-pixel depth from a monocular event camera.
- The implementation of an event camera plugin in the CARLA [4] simulator.
- DENSE - Depth Estimation oN Synthetic Events: a new dataset with synthetic events and perfect ground truth.
- Evaluation of our method on the popular Multi-Vehicle Stereo Event-Camera (MVSEC) Dataset [39] where we show improved performance with respect to the state of the art.

2. Related Work

2.1. Classical Monocular Depth Estimation

Early work on depth prediction used probabilistic methods and feature-based approaches. The K-means clustering approach was used by Achanta *et al* [2] to generate superpixel methods to improve segmentation and depth. Another work proposed multi-scale features with Markov Random Field (MRF) [1]. These methods tend to suffer in uncontrolled settings, especially when the horizontal alignment condition does not hold.

Deep Learning significantly improved the estimate driven by convolutional neural networks (CNN) with a variety of methods. The standard approach is to collect RGB images with ground truth labels and train a network to predict depth on a logarithmic scale. The network is trained in standard datasets that are captured with a depth sensor such as laser scanning. Eigen *et al* [5] presented the first work training a multi-scale CNN to estimate depth in a supervised fashion. More specifically, the architecture consists of two parts, a first estimation based on Alexnet and a second refinement prediction. Their work led to successively major advances in depth prediction [10, 11, 34, 18, 6]. Better losses such as ordinal regression, multi-scale gradient, and reverse Huber (Berhu) loss were proposed in those works. Another set of approaches is to jointly estimate poses and depth in a self-supervised manner. This is the case of Zhou *et al* [35]. Their work proposes to simultaneously predict both pose and depth with an alignment loss computed from the warped consecutive images. Most of the previous works, except for of [18], are specific for the scenario where they have been trained and, thus, they are not domain independent.

2.2. Event-based Depth Estimation

Early works on event-based depth estimation used multi-view stereo [22] and later Simultaneous Localization and Mapping (SLAM) [25, 29, 37, 15] to build a representation of the environment (i.e.: map) and therefore derivate metric depth. These approaches are model-based methods that jointly calculate pose and map by solving a non-linear optimization problem. Model-based methods can be divided into feature-based methods that produce sparse point clouds and direct methods that generate semi-dense depth maps. Both methods either use the scale given by available camera poses or rely on auxiliary sensors such as inertial measurement units (IMU) to recover metric depth.

Purely vision-based methods have investigated the use of stereo event cameras for depth estimation [36, 38] in which they rely on maximizing a temporal (as opposed to

photometric) consistency between the pair of event camera streams to perform disparity and depth estimation. Recently, several learning-based approaches have emerged that have led to significant improvements in depth estimation [40, 33]. These methods have demonstrated more robust performance since they can integrate several cues from the event stream. Among these, [40] presents a feed-forward neural network that jointly predicts relative camera pose and per-pixel disparities. Training is performed using stereo event camera data, similar to [10], and testing is done using a single input. However, this method still generates semi-dense depth maps, since a mask is applied to generate event frame depths at pixels where an event occurred. The work in [33] overcomes these limitations by fusing data from stereo setup to produce dense metric depth but still relies on a stereo setup and a standard feed-forward architecture. Our work compares to the learning-based approaches but goes one step further by predicting dense metric depth for a single monocular camera. We achieve this by exploiting the temporal consistency of the event stream with a recurrent convolution network architecture and training on synthetic and real data. Table 1 provides a comparison of among state of the art methods, model-based and learning-based, where our proposed approach exceeds by grouping all the listed features.

3. Depth Estimation Approach

Events cameras output events at independent pixels and do this asynchronously. Specifically, their pixels respond to changes in the spatio-temporal log irradiance $L(\mathbf{u}, t)$ that produces a stream of asynchronous events. For an ideal sensor, an event $e_i = (\mathbf{u}_i, t_i, p_i)$ is triggered at time t_i if the brightness change at the pixel $\mathbf{u}_i = (x_i, y_i)^\intercal$ exceeds a threshold of $\pm C$. The event polarity p_i denotes the sign of this change.

Our goal is to predict dense monocular depth from a continuous stream of events. The method works by processing subsequent non-overlapping windows of events $\epsilon_k = \{e_i\}_{i=0}^{N-1}$ each spanning a fixed interval $\Delta T = t_{N-1}^k - t_0^k$. For each window, we predict log depth maps $\{\hat{\mathcal{D}}_k\}$, with $\hat{\mathcal{D}}_k \in [0,1]^{W \times H}$. We implement log depth prediction as a recurrent convolutional neural network with an internal state $\mathbf{s}_k$. We train our network in a supervised manner, using ground truth depth maps. The network is first trained in simulation using perfect ground truth and synthetic events and finetuned in a real sequence.

