Jointly learning visual motion and confidence from local patches in event cameras

Daniel R. Kepple[1], Daewon Lee[1], Colin Prepsius[1], Volkan Isler[1], Il Memming Park[2], and Daniel D. Lee[1]

[1] Samsung AI Center - New York
{d.kepple,daewon.l,c.prepscius,ibrahim.i,daniel.d.lee}@samsung.com
[2] Department of Neurobiology and Behavior, Stony Brook University
memming.park@stonybrook.edu

Abstract. We propose the first network to jointly learn visual motion and confidence from events in spatially local patches. Event-based sensors deliver high temporal resolution motion information in a sparse, non-redundant format. This creates the potential for low computation, low latency motion recognition. Neural networks which extract global motion information, however, are generally computationally expensive. Here, we introduce a novel shallow and compact neural architecture and learning approach to capture reliable visual motion information along with the corresponding confidence of inference. Our network makes a prediction of the visual motion at each spatial location using only local events. Our confidence network then identifies which of these predictions will be accurate. In the task of recovering pan-tilt ego velocities from events, we show that each individual confident local prediction of our network can be expected to be as accurate as state of the art optimization approaches which utilize the full image. Furthermore, on a publicly available dataset, we find our local predictions generalize to scenes with camera motions and the presence of independently moving objects. This makes the output of our network well suited for motion based tasks, such as the segmentation of independently moving objects. We demonstrate on a publicly available motion segmentation dataset that restricting predictions to confident regions is sufficient to achieve results that exceed state of the art methods.

1 Introduction

Individual pixels in an event-based camera report only when there is an above-threshold change in the log light intensity in its field of view [1–3]. Such an operation can be performed extremely quickly, without influence from neighboring pixels, and with minimal influence of the absolute light intensity [4]. This creates inherent advantages, e.g., low latency vision without requiring high power or being constrained to uniform, well-lit environments. This makes event-based cameras attractive for motion based tasks which require precise timing and are ideally invariant under extreme changes in lighting conditions[5–9]. Other advantages of event based vision, such as its potential for vision with low compu-

tational cost owing to its sparse, asynchronous output, can only be realized with the development of novel data processing techniques.

Here, we suggest a neural network which takes advantage of event-based vision's potential for low computation in the context of visual motion. Rather than computing dense optical flow, we extend the philosophy of event-based sensing to sparsely recover visual motion information. We simultaneously predict both a region's local flow and its reliability for visual motion predictions. Downstream processes, such as camera pose estimation or motion segmentation, can then use the sparse, confident visual motion information. Similar to biological vision systems [10–14], we compute this visual motion as the projection of optical flow on preferred axes, rather than as a true optical flow.

Our solution therefore has two parts: prediction of the visual motion in each small spatial region, and prediction of the confidence of each region's visual motion prediction. We demonstrate that because our solution is fully local, it can be learned under uniform visual motion conditions and generalize to make reliable predictions in dramatically different conditions, such as those with independently moving objects or unseen motions.

Our contributions are:

1. **Compact visual motion network**: Our formulation has two orders of magnitude fewer parameters than networks which solve similar problems [15, 16]. This makes our network attractive for employment in systems with limited resources.
2. **Accurate local visual motion predictions**: Our network produces confident, fully local predictions which can be expected to as accurate as methods which use the entire image .
3. **Improved performance in downstream tasks**: We show that our sparse predictions still enable motion segmentation and camera pose estimation that competes with state of the art methods.
4. **Novel training approach**: Despite training on the limited domain of pan/tilt camera motions in front of a computer monitor, we show that our network generalizes to realistic datasets with challenging lighting and full 6DOF camera motion.
5. **New dataset**: We have collected a large-scale dataset with 10,000 diverse scenes with precisely controlled known camera movements in static environments.

2 Related work

In this paper, we consider the problem of recovering visual motion in a scene with an event-based system. This is often considered in the context of optical flow, for which event-based neural networks have been proposed with some success [15, 17, 18, 16]. Such networks, however, are deep and require heavy computation to provide a dense optical flow. Furthermore, in part due to the challenge of getting labelled optical flow in dynamic scenes, these networks, with the exception of [17], cannot handle the presence of independently moving objects.

