DeepEvent-VO: Fusing Intensity Images and Event Streams for End-to-End Visual Odometry

Abhay Gupta, Ganesh Iyer, Suhit Kodgule
{abhayg, giyer, skodgule}@andrew.cmu.edu
16-833 Project Report

***Abstract*— With the advent of Deep Learning (DL) algorithms, there has been an increased shift in using these techniques for Visual Odometry. Numerous supervised, self-supervised and unsupervised methods have been proposed. With the recent surge in the use of event-based cameras that capture high-speed intensity changes as log representations, it is possible to encode information that is missed by regular cameras between time steps. Towards this end, we propose DeepEvent-VO, a fusion-based supervised RCNN model that learns geometric features between monocular intensity sequences across time and fuses features learned from the event-frames encoded between these time steps, before passing to LSTMs to learn poses at each time step. To our knowledge, this is the first attempt in fusing features from both intensity and event sequences. We explain the architecture in more detail and show different experiments that we have performed along with ablation studies using different learning strategies. From our learnings, we give directions for the future to help improve the results of this project.**

I. Introduction

Visual odometry (VO), as one of the most essential techniques for pose estimation and robot localization, has attracted significant interest in both the computer vision and robotics communities over the past few decades [1]. It has been widely applied to various robots as a complement to GPS, Inertial Navigation System (INS), wheel odometry, etc.

In the last thirty years, enormous work has been done to develop an accurate and robust monocular VO system. As shown in Figure 1, a classic pipeline [1], [2], which typically consists of camera calibration, feature detection, feature matching (or tracking), outlier rejection (e.g., RANSAC), motion estimation, scale estimation and local optimization (Bundle Adjustment), has been developed and broadly recognized as a golden rule to follow. Although some state-of-the-art algorithms based on this pipeline have shown excellent performance in terms of accuracy and robustness, they are usually hard-coded with significant engineering effort and each module in the pipeline needs to be carefully designed and fine-tuned to ensure the performance. Moreover, the monocular VO has to estimate an absolute scale by using some extra information (e,g., the height of the camera) or prior knowledge, making it prone to big drift and more challenging than the stereo VO.

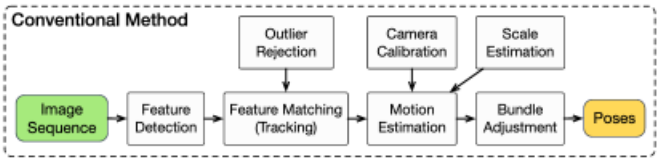


Fig. 1: Conventional framework for Monocular VO

DL has recently been dominating many computer vision tasks with promising results. Unfortunately, for the VO problem, this domination has not arrived yet. In fact, there is limited work on VO, even related to 3D geometry problems. We presume that this is because most of the existing DL architectures and pre-trained models are essentially designed to tackle recognition and classification problems, which drives deep Convolutional Neural Networks (CNNs) to extract high-level appearance information from images. Learning the appearance representation confines the VO to function only in trained environments and seriously hinders the popularization of the VO to new scenarios. This is why the VO algorithms heavily rely on geometric features rather than appearance ones. Meanwhile, a VO algorithm ideally should model motion dynamics by examining the changes and connections on a sequence of images rather than processing a single image. This implies that we require sequential learning, which CNNs are inadequate to provide.

By registering changes in log intensity in the image with microsecond accuracy, event-based cameras offer promising advantages over frame-based cameras in situations with factors such as high-speed motions and difficult lighting. By directly measuring the precise time at which each pixel changes, the event stream directly encodes fine-grain motion information. One interesting application of this is the estimation of optical flow to facilitate the learning of cues which aid in learning better transformations across sequences.

In this report, we propose a fusion-based DL architecture for VO based on Recurrent Convolutional Neural Networks (RCNNs) [3]. Since it is achieved in an end-to-end manner, it does not need any module in the classic VO pipeline (even camera calibration). The main contributions can be summarized as follows: 1) We demonstrate that the monocular VO problem can be enhanced with the use of event-

streams. 2) We show a fusion based strategy that can help learn good geometrical representations, an inherent need for the VO pipeline using CNNs. 3) Finally, we show an end-to-end framework based on RCNNs enabling generalization to new environments.

The rest of the report is categorized in the following manner. Section II covers related work in this direction. Section III describes what event-based time surfaces are. Section IV describes our model and fusion strategy. Section VI shows the results of the models, with and without fusion and also shows some ablation studies. Section VII provides future directions for using event streams for VO.

II. Related Work

There has been a surge in using deep learning for Visual Odometry both using stereo information or monocular RGB streams and more recently using event streams for 6 DoF pose estimation. The rest of the section is divided into parts for DL based VO algorithms using images, using events and models that work on the fusion of more than two streams.

