2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Unsupervised Event-based Learning of Optical Flow, Depth, and Egomotion

Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, Kostas Daniilidis
University of Pennsylvania
{alexzhu, lzyuan, chaneyk, kostas}@seas.upenn.edu

Abstract

In this work, we propose a novel framework for unsupervised learning for event cameras that learns motion information from only the event stream. In particular, we propose an input representation of the events in the form of a discretized volume that maintains the temporal distribution of the events, which we pass through a neural network to predict the motion of the events. This motion is used to attempt to remove any motion blur in the event image. We then propose a loss function applied to the motion compensated event image that measures the motion blur in this image. We train two networks with this framework, one to predict optical flow, and one to predict egomotion and depths, and evaluate these networks on the Multi Vehicle Stereo Event Camera dataset, along with qualitative results from a variety of different scenes.

1. Introduction

Event cameras are a neuromorphically inspired, asynchronous sensing modality, that detect changes in log light intensity. When a change is detected in a pixel, the camera immediately returns an event, $e = \{x, y, t, p\}$, consisting of the position of the pixel, x, y, timestamp of the change, t, accurate to microseconds, and the polarity of the change, p, corresponding to whether the pixel became brighter or darker. The asynchronous nature of the camera, and the tracking in the log image space, provide numerous benefits over traditional frame based cameras, such as extremely low latency for tracking very fast motions, very high dynamic range, and significantly lower power consumption.

However, the novel output of the cameras provide new challenges in algorithm development. As the events simply reflect whether a change has occurred at a given pixel, a model of photoconsistency, as used traditional motion estimation tasks such as optical flow or structure from motion (SFM), applied directly on the events is no longer valid. As a result, there has been a significant research drive to develop new algorithms for event cameras to solve these traditional robotics problems.

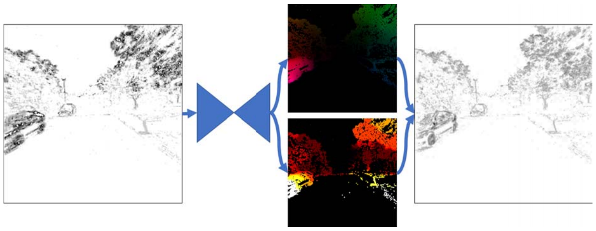

Figure 1: Our network learns to predict motion from motion blur by predicting optical flow (top) or egomotion and depth (bottom) from a set of input, blurry, events from an event camera (left), and minimizing the amount of motion blur after deblurring with the predicted motion to produce the deblurred image (right). Best viewed in color.

There have been recent works by Zhu et al. [24] and Ye et al. [20] that train neural networks to learn to estimate these motion tasks in a self and unsupervised manner. These networks abstract away the difficult problem of modeling and algorithm development. However, both works still rely on photoconsistency based principles, applied to the grayscale image and an event image respectively, and, as a result, the former work relies on the presence of grayscale images, while the latter's photoconsistency assumption may not hold valid in very blurry scenes. In addition, both works take inputs that attempt to summarize the event data, and as a result lose temporal information.

In this work, we resolve these deficiencies by proposing a novel input representation that captures the full spatiotemporal distribution of the events, and a novel set of unsupervised loss functions that allows for efficient learning of motion information from only the event stream. Our input representation, a discretized event volume, discretizes the time domain, and then accumulates events in a linearly weighted fashion similar to interpolation. This representation encodes the distribution of all of the events within the spatiotemporal domain. We train two networks to predict optical flow and ego-motion and depth, and use the predictions to attempt to remove the motion blur generated when

Associated video: https://youtu.be/cdcg-CdV7TU.

978-1-7281-3293-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CVPR.2019.00108

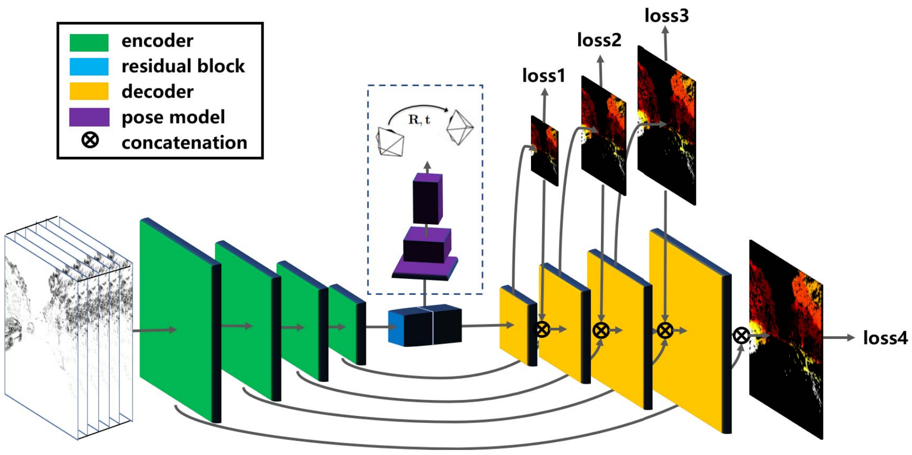

Figure 2: Network architecture for both the optical flow and egomotion and depth networks. In the optical flow network, only the encoder-decoder section is used, while in the egomotion and depth network, the encoder-decoder is used to predict depth, while the pose model predicts the egomotion. At training time, the loss is applied at each stage of the decoder, before being concatenated into the next stage of the network.