3.1. Event Representation

Due to the sparse and asynchronous nature of event data, batches of events ϵ_k need to be converted to tensor-like representations $\mathbf{E}_k$. One way to encode these events is by representing them as a spatio-temporal voxel grid [40, 8] with

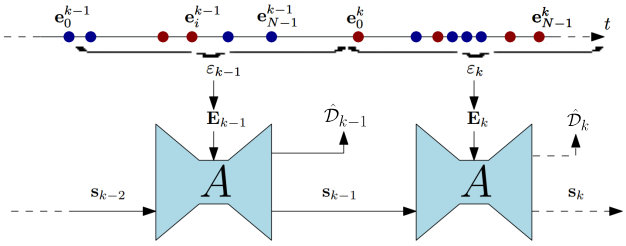


Figure 2: Our network architecture, image adapted from [27]. The event stream is grouped into non-overlapping windows of events and converted to tensor-like voxel grids [40]. These voxel grids are passed to our recurrent fully convolutional neural network to produce normalized log depth predictions.

dimensions $B \times H \times W$. Events within the time window ΔT are collected into B temporal bins according to

$$\mathbf{E}_k(\mathbf{u}_k, t_n) = \sum_{e_i} p_i \delta(\mathbf{u}_i - \mathbf{u}_k) \max(0, 1 - |t_n - t_i^*|) \quad (1)$$

where $t_i^* = \frac{B-1}{\Delta T}(t_i - t_0)$ is the normalized event timestamp. In our experiments, we used $\Delta T = 50ms$ of events and $B = 5$ temporal bins. To facilitate learning, we further normalize the non-zero values in the voxel grid to have zero mean and unit variance.

3.2. Network Architecture

It consists of a recurrent, fully convolutional neural network, based on the UNet architecture [28]. The network input is first processed by a head layer $\mathcal{H}$ and then N_E recurrent encoder layers ($\mathcal{E}^i$) followed by N_R residual blocks ($\mathcal{R}^j$) and N_E decoder layers $\mathcal{D}^l$. A final depth prediction layer $\mathcal{P}$ produces the output of our network. The head layer produces an output with N_b channels, which is doubled at each encoder layer, resulting in a feature map with $N_b \times 2^{N_E}$ output channels. $\mathcal{P}$ performs a depth-wise convolution with one output channel and kernel size 1. We use skip connections between symmetric encoder and decoder layers (see Fig. 2). At the final layer, the activations are squashed by a sigmoid activation function. Each encoder layer is composed of a downsampling convolution with kernel size 5 and stride 2 and a ConvLSTM [32] module with kernel size 3. The encoding layers maintain a state c_k^i which is at 0 for $k = 0$. The residual blocks use a kernel size of 3 and apply summation over the skip connection. Finally, each decoder layer is composed of a bilinear upsampling operation followed by convolution with kernel size 5. We use ReLU except for the prediction layer, and batch normalization [13]. In this work we use $N_E = 3$, $N_R = 2$ and $N_b = 32$ and we unroll the network for $L = 40$ steps.

3.3. Depth Map Post-processing

As usual, in recent work on depth prediction, we train our network to predict a normalized log depth map. Log depth maps have the advantage of representing large depth variations in a compact range, facilitating learning. If $\hat{\mathcal{D}}_k$ is the depth predicted by our model, the metric depth can be recovered by performing the following operations:

$$\hat{\mathcal{D}}_{m,k} = \mathcal{D}_{\text{max}} \exp(-\alpha(1 - \hat{\mathcal{D}}_k)) \tag{2}$$

Where $\mathcal{D}_{\text{max}}$ is the maximum expected depth and α is a parameter chosen, such that a depth value of 0 maps to minimum observed depth. In our case, $\mathcal{D}_{\text{max}} = 80$ meters and $\alpha = 3.7$ corresponding to a minimum depth of 2 meters.

3.4. Training Details

We train our network in a supervised fashion, by minimizing the scale-invariant and multi-scale scale-invariant gradient matching losses at each time step. Given a sequence of ground truth depth maps $\{\mathcal{D}_k\}$, denote the residual $\mathcal{R}_k = \hat{\mathcal{D}}_k - \mathcal{D}_k$. Then the scale-invariant loss is defined as

$$\mathcal{L}_{k,\text{si}} = \frac{1}{n}\sum_{\mathbf{u}}(\mathcal{R}_k(\mathbf{u}))^2 - \frac{1}{n^2}\left(\sum_{\mathbf{u}}\mathcal{R}_k(\mathbf{u})\right)^2, \tag{3}$$

where n is the number of valid ground truth pixels $\mathbf{u}$. The multi-scale scale-invariant gradient matching loss encourages smooth depth changes and enforces sharp depth discontinuities in the depth map prediction. It is computed as follows:

$$\mathcal{L}_{k,\text{grad}} = \frac{1}{n}\sum_{s}\sum_{\mathbf{u}}|\nabla_x \mathcal{R}_k^s(\mathbf{u})| + |\nabla_y \mathcal{R}_k^s(\mathbf{u})|. \tag{4}$$

Here $\mathcal{R}_k^s(\mathbf{x})$ refers to the residual at scale s and the L_1 norm is used to enforce sharp depth discontinuities in the prediction. In this work, we consider four scales, similar to [18]. The resulting loss for a sequence of L depth maps is thus

$$\mathcal{L}_{\text{tot}} = \sum_{k=0}^{L-1}\mathcal{L}_{k,\text{si}} + \lambda\mathcal{L}_{k,\text{grad}}. \tag{5}$$

The hyper-parameter $\lambda = 0.5$ was chosen by cross-validation. We train with a batch size of 20 and a learning rate of 10^{-4} and use the Adam [16] optimizer.