In the realm of optimization, there are also approaches to capture visual motion with event-cameras [19, 20, 17, 21]. Many of these utilize the approach of Contrast Maximization [19]. This approach takes advantage of the edge detection of event-sensors, and the assumption that flows are uniform on small spatiotemporal scales. Events in local regions are warped back in time according to a proposed velocity, and the velocity whose warped image most accurately reconstructs an edge is identified. While this approach requires less computation than a deep network, warped images are still costly to compute. Furthermore, extension of this approach to scenes with independently moving objects requires computing a warped image for each object, compounding computational costs [21].

In keeping with the philosophy of event sensors, we aim for a network which can get visual motion information quickly and at low computational cost. Rather than sacrificing the accuracy of our predictions, we will identify local regions in which accurate visual motion predictions can be cheaply computed. Towards this goal, we propose a novel training framework to enable a network to selectively learn from a large number of examples where some are assumed to be uninformative of the target. Our formulation is most related to Mixture of Experts models [22] and attention networks [23].

In traditional vision, the idea of limiting the domain of one predictor to subregions determined by another has had success in the form of Region Proposal Networks (RPN) [24–26]. Our proposal differs significantly from these approaches by necessity. RPNs are trained using ground truth labels – that is, the true locations of objects in the images are known and used in training. In our case, the ground truth reliability of a subregion in predicting optical flow projections is unknown. Therefore, we developed a novel, joint training procedure to address this problem.

3 Method

3.1 Event Cameras

Unlike traditional cameras which communicate the light intensity at every pixel synchronously according to a frame rate, event-based cameras report the list of pixel locations whose light intensity has changed, the sign of that change, and precise time the change is detected [1–3]. More formally, event-based sensors communicate a stream of events $\mathcal{S} = \{s_i\}_{i=1}^{K},\ s_i = [x_i, y_i, p_i, t_i]$, where x_i, y_i are the spatial indices from the $M \times N$ resolution pixel array and t_i is the time of the i^{th} event, ordered such that $\forall i < K,\ t_i \leq t_{i+1}$ where K is the total number of events. p_i is the *polarity* for event i. Polarity denotes the sign of the change in the light intensity resulting in event s_i.

3.2 Preprocessing

Given the precise temporal resolution and asynchronous nature of a typical event-camera, it is generally the case that any instant will contain only one event. It is therefore necessary to consider past events to perform even basic computer vision tasks. Standard approaches include using time windows [27,

17], batching a fixed number of events together [15], or using "time surfaces", which are monotonically decreasing functions applied to the elapsed time since the last event at each spatial pixel [28, 29, 18].

Our approach is to smoothly integrate the past with multiple time scales which avoids explicitly counting events and allows for event-based processing. Consider a single pixel with coordinate (x, y) detecting polarity p events. Let us define a *leaky integrating pixel* with a voltage variable $v(t)$, similar to membrane potentials in spiking neural networks (SNN) [30], as the first-order filtering of input events represented as a sequence of Dirac delta functions $I(t) = \sum_i \delta(t-t_i)$ for all events:

$$v(t) = \int_{-\infty}^{t} I(s) \cdot e^{-(t-s)/\tau} \mathrm{d}s = \sum_{t_i \leq t} e^{-(t-t_i)/\tau} \tag{1}$$

where $\tau > 0$ is the time constant.

We refer to the voltage of any single pixel as $v_{xy}^{\tau p}(t)$ where the superscript τ indicates the time constant of that neuron and $p \in \{+, -\}$ separates events of different polarity into different images. Subscripts x, y are the spatial indices of the corresponding dynamic vision sensor (DVS) pixel. We refer to the image of all leaky integrating pixels with the same τ and polarity at a time t as $\mathcal{V}^{\tau p}(t)$. In the supplementary material, we show an example of these images for two different τ and both polarities.

Intuitively, an image of leaky integrating pixels accumulates signal in pixels with recent events. The voltage at any given pixel is bounded below by 0 and unbounded from above. The choice of τ then controls the depth of the memory of past events, with large τ approximating an event counter, and small τ providing timing information of events occurring within a short past.

We use two time constants, motivated by delay lines utilized in biological motion detectors [10–14]. We will refer to these decay time constants as τ_{slow} and τ_{fast}. In the supplementary material, we provide an argument for the selection of these time constants in order to predict velocities in a specified range. For the rest of the paper, we use $\tau_{\text{slow}} = 20$ ms and $\tau_{\text{fast}} = 10$ ms.

