Exploring Event Cameras: Event Trajectory Generation, Neuromorphic Data Simulation and Event Based Flow Estimation

Abhinav Gupta, Amogh Tiwari and Avinash Sharma
IIIT Hyderabad

Abstract

Event Cameras are causing a paradigm shift in the computer vision world due to multiple advantages over conventional cameras, including high temporal resolution, high dynamic range and low power consumption. This makes it an ideal choice for capturing fast motion. In this work, we explore various works in this domain that help us understand the concept of event-based data. We implement a trajectory generation network as proposed in EventCap [13], which helps track events in 2D space in an asynchronous manner. We also explore how we can simulate neuromorphic data from conventional high fps videos, in the absence of an event camera. Lastly, we also take a look at representing event-based data as images, for deep learning and CNN applications. In particular, we use a self-supervised flow network to estimate optical flow for data exhibiting human motion.

1. Introduction

An event camera, also known as a neuromorphic camera or a dynamic vision sensor, is an imaging sensor that responds to local changes in brightness. Event cameras do not capture images using a shutter as conventional cameras do. Instead, each pixel inside an event camera operates independently and asynchronously, reporting changes in brightness as they occur, and staying silent otherwise.

These revolutionary sensors that work radically differently from standard cameras. Instead of capturing intensity images at a fixed rate, event cameras measure changes of intensity asynchronously, in the form of a stream of events, which encode per-pixel brightness changes. In the last few years, their outstanding properties such as asynchronous sensing, no motion blur and high dynamic range have led to exciting vision applications, with very low-latency and high robustness. These cameras promise to unlock robust and high- speed perception in situations that are currently not accessible to standard cameras, such as tracking features in the blind time between two frames.

An event camera tracks changes in the log intensity of an image, and returns an event whenever the log intensity changes over a set threshold θ. This event representation is given in [14], and similar notations are used throughout the literature.

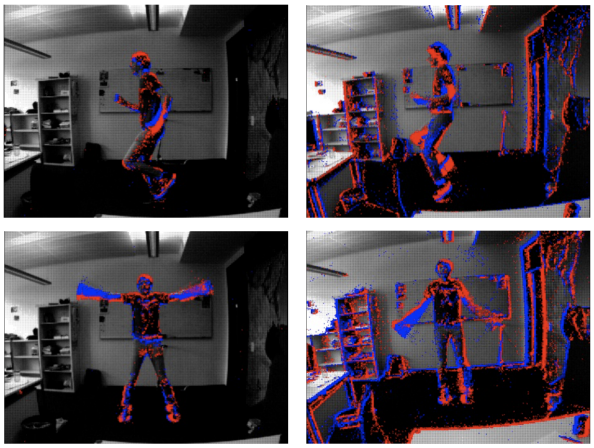

Figure 1. Visualising events from the color event dataset. The event is triggered when a change in the logarithmic intensity exceeds the threshold. Here, the events are coloured according to polarity. The blue indicates an increase in brightness, and red indicates decrease. We use the ROS package *dvs_renderer* to visualise this event stream.

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

Each event contains the pixel location of the change, timestamp of the event and polarity. Here, the event e is represented by the pixel location x, the time of the occurrence of the event t and the polarity p, which represents the sign of the brightness change. The visualisation of events can be seen in 1.

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

In this study, we explore the domain of event-based data, in a quest to formulate and solve a novel problem statement in fast human motion capture. We wish to ultimately attempt to efficiently capture motion from low fps videos using simulated events. Our study can be broken down into three categories, and is summarised as follows:

Figure 2. The *People* scene from the Color Event Dataset [12], which uses the DAVIS346 camera to capture both, the color image frames, as well as the event stream. The image on the left is the intensity image captured. We visualise the corresponding event stream through the image on the right.

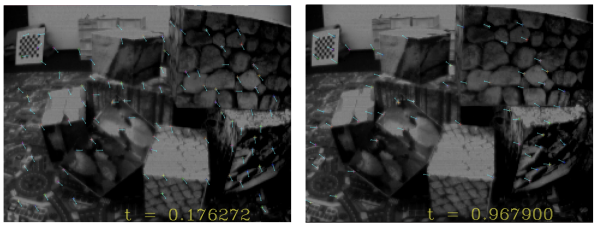

Figure 3. The results of asynchronous photometric 2D feature tracking using events and frames [3] on the *Boxes_6DoF* scene from the event camera dataset [9].