Our network requires training data in the form of events sequences with corresponding depth maps. However, it is difficult to get perfect dense ground truth depth maps in real datasets. For this reason, we propose to first train the network using synthetic data and get the final metric scale by finetuning the network using real events from the MVSEC dataset.

We implement an event camera sensor in CARLA [4] based on the previous event simulator ESIM [23]. The event camera sensor takes the rendered images from the simulator environment and computes per-pixel brightness change to simulate an event camera. The computation is done at a configurable but fixed high framerate (we use 20 times higher than the images frame rate) to approximate the continuous signal of a real event camera. The simulator allows us to capture a variety of scenes with different weather conditions and illumination properties. The camera parameters are set to mimic the event camera at MVSEC with a sensor size of 346×260 pixels (resolution of the DAVIS346B) and a focal length of $83°$ horizontal field of view.

We split DENSE, our new dataset with synthetic events, into five sequences for training, two sequences for validation, and one sequence for testing (a total of eight sequences). Each sequence consists of 1000 samples at 30 FPS (corresponding to 33 seconds), each sample is a tuple of one RGB image, the stream of events between two consecutive images, ground truth depth, and segmentation labels. Note that only the events and the depth, maps are used by the network for training. RGB images are provided for visualization and segmentation labels complete the dataset with richer information. CARLA Towns 01 to 05 are the scenes for training, Town 06 and 07 for validation, and the test sequence is acquired using Town 10. This split results in an overall of 5000 samples for training, 2000 samples for validation, and 1000 samples for testing.

4. Experiments

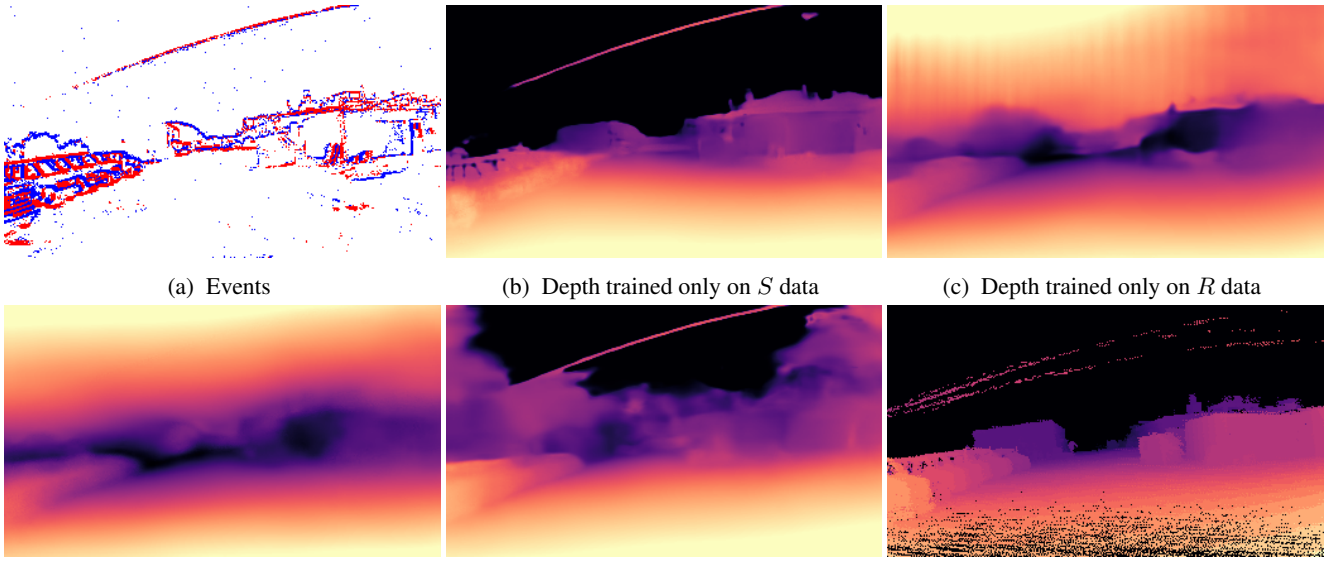
In this section, we present qualitative and quantitative results and compare them with previous methods [40] on the MVSEC dataset. We focus our evaluation on real event data while the evaluation on synthetic data is detailed in Appendix A.

We perform an ablation study by training on synthetic and real events to show the benefits of using synthetic training data from the simulation. We train our recurrent network using the training split of the DENSE dataset (5000 samples) and train for 300 epochs (75000 iterations). We convert the events to a voxel grid synchronized with the ground truth depth which is 30 FPS in simulation. We use each depth image to supervise the training by first converting the absolute depth to a logarithmic scale. A sample is formed by a voxel grid of events between two consecutive frames and the ground truth label. Consecutively, the network is also trained with real data using the MVSEC [2] dataset in the same manner. We unpack the available online data in ROS bag format and pack the stream of events in