the events are projected into the 2D image plane, as visualized in Fig. 1. Our unsupervised loss then measures the amount of motion blur in the corrected event image, which provides a training signal to the network. In addition, our deblurred event images are comparable to edge maps, and so we apply a stereo loss on the census transform of these images to allow our network to learn metric poses and depths.

We evaluate both methods on the Multi Vehicle Stereo Event Camera dataset [26][24], and compare against the equivalent grayscale based methods, as well as the prior state of the art by [24].

Our contributions can be summarized as:

- A novel discretized event volume representation for passing events into a neural network.
- A novel application of a motion blur based loss function that allows for unsupervised learning of motion information from events only.
- A novel stereo similarity loss applied on the census transform of a pair of deblurred event images.
- Quantitative evaluations on the Multi Vehicle Stereo Event Camera dataset [26], with qualitative and quantitative evaluations from a variety of night time and other challenging scenes.

2. Related Work

Since the introduction of event cameras, such as Lichtsteiner et al. [10], there has been a strong interest in the development of algorithms that leverage the benefits provided by these cameras. In the work of optical flow, Benosman et al. [2] showed that normal flow can be estimated by fitting a plane to the events in x-y-t space. Bardow et al. [1] show that flow estimation can be written as a convex optimization problem that solves for the image intensity and flow jointly.

In the space of SFM and visual odometry, Kim et al. [9] demonstrate that a Kalman filter can reconstruct the pose of the camera and a local map. Rebecq et al. [15] similarly build a 3D map, which they localize from using the events. Zhu et al. [25] use an EM based feature tracking method to perform visual-inertial odometry, while Rebecq et al. [16] use motion compensation to deblur the event image, and run standard image based feature tracking to perform visual-inertial odometry.

For model-free methods, self-supervised and unsupervised learning have allowed deep networks to learn motion and the structure of a scene, using only well established geometric principles. Yu et al. [8] established that a network can learn optical flow from brightness constancy with a smoothness prior, while Meister et al. [12] extend this work by applying a bidirectional census loss to improve the quality of the flow. In a similar fashion, Zhou et al. [23] show that a network can learn a camera's egomotion and depth using camera reprojection and a photoconsistency loss. Zhan et al. [22] and Vijayanarasimhan et al. [18] add in a stereo constraint, allowing the network to learn absolute scale, while Wang et al. [19] apply this concept with a recurrent neural network.

Recently, there have been several works, such as [4, 5, 13, 25, **?**] that have shown that optical flow, and other types of motion information, can be estimated from a spatiotemporal volume of events, by propagating the events along the

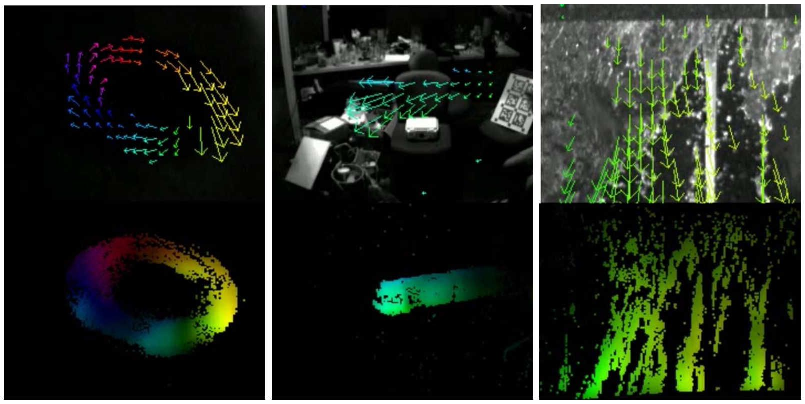

Figure 3: Our flow network is able to generalize to a variety of challenging scenes. Top images are a subset of flow vectors plotted on top of the grayscale image from the DAVIS camera, bottom images are the dense flow output of the network at pixels with events, colored by the direction of the flow. Left to right: Fidget spinner spinning at 13 rad/s in a very dark environment. Ball thrown quickly in front of the camera (the grayscale image does not pick up the ball at all). Water flowing outdoors.

optical flow direction, and attempting to minimize the motion blur in the event image. This concept of motion blur as a loss can be seen as an analogy to the photometric error in frames, as applied to events. In this work, we adapt a novel formulation of this loss from Mitrokhin et al. [13] for a neural network, by generating a single fully differentiable loss function that allows our networks to learn optical flow and structure from motion in an unsupervised manner.