A. Monocular VO using DL

Monocular based VO has been around for a long time, mostly using traditional vision techniques in a pipeline as shown in Figure 1. DeepVO [11], as shown in Figure 2 was one of the first works that extended deep learning and sequence learning to VO for monocular cameras. UnDeepVO [15] was an extension of this model, utilizing disparity from stereo images as a supervisory signal during training to estimate pose and testing the model directly on monocular systems. DeMON [16] uses depth as a supervisory signal and feeds the predicted depth into a pose CNN to learn the ego-motion of the camera. Zhu et al. [17] propose a hybrid pipeline to use both optical-flow and depth as supervisory signals to learn good scene information and then use RANSAC on top of this to learn an inlier mask and the pose. Most of the work in this field is dominated by unsupervised methods.

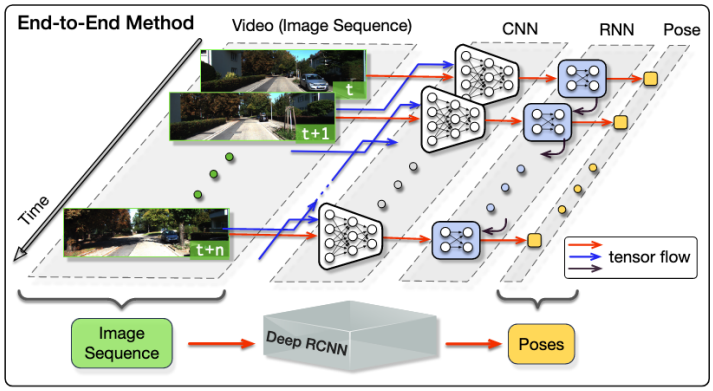


Fig. 2: The architecture for DeepVO. This uses FlowNet to extract features from two consecutive time step RGB images and then uses stacked LSTMs to learn the poses.

B. Event-Based Visual Odometry

Event-based camera streams have recently been used for visual odometry and SLAM. H. Rebecq et. al. [34], and Mueggler et. al. [35] demonstrated the use of event cameras for high-speed visual odometry, visual inertial odometry, and SLAM. Some of the methods include tracking line and point features, and trajectory alignments based on spline fitting and smoothing by optimization.

C. Event-Based Deep Learning

One of the main challenges for supervised learning for events is the lack of labeled data. As a result, many of the early works on learning with event-based data, such as Ghoshet al. [20] and Moeys et al. [21], rely on small, hand-collected datasets.

To address this, recent works have attempted to collect new datasets of event camera data. Mueggler et al. [22], provide handheld sequences with ground truth camera pose, which Nguyen et al. [12] use to train an LSTM network to predict camera pose. In addition, Zhu et al. [23] provide flying, driving and handheld sequences with ground truth camera pose and depth maps, and Binas et al. [24] provide long driving sequences with ground truth measurements from the vehicle such as steering angle and GPS position.

Another approach has been to generate event-based equivalents of existing image-based datasets by recording images from these datasets from an event-based camera (Orchard et al. [25], Hu et al. [26]). Recently, there have also been implementations of neural networks on spiking neuromorphic processors, such as in Amir et al. [33], where a network is adapted to the TrueNorth chip to perform gesture recognition.

In the space of SFM and visual odometry, Kim et al. [27] demonstrate that a Kalman filter can reconstruct the pose of the camera and a local map. Rebecq et al. [28] similarly build a 3D map, which they localize from using the events. Zhu et al. [29] use an EM-based feature tracking method to perform visual-inertial odometry, while Rebecq et al. [30] use motion compensation to deblur the event image, and run standard image-based feature tracking to perform visual-inertial odometry. There has also been work by Zhu et al. [31] on using depth as a supervisory signal for learning ego-motion from event streams in an unsupervised way.

D. Fusion Models for VO

Most of the work for VO using fusion based strategies have focused on fusing the IMU and image streams for Visual-Inertial Odometry. VINet [19] uses the correlation version of FlowNet [6] to compose image features across time and then runs the IMU data through an FC layer followed by LSTM layers, before fusing (simple concatenation) them and estimating the pose at time t. Shamwell et al. [18] propose learning an affine representation for the IMU data across time and use this with the learned depth

from monocular images to estimate the pose of the camera. To the best of our knowledge, this is the first work that fuses event and intensity based streams for VO.

III. Event Representation

An event-based camera tracks changes in the log intensity of an image and returns an event where the log intensity changes over a set threshold θ

$$\log(I_{t+1}) - \log(I_t) \geq \theta \tag{1}$$

Each event contains the pixel location of the change, the timestamp of the event and the polarity, as below

$$e = \{x, t, p\} \tag{2}$$

Because of the asynchronous nature of the events, it is not immediately clear what representation of the events should be used in the standard convolutional neural network architecture. Most modern network architectures expect image-like inputs, with a fixed, relatively low, number of channels (recurrent networks excluded) and spatial correlations between neighboring pixels. Therefore, a good representation is key to fully take advantage of existing networks while summarizing the necessary information from the event stream.