[2] https://daniilidis-group.github.io/mvsec/

Training set	Dataset	Abs Rel↓	Sq Rel ↓	RMSE↓	RMSE log↓	SI log↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
S	outdoor day1	0.698	3.602	12.677	0.568	0.277	0.493	0.708	0.808
R		0.450	0.627	9.321	0.514	0.251	0.472	0.711	0.823
S* → R		0.381	**0.464**	9.621	0.473	0.190	0.392	0.719	0.844
S* → (S+R)		**0.346**	0.516	**8.564**	**0.421**	**0.172**	**0.567**	**0.772**	**0.876**
S	outdoor night1	1.933	24.64	19.93	0.912	0.429	0.293	0.472	0.600
R		0.770	3.133	**10.548**	0.638	0.346	0.327	0.582	0.732
S* → R		**0.554**	**1.798**	10.738	**0.622**	**0.343**	0.390	0.598	0.737
S* → (S+R)		0.591	2.121	11.210	0.646	0.374	**0.408**	**0.615**	**0.754**
S	outdoor night2	0.739	3.190	13.361	0.630	0.301	0.361	0.587	0.737
R		0.400	0.554	**8.106**	**0.448**	**0.176**	0.411	0.720	0.866
S* → R		0.367	**0.369**	9.870	0.621	0.279	0.422	0.627	0.745
S* → (S+R)		**0.325**	0.452	9.155	0.515	0.240	**0.510**	**0.723**	**0.840**
S	outdoor night3	0.683	1.956	13.536	0.623	0.299	0.381	0.593	0.736
R		0.343	0.291	**7.668**	**0.410**	**0.157**	0.451	0.753	**0.890**
S* → R		0.339	0.230	9.537	0.606	0.258	0.429	0.644	0.760
S* → (S+R)		**0.277**	**0.226**	8.056	0.424	0.162	**0.541**	**0.761**	**0.890**

Table 2: Ablation study and evaluation of MVSEC. All rows are the same network with the change in the training set. The Training set is denoted with S (synthetic data from the DENSE training split), R (real data from the training split in *outdoor day2* sequence), S^* (first 1000 samples of the DENSE training split), $S^* \rightarrow R$ (pretrained on S^* and retrained on R), $S^* \rightarrow (S+R)$ (pretrained on S^* and retrained on both datasets). ↓ indicates lower is better and ↑ higher is better. The results are the driving sequences of MVSEC (except for *outdoor day2*). Best values are shown in bold.



(a) Events (b) Depth trained only on S data (c) Depth trained only on R data

(d) Depth trained on $S^* \rightarrow R$ (e) Depth trained on $S^* \rightarrow (S+R)$ (f) Depth ground truth

Figure 3: Ablation study of our method trained with different training sets (see Table 2). Fig. 3a shows the events, from Fig. 3b to Fig. 3e the predicted dense monocular depth using different training sets. Fig. 3f depicts the corresponding ground truth. The depth maps are shown in logarithmic scale and correspond to sample 3562 in the *outdoor day1* sequence of MVSEC.

voxel grid synchronized with the ground truth depth which is 20 FPS. We use *outdoor day2* sequence of MVSEC to train the network as in [40]. More specifically we split the sequence into training, validation, and testing. The training split consist of 8523 samples, the validation split contains 1826 samples and the test split comprise the rest with another 1826 samples. We also train for 300 epochs in the real data (127800 iterations). We perform data augmentation, in synthetic and real data, by random cropping and horizontal flip of the training samples. The network and data loader are implemented in Pytorch.

The quantitative results are shown in Table 2, and are

Dataset	Distance	Frame based			Event based				
		MonoDepth [10]	MegaDepth [18]	MegaDepth^{+} [18]	Zhu et al. [40]	Ours^{S}	Ours^{R}	Ours$^{S^{*}\rightarrow R}$	Ours$^{\#}$
outdoor day1	10m	3.44	2.37	3.37	2.72	4.60	2.70	2.13	**1.85**
	20m	7.02	4.06	5.65	3.84	5.66	3.46	2.68	**2.64**
	30m	10.03	5.38	7.29	4.40	6.10	3.84	3.22	**3.13**
outdoor night1	10m	3.49	2.54	**2.40**	3.13	10.36	5.36	3.31	3.38
	20m	6.33	4.15	4.20	4.02	12.97	5.32	**3.73**	3.82
	30m	9.31	5.60	5.80	4.89	13.64	5.40	**4.32**	4.46
outdoor night2	10m	5.15	3.92	3.39	2.19	6.14	2.80	1.99	**1.67**
	20m	7.80	5.78	4.99	3.15	8.64	3.28	3.14	**2.63**
	30m	10.03	7.05	6.22	3.92	9.57	3.74	4.14	**3.58**
outdoor night3	10m	4.67	4.15	4.56	2.86	5.72	2.39	1.76	**1.42**
	20m	8.96	6.00	5.63	4.46	8.29	2.88	2.98	**2.33**
	30m	13.36	7.24	6.51	5.05	9.27	3.39	3.98	**3.18**

