Learning Event-based Height from Plane and Parallax

Kenneth Chaney Alex Zihao Zhu Kostas Daniilidis

***Abstract*— Event-based cameras are a novel asynchronous sensing modality that provide exciting benefits, such as the ability to track fast moving objects with no motion blur and low latency, high dynamic range, and low power consumption. Given the low latency of the cameras, as well as their ability to work in challenging lighting conditions, these cameras are a natural fit for reactive problems such as fast local structure estimation. In this work, we propose a fast method to perform structure estimation for vehicles traveling in a roughly 2D environment (e.g. in an environment with a ground plane). Our method transfers the method of plane and parallax to events, which, given the homography to a ground plane and the pose of the camera, generates a warping of the events which removes the optical flow for events on the ground plane, while inducing flow for events above the ground plane. We then estimate dense flow in this warped space using a self-supervised neural network, which provides the height of all points in the scene. We evaluate our method on the Multi Vehicle Stereo Event Camera dataset, and show its ability to rapidly estimate the scene structure both at high speeds and in low lighting conditions.**

I. Introduction

Event cameras provide exciting new benefits over traditional cameras allowing for low latency obstacle tracking and motion blur compensation. Autonomous vehicles can benefit greatly from event cameras simply from the lower latency, lower bandwidth, and higher dynamic range that is native to the sensor itself. However, these benefits come with a variety of challenges, mostly stemming from the lack of decades of dedicated research for tasks like optical flow, structure from motion, and other tasks that have been heavily studied for traditional cameras. Algorithms that have been created for traditional cameras rely upon globally synchronous information and can't simply be ported to a sparse asynchronous stream.

Recent advances in structure estimation from monocular images and event based cameras [20], [19], [23] provide methods for constructing local depth maps that can be used for obstacle detection and other tasks that only require local structure. In autonomous vehicles, obstacle detection is a low level requirement to ensure vehicle and environment safety at all points in time, regardless of whether or not higher level software or hardware subsystems fail. Relying upon a full SLAM package to produce local obstacle maps is impractical due to the overall delay in retrieving a fully optimized pose and map; by the time the information is received it may already be irrelevant. On the other hand, LiDAR systems have a relatively low update frequency leading to long delays obstacle information updates. Similarly stereo cameras have a practical upper limit to the frame rate due to the need to process pairs of frames individually.

The authors are with the University of Pennsylvania, Philadelphia, PA, USA. Emails are: {chaneyk, alexzhu, kostas}@seas.upenn.edu.

In this paper, we propose a novel structure estimation method for event cameras that is suitable for on autonomous driving in real world scenarios. We utilize recent advances in unsupervised learning methods for event cameras to train a convolutional neural network to learn height and depth from a loss that utilizes Plane and Parallax (P+P) [15] principles, which reduces the complexity of the dense optical flow by removing rotation while simultaneously providing a direct metric representation of the magnitude of the computed flow. Our network takes as input raw events, and predicts the ratio between the height of each point above the ground plane, and the depth in the camera frame. We show that this ratio can be used to compute the optical flow between a pair of images warped using P+P, and apply a semi-supervised loss to minimize the photometric error between the images.

In order to accurately predict metric depth directly from a scene, a network must learn to make a large number of assumptions about objects in the scene such as cars, pedestrians, and buildings. As such, these networks have a hard time generalizing to other contexts. Predicting relative factors up to a scale that represent the structure of the scene, but need to be scaled or otherwise decoded, allows networks to generalize better. Our method leverages this by predicting the ratio of height and depth. Alone, this provides a relative measurement, but when coupled with a camera to ground calibration, it allows for the system to recapture the full metric information of the scene, in a similar manner to the monocular implementation that accompanies Geiger et al. [3].

Our method runs at 75Hz on a modern high grade GPU, and can estimate scene height and depth in low-light and high speed driving scenarios, making it suitable for night time autonomous driving. We evaluate our method on the Multi Vehicle Stereo Event Camera (MVSEC) dataset [22], and demonstrate our network's ability to accurately predict the heights and depths of objects in the scene. We further show an application of these predictions towards accurately segmenting free space on the ground plane. In our experiments, we demonstrate superiority over image input and depth prediction baselines.