In this work, we chose to use a representation of the events in image form. The input to the network is a 4 channel image with the same resolution as the camera. The first two channels encode the number of positive and negative events that have occurred at each pixel, respectively. This counting of events is a common method for visualizing the event stream, and has been shown in Nguyen et al. [12] to be informative in a learning-based framework to regress 6 DoF pose.

However, the number of events alone discards valuable information in the timestamps that encode information about the motion in the image. To tackle this, we encode the pixels in the last two channels as the timestamp of the most recent positive and negative event at that pixel, respectively. This is similar to the Event-based Time Surfaces used in Lagorce et al. [13] and the "time stamp images" used in Park et al. [14]. An example of this kind of image can be found in Figure 3, where we can see that the flow is evident by following the gradient in the image, particularly for closer (faster moving) objects.

While this representation inherently discards all of the timestamps but the most recent at each pixel, we have observed that this representation is sufficient for the network to estimate the correct flow in most regions. One deficiency of this representation is that areas with very dense events and large motion will have all pixels overridden by very recent events with very similar timestamps. However, this problem can be avoided by choosing smaller time windows, thereby reducing the magnitude of the motion.

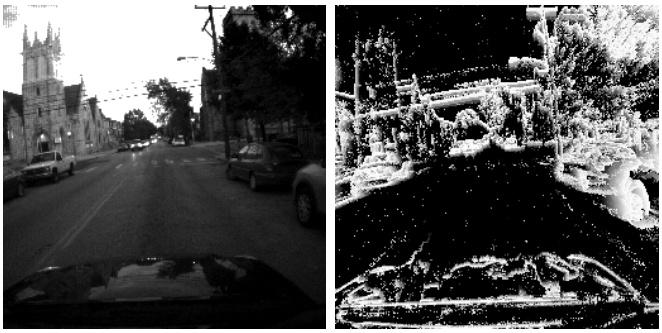


Fig. 3: Example of a timestamp image. Left: Grayscale output. Right: Timestamp image, where each pixel represents the timestamp of the most recent event. Brighter is more recent.

In addition, we normalize the timestamp images by the size of the time window for the image, so that the maximum value in the last two channels is 1. This has the effect of both scaling the timestamps to be on the same order of magnitude as the event counts, and ensuring that fast motions with a small time window and slow motions with a large time window that generate similar displacements have similar inputs to the network.

IV. DeepEvent-VO

In this section, the deep RCNN framework with the fusion strategy realizing VO in an end-to-end fashion is described in detail. It is mainly composed of CNN based feature extraction, feature fusion and RNN based sequential modeling.

A. Architecture of Proposed Model

Most of the current state-of-art CNN architectures, such as VGGNet [4] and GoogLeNet [5] are designed to learn knowledge from appearance and image context. However, these do not serve as good priors for VO, which requires geometric representations. These representations help to derive connections among consecutive image frames, e.g, motion models since VO systems evolve over time and operate on image sequences acquired during movement.

The architecture of the proposed end-to-end VO system is shown in Figure 4, which builds on top of the DeepVO [11] architecture as shown in Figure 2. It takes in a video clip or a monocular image sequence with the delta event-frames between the time frames as input. At each time step, two consecutive images are stacked together to form a tensor for the deep RCNN to learn how to extract motion information and estimate poses. Features are extracted using the model shown in Figure 5 for both intensity and event streams and fused using two `conv` layers. This fused feature is then passed through an LSTM for sequential learning. Each image pair and event-frame yields a pose estimate at each time step through the

network. The VO system develops over time and estimate new poses as images and events are captured.

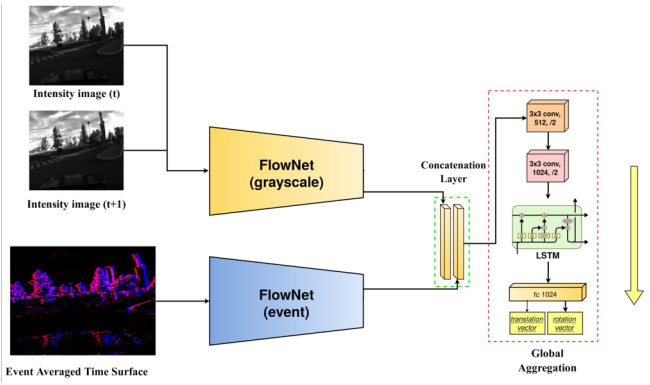


Fig. 4: Our proposed fusion model. This builds on top on DeepVO which can be seen in Figure 8a. Here we utilize two convolutional layers for feature fusion.

The advantage of the RCNN based architecture is to allow simultaneous feature extraction and sequential modeling of VO through the combination of CNN and RNN. More details are given in the sequential sections.