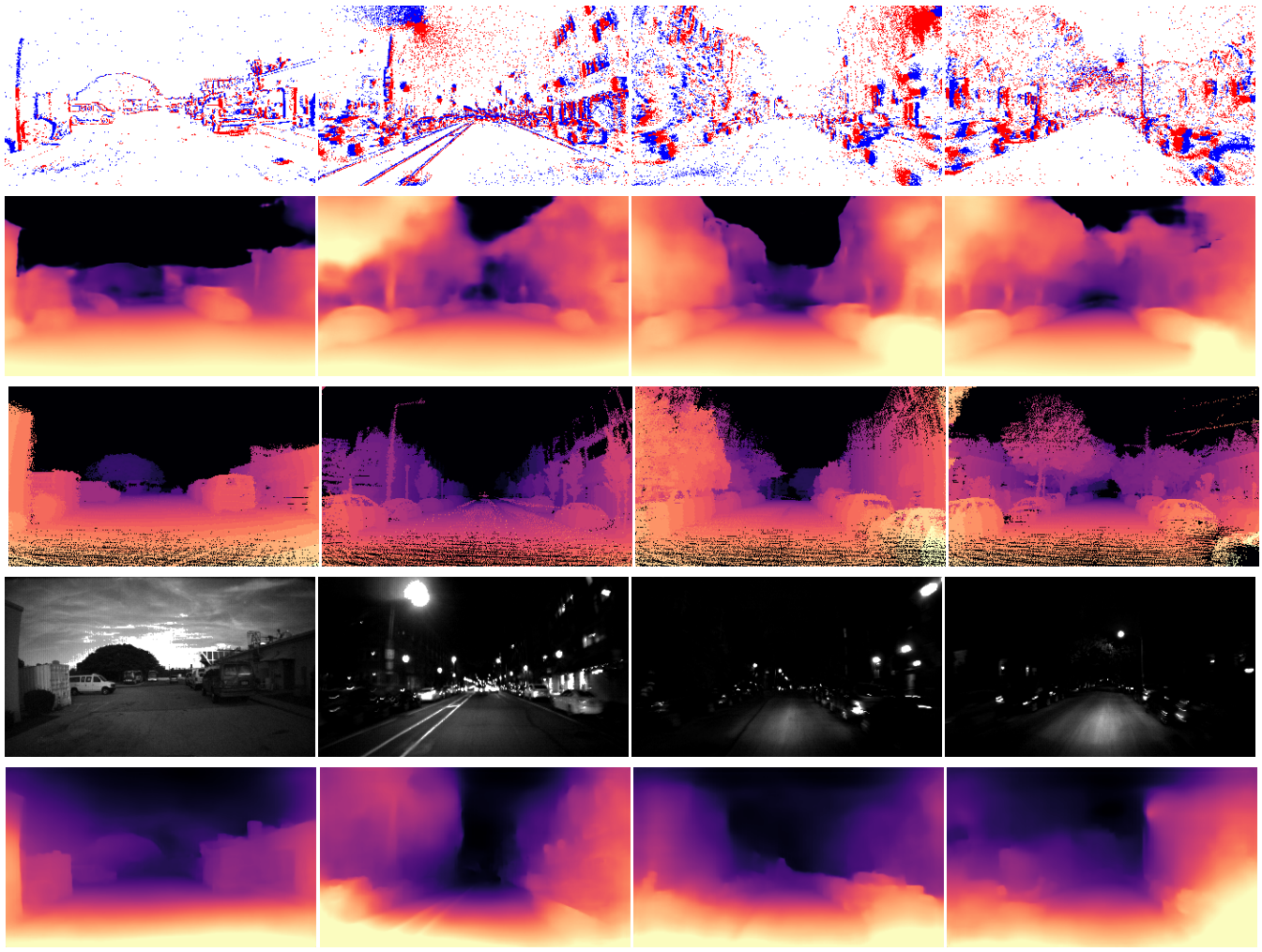
Table 3: Average absolute depth errors (in meters) at different cut-off depth distances (lower is better). MegaDepth^{+} refers to MegaDepth [18] using E2VID [27] reconstructed frames and Ours$^{\#}$ refers to our method trained using $S^{*} \rightarrow (S + R)$. Our results outperform state of the art image-based monocular depth prediction methods [10, 18] while outperforming state of the art event-based methods [40].

supported with qualitative depth maps in Fig 3. The network predicts depth in the logarithmic scale, which is normalized and restored to absolute values by multiplying by the maximum depth clipped at 80 m. We compare the results for different combinations of training sets. We first show the results by training only with synthetic data S. Afterward, we show that training in real data R drastically improve the metrics with respect to only synthetic data S. The incorporation of real events improves all metrics, especially the Absolute relative error - Abs. Rel, which is the most informative value. This is because the network is capable of closing the domain gap when seeing data from the real world. We also notice that pretraining the network with a subset (1000 samples) of the synthetic data S^{*} and then training in real data helps the network to converge faster (first 100 epochs). This is depicted by $S^{*} \rightarrow R$ in Table 2 and the results increase the performance in almost all values with respect to training only with real data R. The penalty of training with real data is in the qualitative depth map (see Fig. 3c and 3d). This is because having perfect align ground truth data with events is hard to obtain in the real world. The lack of perfect ground truth prevents the network to predict depth with sharp edges and have difficulties to mask the sky. For that reason, we mixed synthetic and real data, denoted by $(S + R)$, to fine tune the network and get the best of both. Training with real reduces the errors by predicting the correct metric while using synthetic data helps to estimate qualitatively better depth maps. The combination of synthetic and real training data is possible without changing the loss function since both datasets mimic the DAVIS346B sensor resolution and focal length.