3. Method

Our pipeline consists of a novel volumetric representation of the events, which we describe in Sec. 3.1, which is passed through a fully convolutional neural network to predict flow and/or egomotion and depth. We then use the predicted motion to try to deblur the events, and apply a loss that minimizes the amount of blur in the deblurred image, as described in Sec. 3.2. This loss can be directly applied to our optical flow network, Sec. 3.3. For the egomotion and depth network, we describe the conversion to optical flow in Sec. 3.4.1, as well as a novel stereo disparity loss in Sec. 3.4.2. Our architecture is summarized in Fig. 2.

3.1. Input: The Discretized Event Volume

Selecting the appropriate input representation of a set of events for a neural network is still a challenging problem. Prior works such as Moeys et al. [14] and Maqueda et al. [11] generate an event image by summing the number of events at each pixel. However, this discards the rich temporal information in the events, and is susceptible to motion blur. Zhu et al. [24] and Ye et al. [20] propose image representations of the events, that summarize the number of events at each pixel, as well as the last timestamp and average timestamp at each pixel, respectively. Both works show that this is sufficient for a network to predict accurate optical flow. While this maintains some of the temporal information, a lot of information is still lost by summarizing the high resolution temporal information in the events.

We propose a novel input representation generated by discretizing the time domain. In order to improve the resolution along the temporal domain beyond the number of bins, we insert events into this volume using a linearly weighted accumulation similar to bilinear interpolation.

Given a set of N input events $\{(x_i, y_i, t_i, p_i)\}_{i\in[1,N]}$, and a set B bins to discretize the time dimension, we scale the timestamps to the range $[0, B-1]$, and generate the event volume as follows:

$$t_i^* = (B-1)(t_i - t_1)/(t_N - t_1) \quad (1)$$

$$V(x, y, t) = \sum_i p_i k_b(x - x_i) k_b(y - y_i) k_b(t - t_i^*) \quad (2)$$

$$k_b(a) = \max(0, 1 - |a|) \quad (3)$$

where $k_b(a)$ is equivalent to the bilinear sampling kernel defined in Jaderberg et al. [7]. Note that the interpolation in the x and y dimensions is necessary when camera undistortion or rectification is performed, resulting in non integer pixel positions. In the case where no events overlap between pixels, this representation allows us to reconstruct the exact set of events. When multiple events overlap on a voxel, the summation does cause some information to be lost, but the resulting volume retains the distribution of the events across the spatiotemporal dimensions within the window.

In this work, we treat the time domain as channels in a traditional 2D image, and perform 2D convolution across the x, y spatial dimensions. We found negligible performance increases when using 3D convolutions, for a significant increase in processing time.

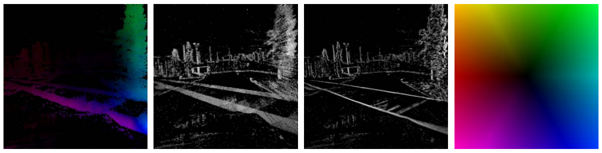

Figure 4: Our network learns to predict motion from motion blur by predicting optical flow or egomotion and depth (1) from a set of input, blurry, events (2), and minimizing the amount of motion blur after deblurring with the predicted motion to produce the deblurred image (3). The color of the flow indicates direction, as draw in the colorwheel (4).

3.2. Supervision through Motion Compensation

As event cameras register changes in log intensity, the standard model of photoconsistency does not directly apply onto the events. Instead, several works have applied the concept of motion compensation, as described in Rebecq et al. [16], as a proxy for photoconsistency when estimating motion from a set of events. The goal of motion compensation is to use the motion model of each event to deblur the event image, as visualized in Fig. 4.

For the most general case of per pixel optical flow, $u(x, y), v(x, y)$, we can propagate the events, $\{(x_i, y_i, t_i, p_i)\}_{i=1,\dots,N}$, to a single time t':

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix} + (t' - t_i) \begin{pmatrix} u(x_i, y_i) \\ v(x_i, y_i) \end{pmatrix} \tag{4}$$

If the input flow is correct, this reverses the motion in the events, and removes the motion blur, while for an incorrect flow, this will likely induce further motion blur.

We use a measure of the quality of this deblurring effect as the main supervision for our network. Gallego et al. [4] proposed using the image variance on an image generated by the propagated events. However, we found that the network would easily overfit to this loss, by predicting flow values that push all events within each region of the image to a line. This effect is discussed further in the supplemental. Instead, we adopt the loss function described by Mitrokhin et al. [13], who use a loss which minimizes the sum of squares of the average timestamp at each pixel.