- We explore the generation of asynchronous event trajectories, as proposed in [13]. We track the events in 2D space in an asynchronous manner and reconstruct the continuous event trajectories between each adjacent intensity images.
- We attempt to simulate event streams from high fps data collected from conventional cameras, by using the Event Camera Simulator [8].
- We use EV-Flownet [14], a self-supervised network to estimate represent events as images, and estimate the optical flow using only the event information.

2. Literature Review

With the recent rise in interest in event-based data, many works have been published in the last few years that exploit events for various computer vision and robot perception tasks. An exhaustive list of papers in this domain can be found in this repository maintained by the RPG lab at ETH Zurich. In this section, we primarily review EventCap [13], which serves as our inspiration for capturing fast human motion.

EventCap [13] is the first approach for the 3D capture of high speed human motions using a single event camera. It is a hybrid and asynchronous motion capture algorithm which leverages both, the event stream as well as the intensity images coming from event camera, in a joint optimisation framework. The inputs into EventCap are the intensity images, the event stream and the textured mesh with skeleton rig. This method combines model-based optimisation and CNN based human pose detection to capture high frequency motion. The advantages of using an event camera is high temporal resolution, low data bandwidth and low power, since every pixel operates independently and asynchronously.

The Event Camera Dataset [9] is the world's first collection of datasets with event-based vision. This dataset includes intensity images, inertial measurements, and ground truth from a motion-capture system. This dataset consists of a variety of indoor and outdoor scenes, captured using a DAVIS240C. The Color Event Camera Dataset [12] is the first color event dataset, featuring 50 minutes of footage with both color frames and color events from the Color-DAVIS346. These event cameras are manufactured by iniVation, famous for creating neuromorphic vision systems. All the data is in binary format as *.rosbag* files, compatible with the Robot Operating System (ROS). Also, it is crucial to note that event cameras do not provide absolute intensity measurements, they measure only changes of intensity. For visualising events, we use *dvs_renderer* [7, 5, 2], a ROS package which helps plot the event data with the intensity image frames. A scene from the CED dataset along with the event visualisation in shown in 2.

3. Asynchronous Event Trajectory Generation

We explore the event trajectory module proposed in [13]. A single event is not of much use to us because it has no structural information. Hence, tracking based on isolated events is not robust. We wish to extract spatio-temporal information from the event stream, for the k-th batch (in time interval $[t_k, t_{k+1}]$) between adjacent intensity images $I[k]$ and $I[k+1]$. To do this, we track the photometric 2D features in an asynchronous manner, which was proposed in [3]. We test this approach out on a scene from the event dataset [9], as shown in 3.

3.1. Feature Tracking

We use [3] for robust tracking of photometric 2D features, resulting in sparse event trajectories. This method leverages the complementarity of event cameras and standard cameras to track visual features with low latency. It first extracts features on intensity images from the standard camera, and subsequently tracks them asynchronously using events. Hence, it makes use of the best of the two data. Event cameras excel at sensing motion at very low latency (only 1 ms), but do not output intensity measurements. Standard cameras generate the intensities, but at a really high latency (10-20 ms). This is the complementar-