The ablation study shows that synthetic data enhance the results. It also shows that the potential of monocular depth prediction grows with respect to the amount of training data. We now further compare our method against the state of the art monocular depth: two image-based techniques, MonoDepth [3] [10] and MegaDepth [18], and the event-based approach from [40] (see Table 3). MegaDepth is further applied to frames reconstructed from events using E2VID [27]. The evaluation is done using the average mean error at depths of 10m, 20m, and 30m since these are the available metrics reported until now in the MVSEC dataset. The values for MonoDepth are directly taken from the evaluation in [40]. Our work gives more accurate depth prediction at all distances with an average improvement overall sequence of 26.25% at 10m, 25.25% at 20m and 21.0% at 30m with respect to values reported in [40]. Our method produces dense depth results up to 50.0% improvement with respect to previous methods in *outdoor night3* sequence of MVSEC. Image-based methods have difficulties to predict depth in low light conditions. MegaDepth applied to reconstructed frames performs more accurately in night sequences. However, the direct use of events (i.e.: end to end without parsing through image reconstruction) in our method gives a better estimate since the events capture increments in contrast at a higher temporal resolution. Appendix B further explains this fact in a high dynamic range situation. In all the night sequences our method outperforms previous approaches except for *outdoor night1* at 10m. This is because *outdoor night1*, compared to other sequences, has a higher amount of moving objects in front of the car. This creates spurious measurements in the path of such objects which have not been removed from the ground truth depth in the dataset.

None of the methods uses samples from night driving sequences at training time, neither image-based methods

[3] MonoDepth performs more accurately than MonoDepth2 for the MVSEC dataset



(a) outdoor day1 (b) outdoor night1 (c) outdoor night2 (d) outdoor night3

Figure 4: Qualitative comparison of the four test sequences of MVSEC dataset. The first row shows the events. Second row our dense depth map predictions. Third row the ground truth depth maps. The Fourth row shows DAVIS grayscale frames and the fifth row the MegaDepth prediction using the grayscale frames. *outdoor day1* corresponds to sample 3741 in the sequence. *outdoor night1* to sample 288 in the sequence. *outdoor night2* corresponds to sample 2837 in the sequence and *outdoor night3* to sample 2901 in the sequence.

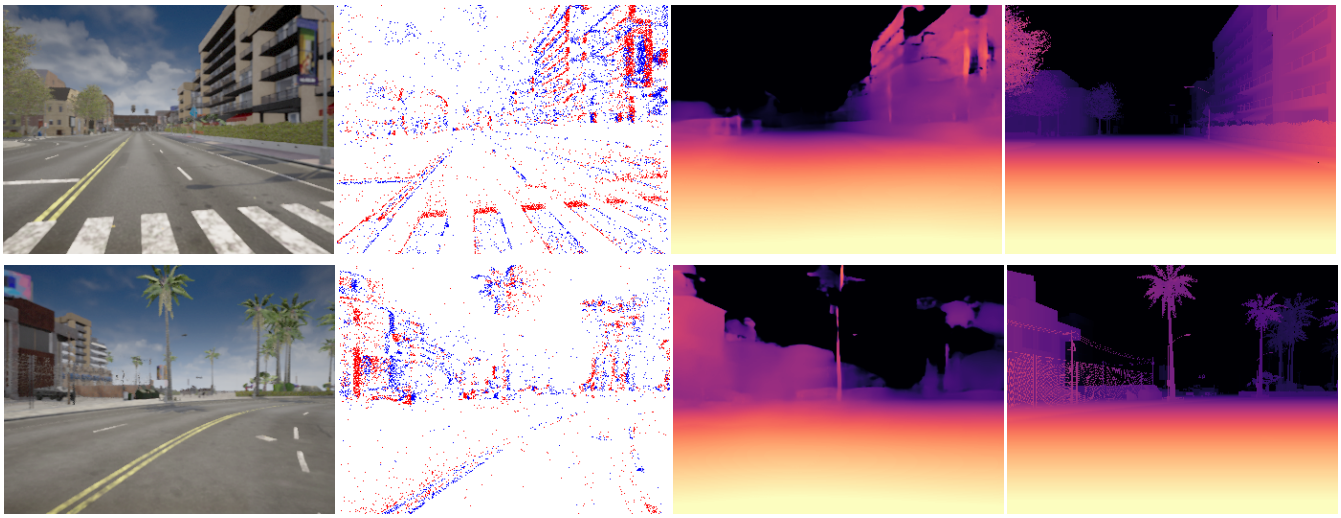
nor event-based solutions. MegaDepth is trained with the MD dataset from images available on the Internet and this achieves superior generalizability than MonoDepth which is trained with KITTI [9] and reported the values from Zhu et al [40]. Fig. 4 contains a visual comparison. Each column corresponds to an MVSEC sequence (Fig. 4a-4d). The first and second row depict DAVIS frames and depth prediction from MegaDepth using standard frames. The third row shows the events warped to a frame at the same synchronized sample. The last two rows are our dense prediction and ground truth. It can be noticed that *outdoor night2* and *outdoor night3* are particularly dark scenes making predictions a challenging task for conventional image-based methods. Our method better preserves the car shapes at both sides of the road, while they are completely neglected by standard images and therefore omitted in the prediction from MegaDepth.