However, the previously proposed loss function is non-differentiable, as the timestamps were rounded to generate an image. To resolve this, we replace the rounding with bilinear interpolation. We apply the loss by first separating the events by polarity and generating an image of the average timestamp at each pixel for each polarity, T_+, T_-:

$$T_{p'}(x, y|t') = \frac{\sum_i \mathbb{1}(p_i = p') k_b(x - x'_i) k_b(y - y'_i) t_i}{\sum_i \mathbb{1}(p_i = p') k_b(x - x'_i) k_b(y - y'_i) + \epsilon} \tag{5}$$

$$p' \in \{+, -\}, \epsilon \approx 0$$

The loss is, then, the sum of the two images squared.

$$\mathcal{L}_{\text{time}}(t') = \sum_x \sum_y T_+(x, y|t')^2 + T_-(x, y|t')^2 \tag{6}$$

However, using a single t' for this loss poses a scaling problem. In (4), the output flows, u, v, are scaled by $(t' - t_i)$. During backpropagation, this will weight the gradient over events with timestamps further from t' higher, while events with timestamps very close to t' are essentially ignored. To mitigate this scaling, we compute the loss both backwards and forwards, with $t' = t_1$ and $t' = t_N$:

$$\mathcal{L}_{\text{time}} = \mathcal{L}_{\text{time}}(t_1) + \mathcal{L}_{\text{time}}(t_N) \tag{7}$$

Note that changing the target time, t', does not change the timestamps used in (5).

This loss function is similar to that of Benosman et al. [2], who model the events with a function Σ_{e_i}, such that $\Sigma_{e_i}(\mathbf{x}_i) = t_i$. In their work, they assume that the function is locally linear, and solve the minimization problem by fitting a plane to a small spatiotemporal window of events. We can see that the gradient of the average timestamp image, $(dt/dx, dt/dy)$, corresponds to the inverse of the flow, if we assume that all events at each pixel have the same flow.

3.3. Optical Flow Prediction Network

Using the input representation and loss described in Sec. 3.1 and 3.2, we train a neural network to predict optical flow. We use an encoder-decoder style network, as in [24]. The network outputs flow values in units of pixels/bin, which we apply to (4), and eventually compute (9).

Our flow network uses the temporal loss in (7), combined with a local smoothness regularization:

$$\mathcal{L}_{\text{smooth}} = \sum_{\vec{x}} \sum_{\vec{y} \in \mathcal{N}(\vec{x})} \rho(u(\vec{x}) - u(\vec{y})) + \rho(v(\vec{x}) - v(\vec{y})) \tag{8}$$

where $\rho(x) = \sqrt{x^2 + \epsilon^2}$ is the Charbonnier loss function [3], and $\mathcal{N}(x, y)$ is the 4-connected neighborhood around (x, y).

The total loss for the flow network is:

$$\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{time}} + \lambda_1 \mathcal{L}_{\text{smooth}} \tag{9}$$

3.4. Egomotion and Depth Prediction Network

We train a second network to predict the egomotion of the camera and the structure of the scene, in a similar manner to [22, 18]. Given a pair of time synchronized discretized event volumes from a stereo pair, we pass each volume into our network separately, but use both at training time to apply a stereo disparity loss, allowing our network to learn metric scale. We apply a temporal timestamp loss

	outdoor day1		indoor flying1		indoor flying2		indoor flying3	
dt=1 frame	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier
Ours	**0.32**	**0.0**	0.58	**0.0**	1.02	4.0	0.87	3.0
EV-FlowNet	0.49	0.2	1.03	2.2	1.72	15.1	1.53	11.9
UnFlow	0.97	1.6	**0.50**	0.1	**0.70**	**1.0**	**0.55**	**0.0**
	outdoor day1		indoor flying1		indoor flying2		indoor flying3	
dt=4 frames	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier
Ours	1.30	9.7	**2.18**	**24.2**	**3.85**	46.8	3.18	47.8
EV-FlowNet	**1.23**	**7.3**	2.25	24.7	4.05	**45.3**	3.45	39.7
UnFlow	2.95	40.0	3.81	56.1	6.22	79.5	**1.96**	**18.2**

Table 1: Quantitative evaluation of our optical flow network compared to EV-FlowNet and UnFlow. For each sequence, Average Endpoint Error (AEE) is computed in pixels, % Outlier is computed as the percent of points with AEE $>$ 3 pix. dt=1 is computed with a time window between two successive grayscale frames, dt=4 is between four grayscale frames.

defined in Sec. 3.2, and a robust similarity loss between the census transforms [21, 17] of the deblurred event images.