The technical contributions of the paper are as follows:

- A novel loss that leverages P+P to isolate static scene components from the motion of the event camera.
- A novel pipeline that trains a neural network to learn the ratio between the height of a point from the ground plane and its depth in the camera frame, using a self-supervised loss, where camera pose and the ground

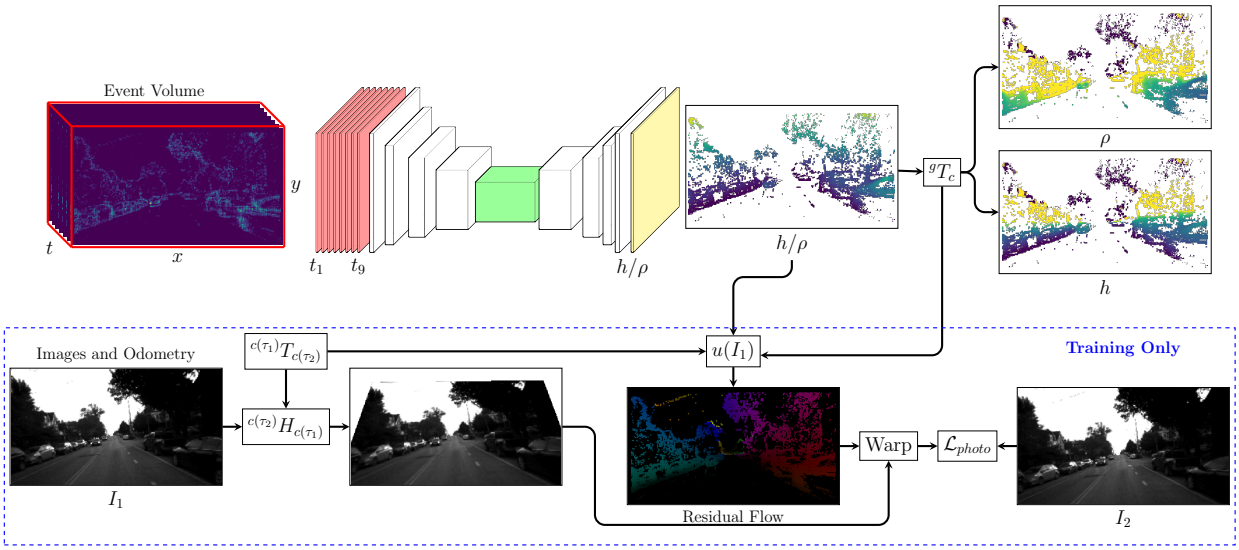


Fig. 1: Our network, which uses the architecture from EV-FlowNet [21], takes as input a raw event volume, and predicts the scene structure in the ratio height (h) over depth (ρ). Given the ground plane calibration, we can recover depth and height independently. During training (blue box), odometry and the predicted scene structure is used in a two stage warping process to warp I_1 into I_1^w. First, a homography, $^{c(\tau_2)}H_{c(\tau_1)}$, warps and aligns the ground plane between the images. Second, the scene structure is used to compute the residual flow to warp any portions of the scene not on the ground plane. I_1^w is compared to I_2 in $\mathcal{L}_{photo}$. At test time, only the calibration from the camera to the ground plane, ^gT_c, is used at test time to compute ρ and h. (Figure best viewed in color)

normal is used at training time, but where only the ground normal is needed at test time.

- Evaluation on challenging high-speed and low-light night time MVSEC dataset scenes, including qualitative results for freespace estimation.

II. Related Work

A. Event Based Cameras

A number of recent works have used deep learning for event cameras. Moeys et al. [11] and Maqueda et al. [9] proposed supervised methods to predict the position of a robot in front of the camera and the steering angle of a vehicle, respectively, while Zhu et al. [21] and Ye et al. [19] propose self and unsupervised methods to predict optical flow and egomotion and depth.

While there have not been many works directly focused on general obstacle detection, there have been a number of works that perform camera pose [7], [14] and depth estimation [13], [24], [12] with event cameras, which could be joined to perform obstacle detection.

Several works also detect independently moving objects from the events stream. Vasco et al. [17] track features in the image and filter those that do not fit the motion predicted by the known camera poses. Mitrokhin et al. [10] extend this by using a 4 parameter camera motion model to model and estimate the rigid motion of the camera, and similarly detect independently moving objects as events that do not fit the model.

However, these methods require extensive engineering to fuse methods together and separately define obstacles from the measurements, whereas our method directly outputs the height of each obstacle, which can easily be thresholded by a single hyperparameter.

B. Plane and Parallax

Plane and Parallax (P+P) has been leveraged for reconstruction of structure from multiple images as well as structureless obstacle detection. Sawhey et al. [15] derives a parameterization of the residual flow based on metric information in each plane, depth relative to the camera and height relative to the ground plane. Irani et al. [5] proposes a method that leverages P+P to retrieve the structure of the scene from multiple uncalibrated images. Wulff et al. [18] propose a method that utilizes P+P to refine the optical flow esimations of the rigid portion of the scenes.