5. Conclusion

In this paper, we presented the first work on monocular dense estimation from events. Our solution exploits the benefits of recurrent convolutional neural networks to infer dense depth from a stream of asynchronous events. We reported results on the Multi Vehicle Stereo Event Camera Dataset (MVSEC) which is the only currently public dataset

Dataset	Abs Rel↓	Sq Rel ↓	RMSE↓	RMSE log↓	SI log↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$	Avg. error 10m↓	Avg. error 20m↓	Avg. error 30m↓
Town06	0.120	0.083	6.640	0.188	0.035	0.855	0.956	0.987	0.31	0.74	1.32
Town07	0.267	0.535	10.182	0.328	0.098	0.774	0.878	0.927	1.03	2.35	3.06
Town10	0.220	0.279	11.812	0.323	0.093	0.724	0.865	0.932	0.61	1.45	2.42

Table 4: Quantitative results on the DENSE dataset. We train the network only on synthetic events from the training split S. The first two sequences are used for validation and the Town10 sequence for testing.



(a) Frame (b) Events (c) Ours on events (d) Ground truth

Figure 5: Qualitative results on DENSE for the Town10 sequence. The first row corresponds to sample 143 and the second row to sample 547 in the sequence.

comprised of events, frames, and ground-truth depth. We showed that training on synthetic data is beneficial for several reasons: it helps the network converge faster, depth maps have a better quality due to perfect ground truth, and simulation captures a larger variety of conditions. Finally, we showed that our methodology produces dense depth maps with more accuracy than existing methods.

6. Acknowledgments

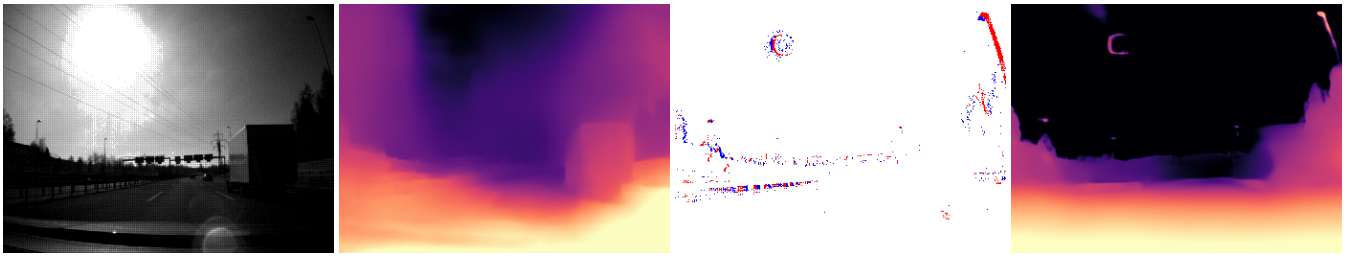
This work was supported by Prophesee, the Swiss National Center of Competence in Research Robotics (NCCR), through the Swiss National Science Foundation, and the SNSF-ERC starting grant. The authors would like to thank Alessio Tonioni (Google Zurich) and Federico Tombari (TU Munich) for their valuable feedback and insights to accomplish this work.

A. Results on Synthetic Data

We show quantitative and qualitative results of our method in the DENSE dataset. Fig 5 shows the qualitative results on the DENSE test sequence corresponding with Town10 in CARLA. We also show the quantitative results for two validation datasets, Town06, and Town07 (see Table 4). The results show metric numbers within the range of state of the art image-based methods in popular datasets like KITTI. This emphasizes our statement that events have enough information to estimate dense monocular depth.

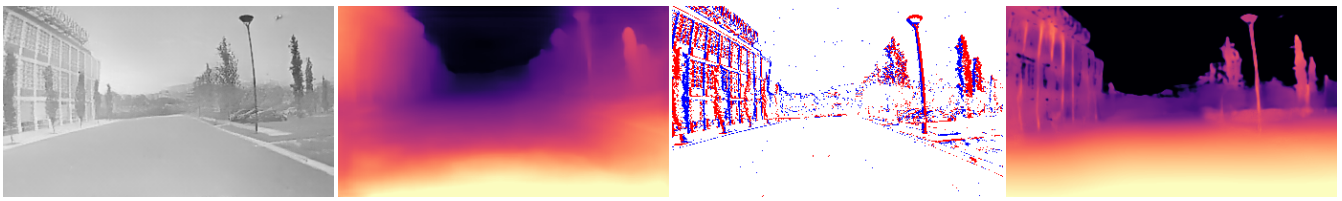
B. Why using events for Depth prediction?