The network predicts Euler angles, (ψ, β, ϕ), a translation, T, and the disparity of each pixel, d_i. The disparities are generated using the same encoder-decoder architecture as in the flow network, except that the final activation function is a sigmoid, scaled by the image width. The pose shares the encoder network with the disparity, and is generated by strided convolutions which reduce the spatial dimension from 16×16 to 1×1 with 6 channels.

3.4.1 Temporal Reprojection Loss

Given the network output, the intrinsics of the camera, K, and the baseline between the two cameras, b, the optical flow, (u_i, v_i) of each event at pixel location (x_i, y_i) is:

$$\begin{pmatrix} x_i^* \\ y_i^* \end{pmatrix} = K\pi\left(R\frac{fb}{d_i}K^{-1}\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} + T\right) \tag{10}$$

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \frac{1}{B-1}\left(\begin{pmatrix} x_i^* \\ y_i^* \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \end{pmatrix}\right) \tag{11}$$

where f is the focal length of the camera, R is the rotation matrix corresponding to (ψ, β, ϕ) and π is the projection function: $\pi\left(\begin{pmatrix} X & Y & Z \end{pmatrix}^T\right) = \begin{pmatrix} \frac{X}{Z} & \frac{Y}{Z} \end{pmatrix}^T$. Note that, as the network only sees the discretized volume at the input, it does not know the size of the time window. As a result, the optical flow we compute is in terms of pixels/bin, where B is the number of bins used to generate the input volume. The optical flow is then inserted into (4) for the loss.

3.4.2 Stereo Disparity Loss

From the optical flow, we can deblur the events from the left and right camera using (4), and generate a pair of event images, corresponding to the number of events at each pixel after deblurring. Given correct flow, these images represent the edge maps of the corresponding grayscale image, over which we can apply a photometric loss. However, the number of events between the two cameras may also differ, and so we apply a similarity loss on the census transforms [21] of the images. For a given window width, W, we encode each pixel with a W^2 length vector, where each element is the sign of the difference between the pixel and each neighbor inside the window. For the left event volume, the right census transform is warped to the left camera using the left predicted disparities, and we apply a Charbonnier loss [3] on the difference between the two images, and vice versa for the right. In addition, we apply a left-right consistency loss between the two predicted disparities, as defined by [6]. Finally, we apply a local smoothness regularizer to the disparity, as in (8). The total loss for the SFM model is:

$$\begin{aligned} \mathcal{L}_{SFM} = & \mathcal{L}_{temporal} + \lambda_2 \mathcal{L}_{stereo} + \\ & \lambda_3 \mathcal{L}_{consistency} + \lambda_4 \mathcal{L}_{smoothness} \end{aligned} \tag{12}$$

	Threshold distance	10m	20m	30m
Sequence	Method	Average depth Error (m)		
outdoor_day1	Ours	**2.72**	**3.84**	**4.40**
	Monodepth	3.44	7.02	10.03
outdoor_night1	Ours	**3.13**	**4.02**	**4.89**
	Monodepth	3.49	6.33	9.31
outdoor_night2	Ours	**2.19**	**3.15**	**3.92**
	Monodepth	5.15	7.8	10.03
outdoor_night3	Ours	**2.86**	**4.46**	**5.05**
	Monodepth	4.67	8.96	13.36

Table 2: Quantitative evaluation of our depth network compared to Monodepth [6]. The average depth error is provided for all points in the ground truth up to 10m, 20m and 30m, with at least one event.

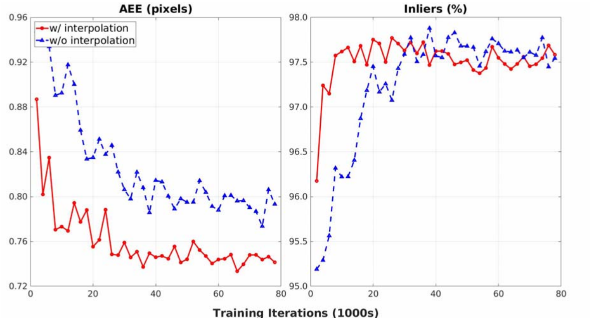

Figure 5: Ablation study on the effects of interpolation on the event volume. Flow prediction errors are shown against a held out validation set on two models with fixed random seed, with and without interpolation.

4. Experiments

4.1. Implementation Details

We train two networks on the full outdoor_day2 sequence from MVSEC [26], which consists of 11 mins of stereo event data driving through public roads. At training, each input consists of $N = 30000$ events, which are converted into discretized event volumes with resolution 256x256 (centrally cropped) and $B = 9$ bins. The weights for each loss are: $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} = \{1.0, 1.0, 0.1, 0.2\}$.