B. CNN Based Feature Extraction

As stated above, we need to learn features that are inherently geometric in nature to facilitate generalization over all possible sequences. Optical Flow across different time steps is one such representation that is very effective. To this end, we adopt the model from [6]. The model image is shown in Figure 5.

The intensity images and the event streams are both passed through this model and the resulting features are then concatenated along the channel dimension and passed through two convolutional layers whose configuration is outlined in Table I. The features fused through are passed to the RNN for sequential learning.

Layer	Receptive Field	Padding	Stride	# Channels
Conv1	3x3	1	1	1024
Conv2	3x3	1	1	1024

TABLE I: Fusion Layers for Features

C. RNN Based Sequential Modeling

To model dynamics and relations among sequences of CNN features, we use a deep RNN for sequential learning. Since RNNs model dependencies of sequences, they are well suited to the VO problem which involves a temporal model (motion model) and sequential data (image sequence). RNNs learn better from intermediate representations such as those generated using the proposed CNN architecture.

RNNs maintain the memory of its hidden states over time and has feedback loops among them, enabling the current hidden state to be a function of previous ones. This enables the RNN to find out connections among the input and previous states in the sequence. Theoretically, RNNs should learn from sequences of arbitrary lengths, but in practice, they suffer from the vanishing gradient problem [8].

Long Short-Term Memory (LSTMs) are capable of learning long dependencies by introducing memory gates and units [9]. It explicitly determines which previous hidden states to be discarded or retained for updating the current state, is expected to learn the motion during pose estimation. An example of an LSTM is shown in Fig. 6.

Given the input x_k at time k and the hidden state h_{k-1} and the memory cell c_{k-1} of the previous LSTM unit, the LSTM updates at time step k according to

$$\begin{aligned}
\mathbf{i}_k &= \sigma\left(\mathbf{W}_{xi}\mathbf{x}_k + \mathbf{W}_{hi}\mathbf{h}_{k-1} + \mathbf{b}_i\right) \\
\mathbf{f}_k &= \sigma\left(\mathbf{W}_{xf}\mathbf{x}_k + \mathbf{W}_{hf}\mathbf{h}_{k-1} + \mathbf{b}_f\right) \\
\mathbf{g}_k &= \tanh\left(\mathbf{W}_{xg}\mathbf{x}_k + \mathbf{W}_{hg}\mathbf{h}_{k-1} + \mathbf{b}_g\right) \\
\mathbf{c}_k &= \mathbf{f}_k \odot \mathbf{c}_{k-1} + \mathbf{i}_k \odot \mathbf{g}_k \\
\mathbf{o}_k &= \sigma\left(\mathbf{W}_{xo}\mathbf{x}_k + \mathbf{W}_{ho}\mathbf{h}_{k-1} + \mathbf{b}_o\right) \\
\mathbf{h}_k &= \mathbf{o}_k \odot \tanh\left(\mathbf{c}_k\right)
\end{aligned} \tag{3}$$

where $\odot$ is element-wise product of two vectors, σ is sigmoid non-linearity, tanh is hyperbolic tangent non-linearity, W terms denote corresponding weight matrices, b terms denote bias vectors, $\mathbf{i}_k, \mathbf{f}_k, \mathbf{g}_k, \mathbf{c}_k$ and $\mathbf{o}_k$ are input gate, forget gate, input modulation gate, memory cell and output gate at time k, respectively.

To model high-level representations and model complex dynamics, two LSTMs are stacked with the hidden states of an LSTM being the input of the other. In our network, each of the LSTM layers has 1000 hidden states. The deep-LSTM outputs a pose estimate at each time step based on the visual features generated from the CNN.

D. Cost Function and Optimization

The proposed RCNN fusion-based VO system can be considered to compute the conditional probability of the poses $\mathbf{Y}_t = (\mathbf{y}_1, \ldots, \mathbf{y}_t)$ given a sequence of the intensity and event images $\mathbf{X}_t = (\mathbf{x}_1, \ldots, \mathbf{x}_t)$ up to a time t in the probabilistic perspective:

$$p(\mathbf{Y}_t \mid \mathbf{X}_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t \mid \mathbf{x}_1, \ldots, \mathbf{x}_t) \tag{4}$$

The modeling and probabilistic inference are performed in the deep RCNN. To find the optimal parameters θ^* for the VO, the DNN maximizes (4):

$$\theta^* = \arg\max_{\theta} p(\mathbf{Y}_t \mid \mathbf{X}_t; \theta) \tag{5}$$

To learn the hyperparameters θ of the DNNs, the Euclidean distance between the ground truth pose $(\mathbf{p}_k, \varphi_k)$ at time k and its estimated one $(\hat{p}_k, \hat{\varphi}_k)$ is minimized. The loss

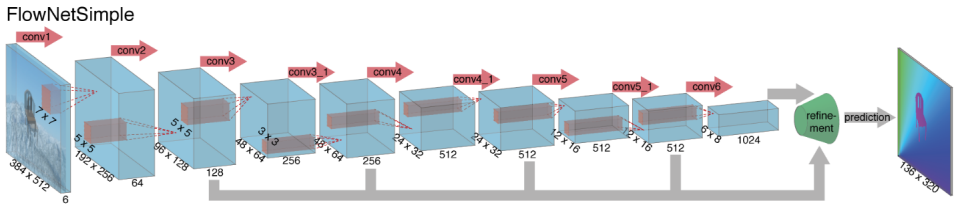


Fig. 5: The model for estimating the optical flow from given inputs. For our model, we do not run the refinement step to extract features.