Simond et al. [16] propose a method to perform obstacle detection without reconstruction of the scene structure, but needs a calibration of the camera against the ground to retrieve metric obstacle information. Mallot et al. [8] describe a scheme to detect obstacles using optical flow and leverage inverse perspective mapping for regularizing the optical flow calculations and relating the flow to obstacle height.

These methods are targeted towards traditional cameras, which we extend in this work for event cameras.

III. METHOD

Our obstacle detection pipeline consists of a fully convolutional neural network that takes in a discretized event volume, as described in Sec. III-A, and predicts the structure component, consisting of the ratio between the height of each pixel above the plane over the depth from the camera. Our training method is inspired by the work by Zhu et al. [21], where supervision is applied through the photoconsistency assumption, applied to the grayscale image output from some event cameras such as the DAVIS [1]. At training time, we gather two grayscale images and all of the events in between them in time, and use P+P to register the previous image to the next. We explain this process in detail in Sec. III-B. Our network then predicts the structure component for each pixel, which we combine with the known camera poses between the two images to estimate the residual flow for each pixel. This residual flow is then used to warp the previous image to the next. We apply a photometric loss to the image pair, which we explain in Sec. III-C. Our method is also summarized in Alg. 1.

Note that, while we assume that the camera pose is known in this work at training time, it can be jointly predicted with the structure component in a similar manner to Zhou et al. [20]. However, the goal of this work is accurate height and depth prediction, so we use known camera pose to further refine the predictions.

A. Input Representation

Several prior works in using events for learning have summarized the events in an image [11], [21], [9], [19]. While this was shown to allow a network to learn accurate optical flow on raw events, this representation inherently loses much of the high temporal resolution of the events by throwing away most of the timestamps. To resolve this issue, we instead represent the events as a 3D spatiotemporal volume, as proposed by Zhu et al. [23]. First, we discretize the time dimension into B bins. However, simply rounding the events and inserting them into the volume would lose a significant amount of information, and so we instead insert events using trilinear interpolation, similar to the bilinear image warping used in Jaderberg et al. [6]:

$$t_i^* = (B-1)(t_i - t_0)/(t_N - t_1) \tag{1}$$

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

$$k_b(a) = \max(0, 1 - |a|) \tag{3}$$

where $k_b(a)$ corresponds to the sampling kernel defined in [6].

In other words, each event contributes to the eight voxels around it in x-y-t space, with contribution to each voxel proportional to one minus the distance from the event to that voxel in each dimension.

For events without any neighbors (e.g. on the boundary of an object), this interpolation allows for the exact timestamp to be recovered, and overall, this representation encodes the full spatiotemporal distribution of the events.

Algorithm 1 Event-based Height from Planar Parallax

Training

Input: events, $\{x_i, y_i, t_i, p_i\}_{i=1,\dots,N}$, images, I_1, I_2, camera pose, ${}^{c_2}T_{c_1}$, ground plane, ${}^{c}T_{g}$.

1: Generate event volume (2).
2: Register I_1 to I_2 using P+P, generating I_1^w (11).
3: Pass the event volume through the network to estimate the structure component, $A(\vec{x})$.
4: Estimate the residual flow, $u(\vec{x})$, (12), and use it to warp I_1^w to $\hat{I}_2$.
5: Apply the loss, $\mathcal{L}_{\text{total}}$, (18) and backpropagate.

Testing

Input: events, ground plane.

1: Generate event volume (2) and predict the structure component, $A(\vec{x})$ with the network.
2: Estimate the height, $h(\vec{x})$ (15) and depth, $\rho(\vec{x})$ (14) of each point.

B. Self-Supervision from Plane and Parallax

For supervision, we apply a plane and parallax (P+P) warping to the grayscale frames accompanying the events, and use the output of the network to estimate the residual optical flow after this warping.

Plane and parallax (P+P) methods warp images (or individual points) through a common reference plane to create parallax between images. The warping will exactly register points that lie on the plane, while points above or below will have some residual flow, which can be parameterized by a rigid structure parameter and camera motion.

The P+P warping can be represented as the homography, ${}^{c(\tau_2)}H_{c(\tau_1)}$, which transforms the ground plane in the $c(\tau_1)$ frame to the $c(\tau_2)$ frame.