4.2. Optical Flow Evaluation

We tested our optical flow network on the indoor_flying and outdoor_day sequences from MVSEC, with the ground truth provided by [24]. Flow predictions were generated at each grayscale frame timestamp, and scaled to be the displacement for the duration of 1 grayscale frame (dt=1) and 4 grayscale frames (dt=4), separately. For the outdoor_day sequence, each set of input events was fixed at 30000, while for indoor_flying, 15000 events were used due to the larger motion in the scene. For comparison against ground truth, we convert our output, (u, v), from units of pixels/bin into units of pixel displacement with the following: $(\hat{u}, \hat{v}) = (u, v) \times (B - 1) \times dt/(t_N - t_0)$.

We present the average endpoint error (AEE), and the percentage of points with AEE greater than 3 pixels, over pixels with valid ground truth flow and at least one event. These results can be found in Tab. 1, where we compare our results against EV-FlowNet [24] and the image method UnFlow [12]. We do not provide results from ECN [20]. As their model assumes a rigid scene, and predicts egomotion and depth, they train on 80% of the indoor_flying sequences, and test on the other 20%. These results thus do not pose a fair comparison to our method, which is only trained on outdoor_day2. We do note that their outdoor_day1 errors are slightly lower than ours, at 0.30 vs 0.32. However, we believe that our method is more general, as it does not rely on a rigid scene assumption.

4.3. Egomotion Evaluation

We evaluate our ego-motion estimation network on the outdoor_day1 sequence from MVSEC. As there is currently no public code to the extent of our knowledge for unsupervised deep SFM methods with a stereo loss, we compare our ego-motion results against SFMLearner [23], and ECN [20], which learn egomotion and depth from monocular images and events. We train the SFMLearner models on the VI-Sensor images from the outdoor_day2 sequence, once again cropping out the hood of the car. These images are of a higher resolution than the DAVIS images, but are from the same scene, and so should generalize as well as training on the DAVIS images. The model is trained from scratch for 100k iterations. As the translation predicted by SFMLearner is only up to a scale, we present errors in terms of angular error. The relative pose errors (RPE) and relative rotation errors (RRE) are computed as: $\text{RPE} = \arccos\left(\frac{t_{\text{pred}} \cdot t_{\text{gt}}}{\|t_{\text{pred}}\|_2 \|t_{\text{gt}}\|_2}\right)$, $\text{RRE} = \|\text{logm}(R_{\text{pred}}^T R_{\text{gt}})\|_2$, where R_{pred} is the rotation matrix corresponding to the Euler angles from the output, and logm is the matrix logarithm.

4.4. Depth Network Evaluation

We compare our depth results against Monodepth [6], which learns monocular disparities from a stereo pair at training time. As the DAVIS grayscale images are not time synchronized, we train on the cropped VI-Sensor images. The model is trained for 50 epochs, and we provide depth errors with thresholds up to 10m, 20m and 30m in the ground truth and with at least one event. In Tab. 3, we provide the scale invariant depth metrics reported by ECN [20].

4.5. Event Volume Ablation

To test the effects of the proposed interpolation when generating the discretized event volume, we provide results in Fig. 5 of flow validation error during training between a model with and without interpolation. These results show that, while both models are able to converge to accurate flow estimates and similar % outliers, the interpolated volume achieves lower AEE.

5. Results

5.1. Optical Flow

From the quantitative results in Tab. 1, we can see that our method outperforms EV-FlowNet in almost all experiments, and nears the performance of UnFlow on the short 1 frame sequences. Qualitative results can be found in Fig. 6.

In general, we have found that our network generalizes to a number of very different and challenging scenes, including those with very fast motions and dark environments. A