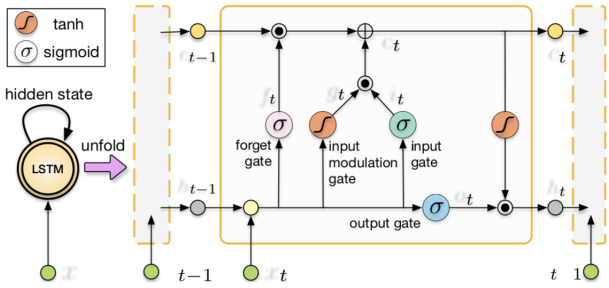


Fig. 6: Folded and unfolded structure of the LSTM unit. $\odot$ and $\oplus$ denote element-wise product and addition of two vectors, respectively.

function is composed of the Mean Square Error (MSE) of all positions $\mathbf{p}$ and orientations φ:

$$\theta* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{t} \|\hat{p}_k - \mathbf{p}_k\|_2^2 + \kappa \|\hat{\varphi}_k - \varphi_k\|_2^2 \tag{6}$$

where $\|.\|$ is 2-norm, κ (100 in experiments) is a scale factor to balance the weights of positions and orientations, and N is the number of samples. The orientation φ is represented by Euler angles rather than quaternion since quaternion is subject to an extra unit contrainst which hinders the optimization problem of DL. We also find that in practice using quaternion degrades the orientation estimate to some extent.

V. Lie Groups for 3D Transformations

A. SO(3) Representation

Elements of the 3D representation group, SO(3), are represented by 3D rotation matrices. Composition and inversion in the group correspond to matrix multiplication and inversion. Because rotation matrices are orthogonal, inversion is equivalent to transposition.

$$\begin{aligned} \mathbf{R} &\in \mathrm{SO}(3) \\ \mathbf{R}^{-1} &= \mathbf{R}^T \end{aligned} \tag{7}$$

The Lie algebra, so(3), is the set of 3x3 skew-symmetric matrices. The generators of so(3) correspond to the derivatives of the rotation around the each of the standard axes, evaluated at the identity:

$$G_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, G_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad G_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{8}$$

An element of so(3) is then represented as a linear representation of the generators:

$$\begin{aligned} \omega &\in \mathbb{R}^3 \\ \omega_1 G_1 + \omega_2 G_2 + \omega_3 G_3 &\in \mathrm{so}(3) \end{aligned} \tag{9}$$

We will simply write $\omega \in so(3)$ as a 3-vector of the coefficients, and use $\omega_\times$ to represent the corresponding skew-symmetric matrix.

B. SO(3) Exponential Maps

The exponential map that takes skew-symmetric matrices to rotation matrices is simply the matrix exponential over a linear combination of the generators.

$$\begin{aligned} \exp(\omega_\times) &\equiv \exp \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \\ &= \mathbf{I} + \omega_\times + \frac{1}{2!}\omega_\times^2 + \frac{1}{3!}\omega_\times^3 + \cdots \end{aligned} \tag{10}$$

Writing the terms in pairs, we have:

$$\exp(\omega_\times) = \mathbf{I} + \sum_{i=0}^{\infty} \left[\frac{\omega_x^{2i+1}}{(2i+1)!} + \frac{\omega_x^{2i+2}}{(2i+2)!} \right] \tag{11}$$

Now we can take advantage of a property of skew-symmetric matrices:

$$\omega_\times^3 = -\left(\omega^T \omega\right) \cdot \omega_\times \tag{12}$$

First extend this identity to the general case:

$$\begin{aligned} \theta^2 &\equiv \omega^T \omega \\ \omega_\times^{2i+1} &= (-1)^i \theta^{2i} \omega_\times \\ \omega_\times^{2i+2} &= (-1)^i \theta^{2i} \omega_\times^2 \end{aligned} \tag{13}$$

Now we can factor the exponential map series and recognize the Taylor expansions in the coefficients:

$$\begin{aligned}\exp(\omega_\times) &= \mathbf{I} + \left(\sum_{i=0}^{\infty} \frac{(-1)^i \theta^{2i}}{(2i+1)!}\right) \omega_\times + \left(\sum_{i=0}^{\infty} \frac{(-1)^i \theta^{2i}}{(2i+2)!}\right) \omega_\times^2 \\ &= \mathbf{I} + \left(\frac{\sin\theta}{\theta}\right) \omega_\times + \left(\frac{1-\cos\theta}{\theta^2}\right) \omega_\times^2 \end{aligned} \tag{14}$$

The above equation is the familiar Rodrigues formula. The exponential map yields a rotation by θ radians around the axis given by ω. Practical implementation of the Rodrigues formula should use the Taylor expansions of the coefficients of the second and third terms when θ is small.