At training time, we apply a P+P warping on the image immediately before the event volume, I_{τ_1}, to the one immediately after, I_{τ_2}. To generate the homography, we assume that the fixed transformation between the camera and the ground frame, ${}^{c}T_{g}$, and the relative pose between the camera frames, ${}^{c(\tau_2)}T_{c(\tau_1)}$, are known. T is composed of the homogeneous form of the rotation, R, and translation, t: $T = [R, t; [0,0,0,1]]$.

The relative pose between camera frames can be decomposed as two transformations from the camera to the ground at the respective times:

$${}^{c(\tau_2)}T_{c(\tau_1)} = {}^{c}T_{g}\,{}^{g}T_{c}\,{}^{c(\tau_2)}T_{c(\tau_1)} \tag{4}$$

$${}^{c(\tau_2)}T_{g} = {}^{c}T_{g} \tag{5}$$

$${}^{c(\tau_1)}T_{g} = ({}^{g}T_{c}\,{}^{c(\tau_2)}T_{c(\tau_1)})^{-1} \tag{6}$$

The homography, ${}^{c(\tau_2)}H_{c(\tau_1)}$, that passes through the ground plane, can then be generated as the composition of two homographies to the ground plane, ${}^{c(\tau_1)}H_g$ and ${}^{c(\tau_2)}H_g$:

$${}^{c(\tau_2)}H_{c(\tau_1)} = {}^{c(\tau_2)}H_{g}\,{}^{c(\tau_1)}H_{g}^{-1} \tag{7}$$

$$\tag{8}$$

Each homography from a camera plane to the ground plane, ${}^{c(\tau_i)}H_g$, is defined as:

$${}^{c(\tau_i)}H_{g} = {}^{c(\tau_i)}R_{g} + \begin{bmatrix} 0 & 0 & {}^{c(\tau_i)}t_{g} \end{bmatrix} \tag{9}$$

So, ${}^{c(\tau_2)}H_{c(\tau_1)}$, can be completely constructed in terms of the known poses:

$$
\begin{aligned}
{}^{c(\tau_2)}H_{c(\tau_1)} = &\left({}^{c(\tau_2)}R_g + \begin{bmatrix} 0 & 0 & {}^{c(\tau_2)}t_g \end{bmatrix}\right) \\
&\left({}^{c(\tau_1)}R_g + \begin{bmatrix} 0 & 0 & {}^{c(\tau_1)}t_g \end{bmatrix}\right)^{-1}
\end{aligned} \tag{10}
$$

Given the homography, every pixel in the previous image, I_1 can be warped according to the following equation to generate the warped image I_1^w:

$$
\mu p(\tau_2) = {}^{c(\tau_2)}H_{c(\tau_1)} p(\tau_1) \tag{11}
$$

C. Residual Flow Loss

After P+P, the only optical flow between the images I_1^w and I_2 corresponds to flow induced by the height of the point off the ground plane. We train our network to learn a rigid structure parameter which can be used to recover the flow, which is used to further warp I_1^w. Finally, we apply a photometric loss on the warped images, as in Zhu et al. [21].

This residual flow, $u(\vec{x})$, can be written as:

$$
u(\vec{x}) = \frac{A(\vec{x})b}{A(\vec{x})b - 1}(e - \vec{x}) \tag{12}
$$

$$
A(\vec{x}) = \frac{h(\vec{x})}{\rho(\vec{x})}, b = \frac{{}^{c_2}t_{c_1}(3)}{{}^{c}t_g(3)} \tag{13}
$$

where $h(\vec{x})$ and $\rho(\vec{x})$ are the height and depth of point $\vec{x}$, respectively, e is the epipole in the image, ${}^{c_2}t_{c_1}(3)$ is the translation along the Z axis and ${}^{c}t_g(3)$ is the height of the camera c above the ground plane. We refer to Wulff et al. [18] for the full derivation of this equation. Figure 2 provides context for the geometric relations between the key components relating to the residual flow and why the residual flow projects where it does.

As a result of the warping, the flow for each pixel can now be fully parameterized by a single scalar, composed of a rigid structure component, $A(\vec{x})$ and a time varying component, b, which we assume is known.

Given an event volume from (2), we train our network to learn the structure component, $A(\vec{x})$, from which we can exactly recover the height, $h(\vec{x})$, and depth, $\rho(\vec{x})$, of each point:

$$
\rho(\vec{x}) = \frac{{}^{g}t_c(3)}{A(\vec{x}) - {}^{g}R_c(3,:)\vec{x}} \tag{14}
$$

$$
h(\vec{x}) = A(\vec{x})\rho(\vec{x}) \tag{15}
$$

Predicting $A(\vec{x})$ has advantages over directly predicting the height, $h(\vec{x})$. While one can recover depth from height for most points, this relationship fails for vectors parallel to the ground. For any such vector, there are an infinite number of possible depths at that height, and so a large area of the image would be unstable during training. Predicting $A(\vec{x})$ avoids this issue.