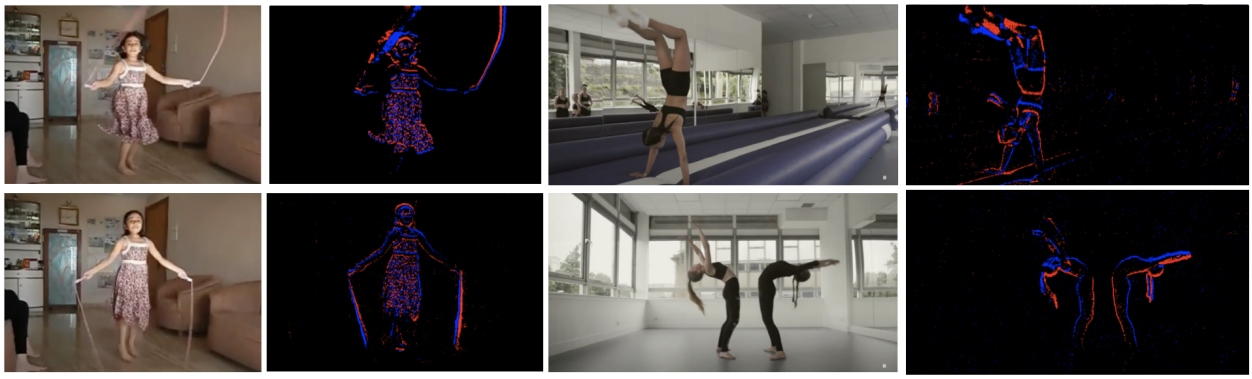

Figure 4. Simulating events from conventional high fps videos using ESIM [11]. Here, we simulate a 240 fps video of a girl skipping and a video of ballet dancers shot at 120 fps, collected from YouTube. When the girl is skipping (column 1), her surroundings are completely stationary. Hence, none of those pixels register an event, and the simulated results are blacked out for those pixels in the frame (column 2). It is interesting to note that as the woman performs an acrobatic stunt (column 3), her reflection in the mirror is also simulated by the event camera (column 4), as there is movement detected in that part of the frame, leading to a change in the logarithmic brightness of the pixels.

ity which [3] exploits. The DAVIS camera functions on this exact principle, comprising of an asynchronous event-based sensor and a standard frame-based camera in the same pixel array. The frames provide a photometric representation that does not depend on motion direction and the events provide low-latency updates.

Conventional methods to feature tracking cannot track in the blind time between consecutive frames, and are expensive because they process information from all pixels, even in the absence of motion in the scene. The method proposed in [3] works by extracting corners in frames and subsequently tracking them using only events. This allows us to take advantage of the asynchronous, high dynamic range and low-latency nature of the events to produce feature tracks with high temporal resolution. The event camera helps in filling up the blind time between consecutive frames. This is the first principled method that uses raw intensity measurements directly, based on a generative event model within a maximum-likelihood framework. As a result, it produces feature tracks that are both more accurate and longer than the state of the art, across a wide variety of scenes.

3.2. Sharpening Intensity Images

The feature tracking method [3] relies on sharp intensity images for gradient calculation. However, the intensity images from the event dataset suffer from severe motion blur due to the fast motion. To counter this problem, [13] uses the event-based double integral (EDI) model proposed in [10] to sharpen the adjacent intensity images $I[k]$ and $I[k+1]$ before extracting 2D features from them.

This Event-based Double Integral (EDI) model [10] can reconstruct a high frame-rate, sharp video from a single blurry frame and its event data. So we use this method to sharpen our intensity images, before tracking the photometric features. The feature tracking can also drift over time. To reduce this tracking drifting, [13] applies the feature tracking method in both, the forward and reverse directions. These directional tracking results are then stitched by associating the closest backward feature position to each forward feature position at the central timestamp. Each batch is sliced into many tracking frames, in order to achieve motion capture at very high tracking frame rate (like 1000 fps).

4. Neuromorphic Data Simulation

Event cameras are rare and expensive sensors. The DAVIS346 sensor costs six thousand dollars, which only a tiny fraction of research groups can afford. The event camera simulator [11] was introduced in this spirit, in order to simulate a large amount of reliable event data from conventional camera images. The key component of the simulator is an adaptive rendering scheme that only samples frames when necessary, through a tight coupling between the rendering engine and the event simulator. Instead of choosing an arbitrary rendering frame-rate, and sampling frames uniformly across time at the chosen frame-rate, [11] proposes to sample frames *adaptively*, adapting the sampling rate based on the predicted dynamics of the visual signal. Uniform sampling fails in fast-varying parts of signals. The adaptive sampling strategy is able to extract more samples in fast varying regions and can thus simulate events more accurately.

We choose videos from YouTube that are shot using conventional cameras at high fps, and feed them into the event camera simulator, which is able to accurately simulate event event information from these images. Our results are summarised in 4. We simulate a 240 fps video of a girl skipping and a video of ballet dancers shot at 120 fps, collected from

YouTube. We can notice from the qualitative results that the events are simulated only when there is motion in the frame, that is, when there is a change in the logarithmic intensity of that particular pixel. For instance, when the girl is skipping, her surroundings are completely stationary. Hence, none of those pixels register an event, and the simulated results are blacked out for those pixels in the frame. Similarly, in the ballet dancing results, it is interesting to note that as the woman performs an acrobatic stunt, her reflection in the mirror is also simulated by the event camera, as there is movement detected in those pixels.