VI. Experiments

A. Dataset & Code

We use the Multi-Vehicle Stereo Event Camera Dataset [22] for our model training and evaluation. The data we used for our experiments can be found on Box. The code can be found on Github and we have submitted a copy along with the report. Additional instructions related for running the code can be seen in the README provided there.

B. Experiments

The dataset consists of 2 outdoor daytime sequences. We trained on 3400 frames from the first sequence, and 5173 frames from the second sequence. We then tested on 1 test sequences consisting of 1043 frames. Due to limitations of the dataset, we faced a shortage in the number of sequences in comparison to dataset requirements of other end-to-end sequence-sequence models. All our models have been written in PyTorch (0.4.1) and Python (3.6). For training, we used Adam optimizer with a learning rate of 1e-3, β_1 value of 0.7, the momentum of 0.9, weight-decay of 5e-6 and a dropout of 0.5 while training. To facilitate the learning of LSTM and prevent exploding gradients, we use a gradient clipping of 20 and set our image height and width fixed at 256 each. To check how the model learns, we are plotting the loss in tensorboard, also we are plotting trajectories after every validation evaluation. For FlowNet, we are using pre-trained weights from [32].

C. Ablation Studies & Results

To test the model, we set up a DeepVO baseline and also train our DeepEvent-VO (DEVO) model using both intensity images and event frames. The results of this can be seen in Figure 8a and 9a for the DeepVO and fusion models respectively.

We conducted several ablation studies to see what different components of our models are learning. This type of study specifically helps us learn which part of the network is performing well and which parts need more refinement in terms of learning strategies.

Firstly, we froze the weights of FlowNet for both DeepVO and DEVO models for the intensity image stream and then trained on the data. The results of this can be seen in Figures 8c and 9c for the DeepVO and DEVO models respectively. For the second experiment, we learned a network from scratch. For this, we also learned a FlowNet based only intensity images (concatenated 2 channel inputs). The results of this can be seen in Figures 8b and 9b for the DeepVO and DEVO models respectively.

The total loss (rotation + translation) of the different models for both training and validation are shown in Figure 7a and 7b. One observation we make is that a lower loss is not indicative of a better model. This may be because the model is stuck in a bad local minimum which may have lower loss values but has not actually learned good representations of the data.

From the results, we can see that DEVO outperforms DeepVO consistently. One thing to notice is that once we freeze the weights the model performs better, an indicator that strong priors help the network to learn better and faster. Also, training from scratch has its own advantages as we are learning features that are inherent to the intensity and event streams. These studies lay a foundation for several future directions discussed in Section VII.

D. Technical Challenges

One of the major challenges that we faced was in terms of collecting data. Since it is very hard to synchronize pose messages and event streams, we had to run our rosbags at 0.05 times the standard rate of play. This resulted in an increased delay in collecting datasets. Further, unlike datasets like KITTI, the dataset only consisted of 2 outdoor day time sequences. This prevented us from further generalizing. Nevertheless, we demonstrate better results for two variants, as per our ablation study.

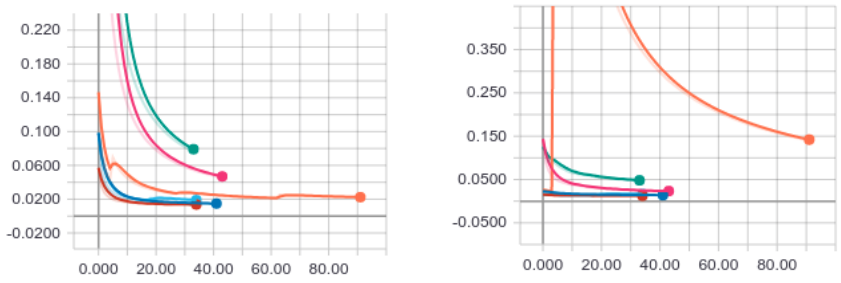
Another major challenge was to find sufficient GPU resources to run our models and also finding sufficient data to train such a large model. Also, since these models are black-box in terms of their learning, it is not possible to debug the different stages independently to facilitate better learning. A more conceptual challenge was in designing the model and setting up different ablation environments for testing the fusion strategy.