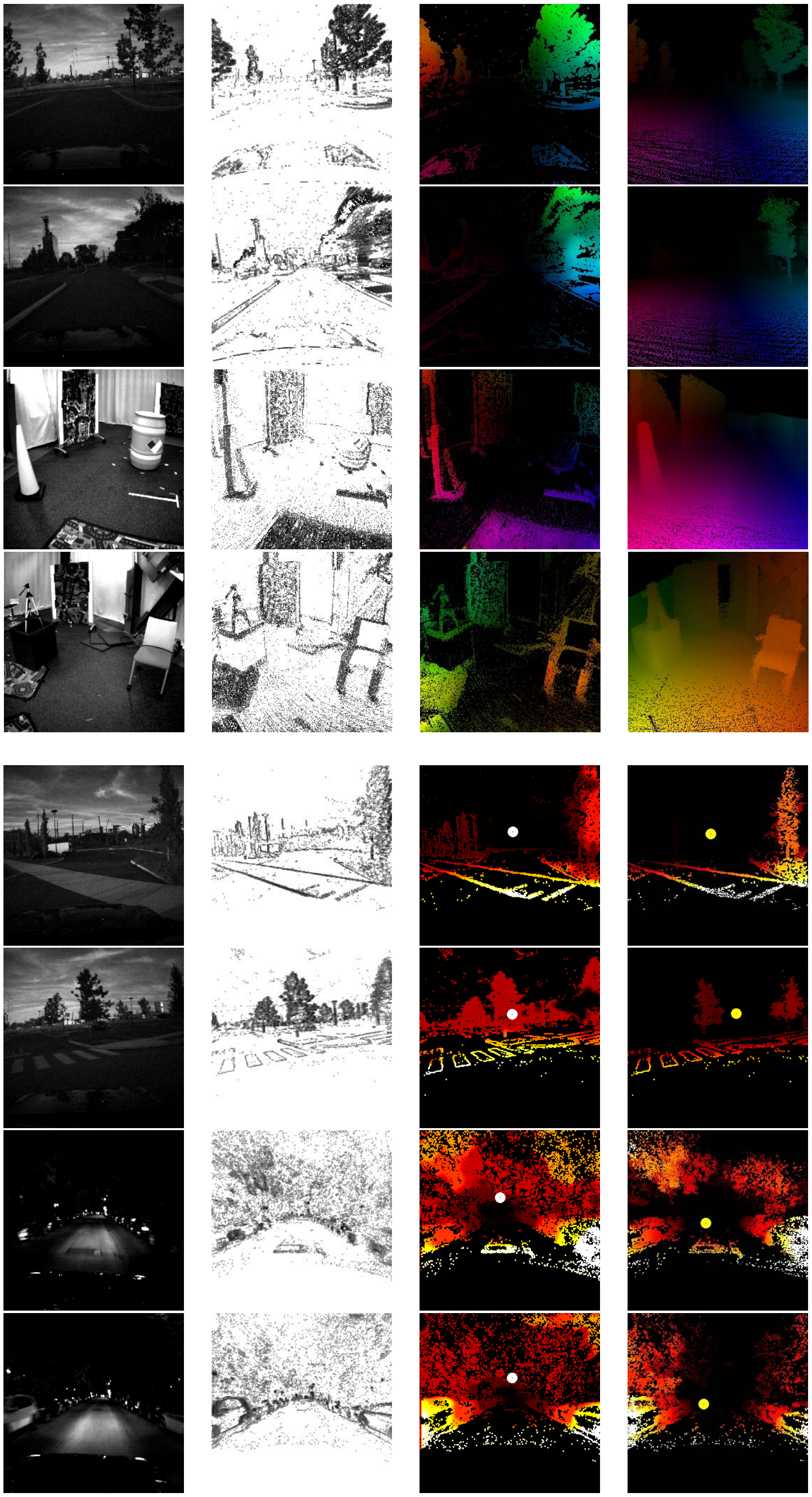

Figure 6: Qualitative outputs from the optical flow and egomotion and depth network on the indoor_flying, outdoor_day and outdoor_night sequences. From left to right: Grayscale image, event image, depth prediction with heading direction, ground truth with heading direction. Top four are flow results, bottom four are depth results. For depth, closer is brighter. Heading direction is drawn as a circle. In the outdoor_night results, the heading direction is biased due to events generated by flashing lights.

Sequence	Method	Abs Rel	RMSE log	SILog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
outdoor_day1	Ours	0.36	0.41	0.16	0.46	0.73	0.88
	ECN	**0.33**	**0.33**	**0.14**	**0.97**	**0.98**	**0.99**
outdoor_night	Ours	**0.37**	**0.42**	**0.15**	0.45	0.71	0.86
	ECN	0.39	**0.42**	0.18	**0.95**	**0.98**	**0.99**

Table 3: Quantitative evaluation of standard depth metrics from our depth network against ECN [20]. Left to right, the metrics are: absolute relative distance, RMSE log, scale invariant log, and the percentage of points with predicted depths beyond 1.25, 1.25^2 and 1.25^3 times larger or smaller than the ground truth.

	ARPE (deg)	ARRE (rad)
Ours	7.74	0.00867
SFM Learner [23]	16.27	0.00939
ECN [20]	**3.98**	**0.000267**

Table 4: Quantitative evaluation of our egomotion network compared to SFM Learner. ARPE: Average Relative Pose Error. ARRE: Average Relative Rotation Error.

few examples of this can be found in Fig. 3. We believe this is because the events do not have the fine grained intensity information at each pixel of traditional images, and so there is less redundant data for the network to overfit.

5.2. Egomotion

Our model trained on outdoor_day2 was able to generalize well to outdoor_day1, despite the environment changing significantly from an outdoor residential environment to a closed office park area. In Tab. 2, we show that our relative pose and rotation errors are significantly better than that of SFM-Learner, but worse than ECN. However, ECN only predicts 5dof pose, up to a scale factor, while our network must learn the full 6dof pose with scale. We believe that additional training data may bridge this gap.