5. Event Based Flow Estimation

We now move on to the estimation of optical flow using event cameras. Event cameras show great promise for accurate flow predictions as compared to conventional cameras, which would suffer in situations of high-speed motions and high dynamic range scenes. Recent deep learning based approaches for flow estimation have shown promising results - like FlowNet2.0 [4]. However, these approaches keep frame-based images in mind. Moreover, they are supervised over well annotated frame images and such wealth is not available for events. We use EV-FlowNet [14], which is a self-supervised deep network learning pipeline for optical flow estimation for event based cameras.

5.1. EV-FlowNet

This work [14] introduces an image based representation of a given event stream, that is, we represent the event information as an image, which is then fed into a self-supervised convolutional neural network as the sole input. The corresponding grayscale intensity images captured from the same camera at the same time as the events, are used to only supervise the network and provide a loss function for training. Hence, the network is able to accurately predict the flow from only the events.

We cannot use networks such as FlowNet2.0 [4] due to the asynchronous output of the event-based camera. This does not easily fit into the synchronous, frame-based inputs expected by image-based paradigms. Moreover, there is a lack of labeled training data which necessary for supervised training methods. EV-FlowNet [14] overcomes these issues, since it utilises the event information and can be trained in a self-supervised fashion. Hence, the flow can be predicted given only a set of events and the corresponding grayscale images generated from the same camera, like the DAVIS 240, circumventing the need for expensive labeling of data. These images along with the self-supervised loss function, which is adapted from UnFlow [6], are sufficient for the network to learn to predict accurate optical flow from events alone. So no supervision is required from the ground truth flow.

Figure 5. Examples of timestamp images, a 4 channel image representation of events as proposed in [14]. Left: Grayscale images. Right: The corresponding timestamp images, where each pixel represents the timestamp of the most recently occurred event. A brighter pixel denotes the event has occurred more recently.

5.2. Image Representation of Events

Most modern CNN architectures expect image-like inputs with fixed number of channels spatial correlations between neighbouring pixels. This paper proposes a novel image-based representation of an event stream, which fits into any standard image-based neural network architecture. The event stream is summarized by an image with channels representing the number of events and the latest timestamp at each polarity at each pixel. Hence, we also call this a "timestamp image". This compact representation preserves the spatial relationships between events, while maintaining the most recent temporal information at each pixel. Examples of timestamp images are shown in 5.

Perhaps, the most complete representation that preserves all of the information in each event would be to represent the events as a (n $\times$ 4) matrix, where each column contains the information of a single event. However, this does not directly encode the spatial relationships between events that is typically exploited by convolutions over images. The number of events alone discards valuable information in the timestamps that encode information about the motion in the image. Incorporating timestamps in image form is a challenging task.

One possible solution would be to have k channels, where k is the most events in any pixel in the image, and stack all incoming timestamps. However, this would result in a large increase in the dimensionality of the input. So instead, this paper represents the events as a 4 channel image. The first two channels encode the number of pos-

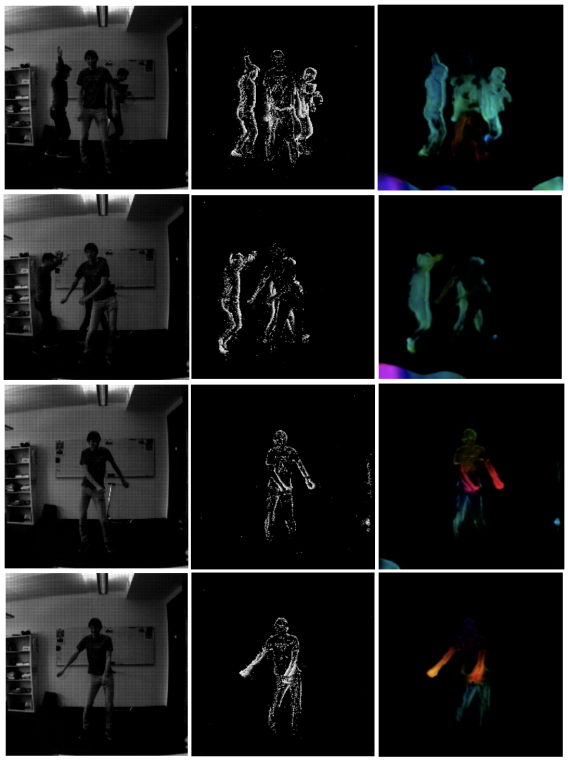

Figure 6. Predicting the optical flow using EV-Flownet [14] on the *People* scene from the event dataset [12]. Left: Grayscale frame images. Middle: corresponding timestamp images. Right: Flow predictions from the network.

itive and negative events that have occurred at each pixel, respectively. The other two channels of this image encode the timestamp of the most recent positive and negative event at that pixel, respectively.