State of the art research in computer vision has demonstrated that salient edges and texture are more relevant than color to estimate depth [12]. Event cameras capture salient edges as well as detailed changes in brightness with high temporal-spatial resolution. This makes event cameras suitable sensors to predict depth. We present two qualitative cases in this Appendix. Fig. 6 shows a comparison by estimating depth from events in an HDR scenario. Our method predicts depth directly from events while MegaDepth uses the greyscale frame. MegaDepth shows difficulties to mask the sky due to overexposure while the prediction from events better approximates the structure of the scene. However, our method has some difficulties to predict the truck on the right side of the image. This is because our DENSE dataset and the *outdoor day2* sequence from MVSEC do not have moving trucks, so the network has never seen such a situation during training. Fig. 7 depicts the case of predicting depth from a reconstructed frame using E2VID [27], which has HDR and does not suffer from motion blur. The reconstruction shows that depth prediction directly from events has more level of detail than from a reconstructed



(a) Frame (b) MegaDepth on frames [18] (c) Events (d) Ours on events

Figure 6: Qualitative results in a high dynamic range (HDR) situation facing the sun when driving a car on a highway. Fig. 6a grayscale frame from a DAVIS camera. Fig. 6b Depth prediction from MegaDepth. Fig. 6c Events and Fig. 6d our depth map prediction using events



(a) E2VID[27] (b) MegaDepth on E2VID (c) Events (d) Ours on events

Figure 7: Qualitative results comparing depth prediction using MegaDepth on reconstructed frames from E2VID and our method using events.

frame.

References

[1] S. H. C. A. Saxena and A. Y. Ng. Learning depth from single monocular images. 2

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 2

[3] M. S. Ashutosh Saxena and A. Y. Ng. Make3d: Learning 3d scene structure from asingle still image. 2009. 1

[4] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Conf. on Robotics Learning (CoRL)*, 2017. 2, 4

[5] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. 2

[6] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 2

[7] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza. Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 1

[8] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Int. Conf. Comput. Vis. (ICCV)*, 2019. 3

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Research*, 32(11):1231–1237, 2013. 1, 7

[10] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017. 2, 3, 6

[11] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3838, 2019. 2

[12] J. Hu, Y. Zhang, and T. Okatani. Visualization of convolutional neural networks for monocular depth estimation, October 2019. 8

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. Int. Conf. Mach. Learning (ICML)*, 2015. 3

[14] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. J. Davison. Simultaneous mosaicing and tracking with an event camera. In *British Mach. Vis. Conf. (BMVC)*, 2014. 1

[15] H. Kim, S. Leutenegger, and A. J. Davison. Real-time 3D reconstruction and 6-DoF tracking with an event camera. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 349–364, 2016. 1, 2

[16] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. *Int. Conf. Learn. Representations (ICLR)*, 2015. 4

[17] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *Int. J. Robot. Research*, 34(4-5):705–724, 2015. 1

[18] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 2, 4, 6, 9

[19] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits*, 43(2):566–576, 2008. 1

[20] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1

[21] C. Posch, D. Matolin, and R. Wohlgenannt. A QVGA 143dB dynamic range asynchronous address-event PWM dynamic image sensor with lossless pixel-level video compression. In *IEEE Intl. Solid-State Circuits Conf. (ISSCC)*, pages 400–401, 2010. 1

[22] H. Rebecq, G. Gallego, and D. Scaramuzza. EMVS: Event-based multi-view stereo. In *British Mach. Vis. Conf. (BMVC)*, 2016. 1, 2

[23] H. Rebecq, D. Gehrig, and D. Scaramuzza. ESIM: an open event camera simulator. In *Conf. on Robotics Learning (CoRL)*, 2018. 4

[24] H. Rebecq, T. Horstschaefer, and D. Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In *British Mach. Vis. Conf. (BMVC)*, 2017. 1, 2

[25] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza. EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real-time. *IEEE Robot. Autom. Lett.*, 2(2):593–600, 2017. 1, 2

[26] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019. 1

[27] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019. 1, 3, 6, 8, 9

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. 3

[29] A. Rosinol Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza. Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios. *IEEE Robot. Autom. Lett.*, 3(2):994–1001, Apr. 2018. 1, 2

[30] C. Scheerlinck, N. Barnes, and R. Mahony. Continuous-time intensity estimation using event cameras. In *Asian Conf. Comput. Vis. (ACCV)*, 2018. 1

[31] C. Scheerlinck, H. Rebecq, D. Gehrig, N. Barnes, R. Mahony, and D. Scaramuzza. Fast image reconstruction with an event camera. In *IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2020. 1

[32] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Conf. Neural Inf. Process. Syst. (NIPS)*, 2015. 3

[33] S. Tulyakov, F. Fleuret, M. Kiefel, P. Gehler, and M. Hirsch. Learning an event sequence embedding for dense event-based deep stereo. In *Int. Conf. Comput. Vis. (ICCV)*, 2019. 1, 2, 3

[34] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018. 2

[35] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017. 2

[36] Y. Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, and D. Scaramuzza. Semi-dense 3D reconstruction with a stereo event camera. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 242–258, 2018. 1, 2

[37] A. Z. Zhu, N. Atanasov, and K. Daniilidis. Event-based visual inertial odometry. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 5816–5824, 2017. 1, 2

[38] A. Z. Zhu, Y. Chen, and K. Daniilidis. Realtime time synchronized event-based stereo. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 438–452, 2018. 1, 2

[39] A. Z. Zhu, D. Thakur, T. Ozaslan, B. Pfrommer, V. Kumar, and K. Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robot. Autom. Lett.*, 3(3):2032–2039, July 2018. 2

[40] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019. 1, 2, 3, 4, 5, 6, 7