E. Timeline Evaluation

We were mostly able to abide by the provided timeline in the initial proposal. One section that took more time than anticipated was getting all the data aligned correctly as described above. Hence, we were not able to experiment as much as we wanted with different fusion strategies or models, given other constraints.

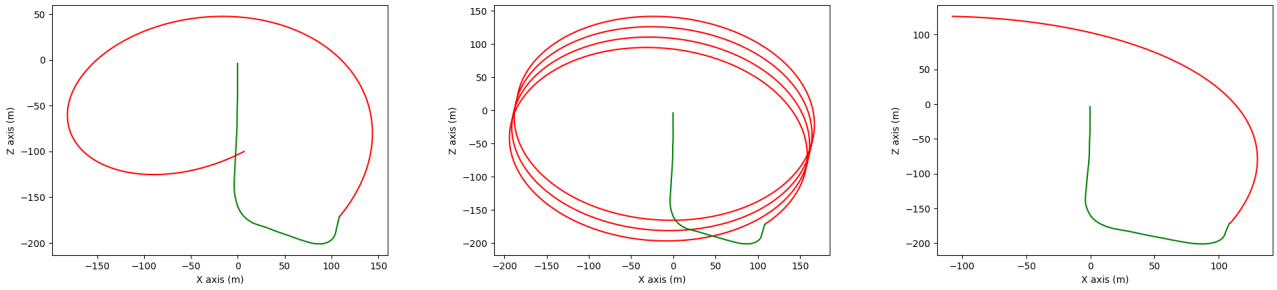
VII. Conclusions And Future Work

In this report, we have shown a simple fusion strategy for event-based streams and intensity images. To our

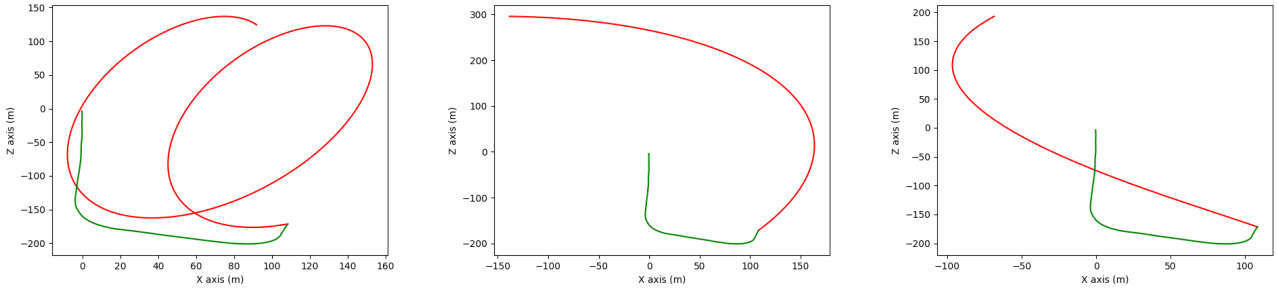


(a) Train Loss (b) Validation Loss

Fig. 7: Train and Validation loss for DEVO (orange), DeepVO (dark blue), DEVO freeze (light blue), DeepVO freeze (red), DEVO scratch (green) and DeepVO scratch (pink). Best viewed in color.



(a) DeepVO (b) DeepVO-Scratch (c) DeepVO-Freeze



(a) DEVO (b) DEVO-Scratch (c) DEVO-Freeze

Fig. 9: Ablation Study Performed over multiple variations and experiments. Figure (c) on row 2 of the figure was found to be the best performing variant. This can be denoted to the fact that the gradients are only propagated over the event branch and aggregation module of the network. As a result, the event branch tries to create a similar representation for flow, resulting in a better trajectory output as compared to its other counterparts. All our results are compared at the 31st epoch of validation. Further, these are not aligned trajectories and purely the output of the network at test time. Here, green is the ground-truth and red is the predicted trajectory. Best viewed in color.

knowledge, this is the first model of its type to fuse event and image streams for end-to-end VO. We have performed ablation studies using different strategies on the proposed model.

One of the biggest drawbacks of the current method is that it is a supervised algorithm, hence it faces a data crunch like most deep learning algorithms. To align with the trend in the community, in the future, we propose trying out self-supervised or unsupervised strategies and also incorporating full stereo information to get rid of the many potential problems in monocular streams. Also, currently, we feel that the network for learning the event-based geometric features are over-parameterized. One other direction for us to think about is the use of different fusion strategies, as we propose a very simple method.

REFERENCES

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Tutorial," IEEE Robotics & Automation Magazine, vol. 18, no. 4, pp. 8092, 2011.

[2] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part II: Matching, robustness, optimization, and applications," IEEE Robotics & Automation Magazine, vol. 19, no. 2, pp. 7890, 2012.

[3] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," IEEE Transactions on Pattern Analysis and Machine Intelligence, to appear.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 19.