Using (12), we can use the network predictions to estimate the optical flow at every pixel. This flow is used to further warp I_1^w towards I_2, generating $\hat{I}_2$ to remove the residual flow. Our loss, then, is the photometric loss plus a local smoothness loss:

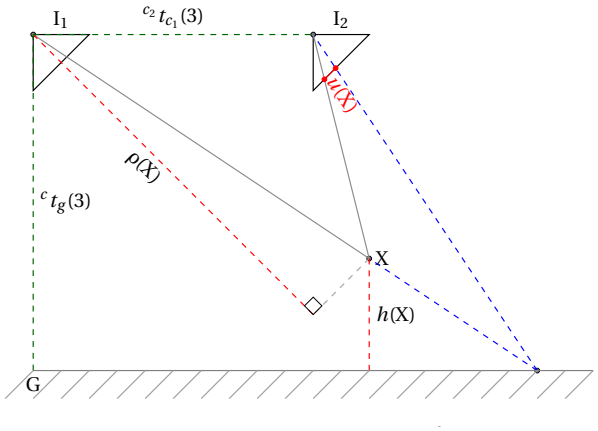


Fig. 2: P+P methods rely upon using a homography to warp from one image, I_1 to another I_2. This can be thought of as projecting a point, X, onto the ground plane and then back up to I_2. Points that don't lie on the ground plane create a residual, $u(X)$. The structure of X, $\rho(X)$ and $h(X)$, and camera motion, ${}^{c_2}t_{c_1}(3)$ and ${}^{c}t_g(3)$, relate to the residual flow by (12). (Figure best viewed in color)

$$
\mathcal{L}_{\text{photometric}} = \sqrt{(I_2 - \hat{I}_2)^2 + \epsilon^2} \tag{16}
$$

$$
\mathcal{L}_{\text{smoothness}} = \sum_{\vec{x}} \sum_{\vec{y} \in N(\vec{x})} \sqrt{(A(\vec{x}) - A(\vec{y}))^2 + \epsilon^2} \tag{17}
$$

$$
\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{photometric}} + \lambda \mathcal{L}_{\text{photometric}} \tag{18}
$$

In both losses, we use the robust Charbonnier loss [2]. $N(\vec{x})$ is the 4-connected neighborhood around $\vec{x}$ and λ is a hyperparameter weight.

IV. Results

A. Implementation Details

Our network was trained on the outdoor_day2 sequence from MVSEC, which contains roughly 20,000 images from a DAVIS 346b stereo pair, of which we only use the left camera's events and images. The events and images are cropped to 176×336 pixels to remove the hood of the car in the images. The weighting parameter for the smoothness loss, λ, in the loss is set to 0.2, and all networks were trained for 50 epochs for all experiments. The ground plane to camera extrinsic calibration was computed by applying a RANSAC plane fit to each ground truth depth image in outdoor_day2 and taking the median plane. This calibration was then used for all other experiments. Camera poses were gathered from the ground truth odometry provided in each sequence.

B. Experiments

1) Depth and Height Evaluation: We compare our model, which uses an event volume as input, against an implementation that uses images as input instead of events, which