As the network was only trained on driving sequences, we were unable to achieve good egomotion generalization to the outdoor_night sequences. We found that this was due to the fluorescent lamps found at night, which generated many spurious events due to their flashing that were not related to motion in the scene. As our egomotion network takes in global information in the scene, it tended to perceive these flashing lights as events generated by camera motion, and as a result generated an erroneous egomotion estimate. Future work to filter these kinds of anomalies out will be necessary. For example, if the rate of the flashing is known a-priori, the lights can be simply filtered by detecting events generated at the desired frequency.

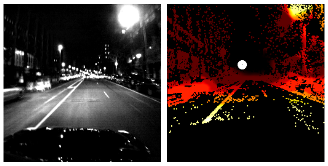

Figure 7: Failure case of our depth network. The flashing street light is detected as very close due to spurious events.

5.3. Depth

Our depth model was able to produce good results for all of the driving sequences, although it is unable to generalize to the flying sequences. This is likely because the network must memorize some concept of metric scale, which cannot generalize to completely different scenes. We outperform Monodepth in all of the sequences, which is likely because the events do not have intensity information, so the network is forced to learn geometric properties of objects. In addition, the network generalizes well even in the face of significant noise at night, although flashing lights cause the network to predict very close depths, such as in Fig. 7.

For the scale invariant metrics in Tab. 3, our method compares comparably to ECN [20] in most errors, despite having to predict the absolute scale of the depth, whereas the depths in ECN are corrected for scale. However, our δ percentages are lower than expected. We believe that additional training data can alleviate this issue in the future.

6. Acknowledgements

Thanks to Tobi Delbruck and the team at iniLabs and iniVation for providing and supporting the DAVIS-346b cameras, and to the Telluride Neuromorphic Cognition Engineering Workshop 2018 for the helpful discussions. This work was supported in part by the Semiconductor Research Corporation (SRC) and DARPA. We also gratefully appreciate support through the following grants: NSF-DGE-0966142 (IGERT), NSF-IIP-1439681 (I/UCRC), NSF-IIS-1426840, NSF-IIS-1703319, NSF MRI 1626008, ARL RCTA W911NF-10-2-0016, ONR N00014-17-1-2093, the Honda Research Institute and the DARPA FLA program.

References

[1] P. Bardow, A. J. Davison, and S. Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 884–892, 2016. 2

[2] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi. Event-based visual flow. *IEEE transactions on neural networks and learning systems*, 25(2):407–417, 2014. 2, 4

[3] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 2, pages 168–172. IEEE, 1994. 4, 5

[4] G. Gallego, H. Rebecq, and D. Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2018. 2, 4

[5] G. Gallego and D. Scaramuzza. Accurate angular velocity estimation with an event camera. *IEEE Robotics and Automation Letters*, 2(2):632–639, 2017. 2

[6] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017. 5, 6

[7] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 3

[8] J. Y. Jason, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016. 2

[9] H. Kim, S. Leutenegger, and A. J. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European Conference on Computer Vision*, pages 349–364. Springer, 2016. 2

[10] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128x128 120 dB 15μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008. 2

[11] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018. 3

[12] S. Meister, J. Hur, and S. Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2, 6

[13] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos. Event-based moving object detection and tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018. 2, 3, 4

[14] D. P. Moeys, F. Corradi, E. Kerr, P. Vance, G. Das, D. Neil, D. Kerr, and T. Delbrück. Steering a predator robot using a mixed frame/event-driven convolutional neural network. In *Event-based Control, Communication, and Signal Processing (EBCCSP), 2016 Second International Conference on*, pages 1–8. IEEE, 2016. 3

[15] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, 2(2):593–600, 2017. 2

[16] H. Rebecq, T. Horstschaefer, and D. Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In *British Machine Vis. Conf.(BMVC)*, volume 3, 2017. 2, 4

[17] F. Stein. Efficient computation of optical flow using the census transform. In *Joint Pattern Recognition Symposium*, pages 79–86. Springer, 2004. 5

[18] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017. 2, 4

[19] R. Wang, J.-M. Frahm, and S. M. Pizer. Recurrent neural network for learning densedepth and ego-motion from video. *arXiv preprint arXiv:1805.06558*, 2018. 2

[20] C. Ye, A. Mitrokhin, C. Parameshwara, C. Fermüller, J. A. Yorke, and Y. Aloimonos. Unsupervised learning of dense optical flow and depth from sparse event data. *arXiv preprint arXiv:1809.08625*, 2018. 1, 3, 6, 8

[21] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *European conference on computer vision*, pages 151–158. Springer, 1994. 5

[22] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018. 2, 4

[23] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 2, 6, 8

[24] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. 1, 2, 3, 4, 6

[25] A. Z. Zhu, N. Atanasov, and K. Daniilidis. Event-based visual inertial odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5391–5399, 2017. 2

[26] A. Z. Zhu, D. Thakur, T. Ozaslan, B. Pfrommer, V. Kumar, and K. Daniilidis. The multi vehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018. 2, 6