[6] A. Dosovitskiy, P. Fischery, E. Ilg, C. Hazirbas, V. Golkov, P. van derSmagt, D. Cremers, T. Broxet al., "Flownet: Learning optical flow with convolutional networks," inProceedings of IEEE international conference on Computer Vision (ICCV). IEEE, 2015, pp. 27582766

[7] R. Hartley and A. Zisserman, "Multiple view geometry in computer vision." Cambridge university press, 2003.

[8] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," 2016, book in preparation for MIT Press.

[9] W. Zaremba and I. Sutskever, "Learning to execute," arXiv preprint arXiv:1410.4615, 2014.

[10] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks." in Proceedings of International Conference on Machine Learning (ICML), vol. 14, 2014, pp. 17641772

[11] Wang, Sen, Ronald Clark, Hongkai Wen, and Niki Trigoni. "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks." In 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 2043-2050. IEEE, 2017.

[12] Nguyen, Anh, Thanh-Toan Do, Darwin G. Caldwell, and Nikos G. Tsagarakis. "Real-Time 6DOF Pose Relocalization for Event Cameras with Stacked Spatial LSTM Networks." arXiv preprint arXiv:1708.09011 (2017).

[13] Lagorce, Xavier, Garrick Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. "Hots: a hierarchy of event-based time-surfaces for pattern recognition." IEEE transactions on pattern analysis and machine intelligence 39, no. 7 (2017): 1346-1359.

[14] Park, Paul KJ, Baek Hwan Cho, Jin Man Park, Kyoobin Lee, Ha Young Kim, Hyo Ah Kang, Hyun Goo Lee, et al. "Performance improvement of deep learning based gesture recognition using the spatiotemporal demosaicing technique." In 2016 IEEE International Conference on Image Processing (ICIP), pp. 1624-1628. IEEE, 2016.

[15] Li, Ruihao, Sen Wang, Zhiqiang Long, and Dongbing Gu. "Undeepvo: Monocular visual odometry through unsupervised deep learning." In 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 7286-7291. IEEE, 2018.

[16] Ummenhofer, Benjamin, et al. "Demon: Depth and motion network for learning monocular stereo." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[17] Zhu, Alex Zihao, et al. "Robustness meets deep learning: An end-to-end hybrid pipeline for unsupervised learning of egomotion." arXiv preprint arXiv:1812.08351 (2018).

[18] Shamwell, E. Jared, Sarah Leung, and William D. Nothwang. "Vision-Aided Absolute Trajectory Estimation Using an Unsupervised Deep Network with Online Error Correction." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.

[19] Clark, Ronald, et al. "Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem." Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[20] Rohan Ghosh, Abhishek Mishra, Garrick Orchard, and Nitish V Thakor. Real-time object recognition and orientation estimation using an event-based camera and CNN. In Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE, pages 544547. IEEE, 2014

[21] Diederik Paul Moeys, Federico Corradi, Emmett Kerr, Philip Vance, Gautham Das, Daniel Neil, Dermot Kerr, and Tobi Delbru ck. 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 18. IEEE, 2016.

[22] Alex Zihao Zhu, Dinesh Thakur, Tolga Ozaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The Multi Vehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception. arXiv preprint arXiv:1801:10202, 2018.

[23] Jonathan Binas, Daniel Neil, Shih-Chii Liu, and Tobi Delbru ck. DDD17: End-To-End DAVIS Driving Dataset. CoRR, abs/1711.01458, 2017.

[24] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. Frontiers in neuroscience, 9, 2015.

[25] Yuhuang Hu, Hongjie Liu, Michael Pfeiffer, and Tobi Delbruck. DVS benchmark datasets for object tracking, action recognition, and object recognition. Frontiers in Neuroscience, 10, 2016.

[26] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A Low Power, Fully Event-Based Gesture Recognition System. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 72437252, 2017.

[27] 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 349364. Springer, 2016

[28] 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):593600, 2017.

[29] 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 53915399, 2017

[30] 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

[31] Zhu, Alex Zihao, et al. "Unsupervised Event-based Learning of Optical Flow, Depth, and Egomotion." arXiv preprint arXiv:1812.08156 (2018).

[32] Pinard Clement, "Flownet", Github, Last accessed on May 08. 2019.

[33] Arnon Amir, Brian Taba, David Berg, Timothy Melano,Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak,Alexander Andreopoulos, Guillaume Garreau, MarcelaMendoza, et al.A Low Power, Fully Event-BasedGesture Recognition System. InProceedings of the IEEEConference on Computer Vision and Pattern Recognition, pages 72437252, 2017.

[34] Rebecq, Henri, Timo Horstschaefer, and Davide Scaramuzza. "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization." British Machine Vis. Conf. (BMVC). Vol. 3. 2017.

[35] Mueggler, Elias, Basil Huber, and Davide Scaramuzza. "Event-based, 6-DOF pose tracking for high-speed maneuvers." 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014.