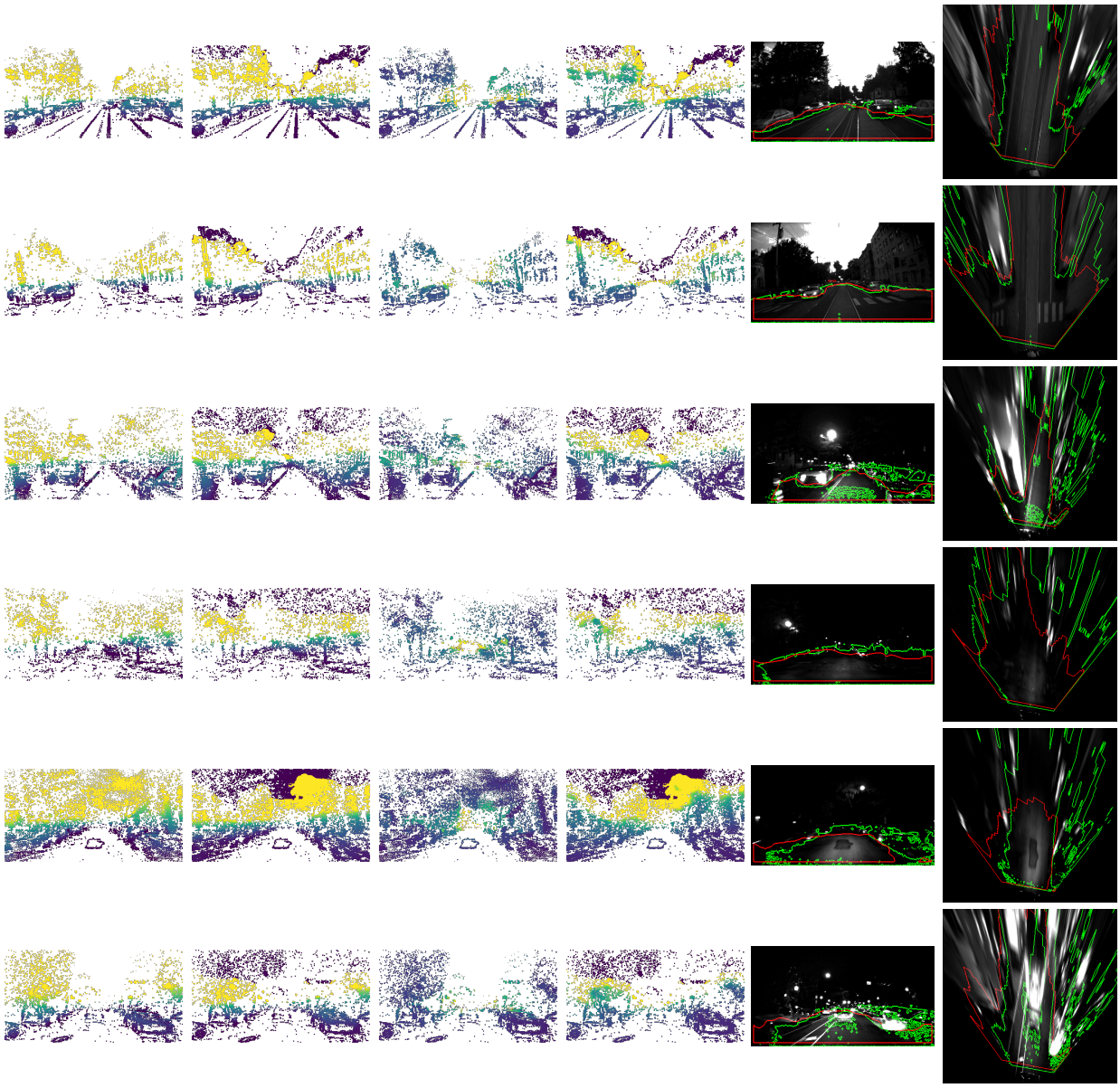


Fig. 3: Output of our method, Left to right: (a-d) The ground truth and predictions of height and depth at pixels with events over the time window in which events were collected (e) The grayscale image overlayed with the ground truth (green) and network (red) free space regions in the camera frame (f) The freespace image projected into the ground frame.

we label Image P+P. Additionally we compare against a weakly supervised network which directly predicts depth from events, which we label Event Depth. This model utilizes the odometry feed and predicted depths to warp between images for a photometric loss, through the standard motion field equations [4]. This is similar to the warping used in unsupervised methods such as SFMLearner [20], except here the pose is taken from the ground truth, rather than directly regressed. Qualitative results from these experiments can be found in Fig. 3.

For testing, we evaluate on the Outdoor Day 1 and Outdoor Night 1-3 sequences from MVSEC. Each network constructs depth and height maps, which is compared directly against the ground truth provided with the sequences. For the Event Depth network, heights are estimated given the predicted depths and provided ground plane as follows:

$$X(\vec{x}) = {}^{g}R_{c(\tau_2)}\rho(\vec{x})K^{-1}\begin{pmatrix}\vec{x}\\1\end{pmatrix} + {}^{g}t_{c(\tau_2)} \tag{19}$$

$$h(\vec{x}) = X(\vec{x})(3) \tag{20}$$

The same architecture and parameters are used across all three networks, with the only differences being the input representation for Image P+P and the loss function for Event Depth.