5.3. Caveats of using a timestamp image

While this representation inherently discards all of the timestamps but the most recent at each pixel, this representation is sufficient for the network to estimate the correct flow in most regions, for most datasets. But one huge 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. This problem may be avoided by choosing smaller time windows, thereby reducing the magnitude of the motion. Alternatively, one may choose a richer representation, like a 6-channel image, which would encode more information about the events.

5.4. Results on CED People Dataset

We use a model of EV-Flownet pretrained on the Multi Vehicle Stereo Event Camera (MVSEC) Dataset [15]. This mostly comprises of stereo event data collected from cars, motorbikes and hexacopters and the model has been trained for such driving scenes, without any humans. However, it is able to generalise decently well to human data as well, as evident from the qualitative results in 6. We alter the code a bit since the CED dataset [12] is monocular while MVSEC [15] is a stereo dataset . We obtain accurate flow results for this dataset and it works well for single as well as multiple human figures in the frame.

5.5. Conclusion and Future Work

In this study, we explore the recent domain of event cameras, and study its usefulness in capturing fast human motion. In particular, we generate asynchronous event trajectories as proposed in [13], we simulate event streams from high fps videos using [11] and use a self supervised network [14] for estimating the optical flow using only events. We wish to formulate a novel problem statement and solution for human motion capture.

One problem we would like to attack is the capture of motion from low fps videos. We wish to implement the complete method proposed in [13], but test it for low fps conventional videos by simulating event streams from these videos using [11]. Another pertinent problem we could explore is motion deblurring along with body part segmentation using event cameras. We can place a 240 fps camera and a 30 fps camera adjacently and collect data exhibiting fast human motion. We can simulate events from the 240 fps video feed and simultaneously deblur the data obtained from the 30 fps camera using the event stream. We can further use the ground truth obtained from the 240 fps camera and perform semantic segmentation using [1] for evaluation.

References

[1] I. Alonso and A. C. Murillo. Ev-segnet: Semantic segmentation for event-based cameras. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 5

[2] C. Brändli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck. A 240 × 180 130 db 3 µs latency global shutter spatiotemporal vision sensor. *Solid-State Circuits, IEEE Journal of*, 49:2333–2341, 10 2014. 2

[3] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza. EKLT: Asynchronous, photometric feature tracking using events and frames. *Int. J. Comput. Vis.*, 2019. 2, 3

[4] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 4

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

[6] S. Meister, J. Hur, and S. Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, New Orleans, Louisiana, Feb. 2018. 4

[7] E. Mueggler, B. Huber, and D. Scaramuzza. Event-based, 6-dof pose tracking for high-speed maneuvers. 09 2014. 2

[8] E. Mueggler, H. Rebecq, G. Gallego, T. Delbrück, and D. Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *CoRR*, abs/1610.08336, 2016. 2

[9] E. Mueggler, H. Rebecq, G. Gallego, T. Delbrück, and D. Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *CoRR*, abs/1610.08336, 2016. 2

[10] L. Pan, C. Scheerlinck, X. Yu, R. Hartley, M. Liu, and Y. Dai. Bringing a blurry frame alive at high frame-rate with an event camera. pages 6813–6822, 06 2019. 3

[11] H. Rebecq, D. Gehrig, and D. Scaramuzza. ESIM: an open event camera simulator. *Conf. on Robotics Learning (CoRL)*, Oct. 2018. 3, 5

[12] C. Scheerlinck, H. Rebecq, T. Stoffregen, N. Barnes, R. E. Mahony, and D. Scaramuzza. CED: color event camera dataset. *CoRR*, abs/1904.10772, 2019. 2, 5

[13] L. Xu, W. Xu, V. Golyanik, M. Habermann, L. Fang, and C. Theobalt. Eventcap: Monocular 3d capture of high-speed human motions using an event camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 1, 2, 3, 5

[14] 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, 4, 5

[15] 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. 5