outdoor_day1	Average Depth Error (m)			Average Height Error (m)		
Threshold	ρ <10m	ρ <20m	ρ <100m	-0.5m< h <5.0m	0.1m< h <5.0m	1.0m< h <5.0m
Event P+P	6.26	8.37	**10.77**	0.55	0.69	1.00
Image P+P	**4.15**	**7.45**	12.06	1.09	1.18	1.44
Event Depth	14.49	13.83	13.37	**0.48**	**0.59**	**0.81**
outdoor_night1						
Event P+P	**2.93**	**4.30**	**6.36**	**0.41**	**0.42**	**0.50**
Image P+P	4.57	7.33	10.61	1.00	1.15	1.44
Event Depth	15.62	15.88	15.27	0.48	0.53	0.62
outdoor_night2						
Event P+P	**2.89**	**5.07**	**6.94**	**0.37**	**0.40**	**0.55**
Image P+P	4.41	7.75	10.58	1.27	1.58	2.03
Event Depth	10.8	11.09	10.82	0.40	0.47	0.60
outdoor_night3						
Event P+P	**3.24**	**5.61**	**7.64**	**0.41**	**0.47**	0.70
Image P+P	4.45	8.01	10.97	1.41	1.84	2.35
Event Depth	13.17	12.39	11.50	0.42	0.51	**0.66**

TABLE I: Results of the baseline networks against our network on all testing scenes. For all evaluation, only pixels with events during the relevant time window are evaluated. The thresholds for depth and height are applied to the ground truth depth and height images to create a additional mask to evaluate within.

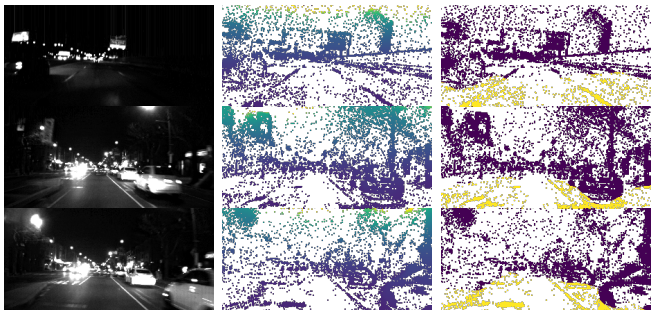
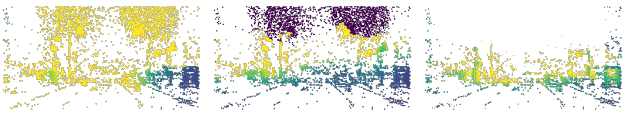


Fig. 4: Qualitative results from the motorcycle sequence. Left to right: (a) Grayscale image (b) Predicted height over pixels with events (c) Freespace mask (yellow is free).



(a) Event Depth (b) Event P+P (c) Ground Truth

Fig. 5: A comparison of the same scene using the Event Depth and Event P+P methods. The Event Depth method estimates all objects to be far away. This explains the trends seen in Table I.

Quantitative results were computed as absolute error with respect to the ground truth provided by MVSEC for both height and depth. In our experiments, we found that, as the sensors were mounted on the middle of the car, i.e. at least 2.5m from the front of the car, there were very few visible points very close to the sensors, especially after cropping the hood of the car. Therefore, we have set the lowest threshold for depth at 10 m. These results can be found in Table I.

Our method, Event P+P, provides results close to or better than the baselines in all cases in the direct comparison. In addition, the network is able to generalize to the night time sequences where there are significantly more noisy events, as well as to a change in environment in outdoor_day1, which is inside an office park as compared to the suburban roads observed in the training set. Overall, it seems that it is easier for the network to generalize to a noisier night time environment than it is to a visually different scene, due to the higher errors in the latter case.

The depth network, perhaps surprisingly, performs significantly worse than the P+P methods across the depth metrics. One possible explanation is that, while the provided lidar odometry is reasonably accurate, there is nonetheless noise in the provided poses. As a result, we found that this resulted in increased error in the Event Depth network's depth estimations, particularly for closer points. A qualitative example of this can be found in Fig. 5. This can also be seen in the quantitative results where depth error for Event Depth decreases when points further away are included. This suggests that using the proposed P+P loss is more robust to error in the ground truth pose than a motion stereo loss.

One deficiency of our method seen in experiments is that the network tends to oversmooth the sky regions into objects at test time. This can be seen in the qualitative results where the higher regions in the image appear purple. This is likely because the sky points are perceived to be at near infinite depth. As a result, the structure term, $A := h(\vec{x})/\rho(\vec{x})$, is close to 0. With very small values for A, it is difficult to properly resolve the height and depth, due to noise in the system. Hence, points near these regions in the image are occasionally classified to have near 0 depth and height. However, for applications such as freespace detection in a 2D (e.g. automotive) setting, this is not a problem, as these points are simply detected to lie on the ground plane, avoiding detection of false positives. As the points are typically very high up, out of the path of the vehicle, they would also not be considered obstacles in the path of the vehicle. Future

work can consider fine tuning or modifying the smoothness term to reduce this effect.

2) Freespace Estimation: As a downstream application of this work, the height maps can be used to classify each pixel as freespace or an obstacle, given the transform between the camera and the ground plane. In other words, to classify whether each pixel belongs to the ground plane or should be considered an obstacle. To demonstrate this, we threshold the heights such that any point with height $<$0.1 m is considered freespace, while any point above this is an obstacle.

Qualitative results of this heuristic applied on both the network outputs and ground truth depths can be seen in Figure 3. The results demonstrate examples where this method, with a relatively simple heuristic, is able to successfully segment obstacles such as other cars from the freespace. Using such a simple heuristic can allow for a fast initial detection of upcoming obstacles in the scene (provided a sufficiently powerful GPU).

3) High Speed Tests: In order to demonstrate the ability of our proposed pipeline on different vehicles and fast motions, we tested our network on the motorcycle sequence from MVSEC. This sequence contains a motorcycle driving at night on surface streets and highway, with speeds up to 140 km/hr. Ground truth depths are not available, but nevertheless we are able to provide qualitative results from the output of the network in Fig. 4. Due to the lack of ground truth, the camera to ground calibration was only roughly tuned manually. In particular, these results show that it is possible to accurately segment other vehicles from free space, by thresholding points with height $<$0.1 m as freespace.

V. Conclusion

In this work, we have demonstrated a novel pipeline for self-supervised prediction of heights and depths from events from an event camera. We show that our method works in a number of challenging driving scenes, including night time scenarios, and improves upon baselines with image inputs and direct depth predictions. We also show that our method allows us to accurate estimate the free space on the ground plane, and hope that this work can drive future work in providing high speed safe reactive methods using event cameras for autonomous vehicles.

Acknowledgments: We gratefully appreciate support through the following grants: NSF-IIP-1439681 (I/U-CRC), NSF-IIS-1703319, NSF MRI 1626008, ARL RCTA W911NF-10-2-0016, ONR N00014-17-1-2093, ARL DCIST CRA W911NF-17-2-0181, Amazon Robotics Award, and by Honda Research Institute. This work was supported in part by the Semiconductor Research Corporation (SRC) and DARPA.

References

[1] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck. A 240$\times$ 180 130 dB 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014.

[2] 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.

[3] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.

[4] B. Horn, B. Klaus, and P. Horn. *Robot vision*. MIT press, 1986.

[5] M. Irani, P. Anandan, and M. Cohen. Direct recovery of planar-parallax from multiple frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1528–1534, 2002.

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

[7] 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.

[8] H. A. Mallot, H. H. Bülthoff, J. Little, and S. Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, 64(3):177–185, 1991.

[9] A. I. Maqueda, A. Loquercio, G. Gallego, N. Garcıa, 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.

[10] A. Mitrokhin, C. Fermuller, C. Parameshwara, and Y. Aloimonos. Event-based moving object detection and tracking. *arXiv preprint arXiv:1803.04523*, 2018.

[11] 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.

[12] E. Piatkowska, A. Belbachir, and M. Gelautz. Asynchronous stereo vision for event-driven dynamic stereo sensor using an adaptive cooperative approach. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 45–50, 2013.

[13] 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.

[14] H. Rebecq, T. Horstschaefer, and D. Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In *British Machine Vision Conference*, 2017.

[15] H. S. Sawhney. 3d geometry from planar parallax. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 94, pages 929–934, 1994.

[16] N. Simond and M. Parent. Obstacle detection from IPM and super-homography. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 4283–4288. IEEE, 2007.

[17] V. Vasco, A. Glover, E. Mueggler, D. Scaramuzza, L. Natale, and C. Bartolozzi. Independent motion detection with event-driven cameras. In *Advanced Robotics (ICAR), 2017 18th International Conference on*, pages 530–536. IEEE, 2017.

[18] J. Wulff, L. Sevilla-Lara, and M. J. Black. Optical flow in mostly rigid scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 7. IEEE, 2017.

[19] 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.

[20] 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.

[21] 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.

[22] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis. The Multivehicle Stereo Event Camera Dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018.

[23] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019.

[24] A. Zihao Zhu, Y. Chen, and K. Daniilidis. Realtime time synchronized event-based stereo. In *The European Conference on Computer Vision (ECCV)*, September 2018.