3.3 Visual motion model and assumptions

Visual motion in traditional vision can be defined as a vector field of pixel translations between two image frames. As event-based cameras do not have temporal frames, we will define visual motion for events. Let $[u_{x_i y_i}(t_i), v_{x_i y_i}(t_i)]$ be the visual motion at (x_i, y_i) at time t_i. Assuming nonzero $[u_{x_i y_i}(t_i), v_{x_i y_i}(t_i)]$, noiseless observation, and constant motion, an event $s_i = [x_i, y_i, p_i, t_i]$ will produce an event s_j at time $t_j > t_i$ at location $x_j = x_i + (t_j - t_i) u_{x_i y_i}(t_i)$ and $y_j = y_i + (t_j - t_i) v_{x_i y_i}(t_i)$.

Instead of just considering flows $[u_{xy}, v_{xy}]$ corresponding to camera axes x and y, we will consider $[u_{xy}^p, v_{xy}^p]$ where p enumerates $\theta_p \in \{0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8}\}$:

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + (t_j - t_i) \begin{bmatrix} \cos(\theta_p) & -\sin(\theta_p) \\ \sin(\theta_p) & \cos(\theta_p) \end{bmatrix}^T \begin{bmatrix} u_{x_i y_i}^p \\ v_{x_i y_i}^p \end{bmatrix} \tag{2}$$

Our approach will assume events within small spatial neighborhoods experience uniform visual motion [31]. Furthermore, we assume that pure pan-tilt egomotion generates uniform visual motion over the image. Such an assumption is justified in a camera with a moderate viewing angle, such as our Samsung Gen 3 DVS's 45 degree view, and pan/tilt egomotions constrained within a 20 degree cap. Under this assumption, the visual motion at any location in the image is equal to the pan-tilt of the camera.

3.4 Network architecture

Our approach uses two convolutional networks, one for predicting visual motion, and the other for predicting confidence (Figure 1). We design both convolutional networks such that each is equivalent to a fully connected subnetwork applied at each small spatial window. This is achieved in both networks by one 15×15 convolution followed by 1×1 convolutions. This subnetwork design is relevant for our training approach, and we will refer to the fully connected subnetworks as f and g for the visual motion and confident networks respectively (Figure 1).

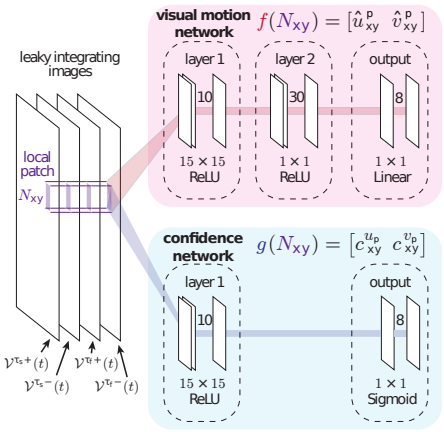

Fig. 1: Convolutional architecture implementing many parallel fully connected networks. Computation path for a local patch N_{xy} is highlighted. The convolutions applied to this patch are mathematically equivalent to the fully connected network f (in red), and g (in blue).

Local, fully connected visual motion network f The input to our visual motion subnetwork f is the neighborhood of a point (x, y) in the four leaky integrating images (described in section 3.2):

$$N_{xy}(t) = \left\{ v^{\tau_{slow}-}_{(x+a)(y+b)}(t), v^{\tau_{fast}-}_{(x+a)(y+b)}(t), \quad v^{\tau_{slow}+}_{(x+a)(y+b)}(t), v^{\tau_{fast}+}_{(x+a)(y+b)}(t) \right\}_{a,b=-7}^{7}$$

The subnetwork is then a function $f\colon \mathbb{R}^{4\times 15^2} \to \mathbb{R}^9$,

$$f\left(N_{xy}(t)\right) = [\hat{u}^p_{xy}(t), \hat{v}^p_{xy}(t)]_p,$$

that estimates the visual motion for projection p, i.e $[\hat{u}^p_{xy}(t), \hat{v}^p_{xy}(t)]_p$, from the information accumulated in the input images around (x, y). This prediction is ill-posed due to the aperture problem [32], observation noise [1], as well as the unknown correspondence and the typical sparseness of events. For this reason, we expect that f will typically only be able to read out velocity in a subset of N_{xy}'s from the full leaky integrating image. On this subset, however, a shallow network may be sufficient to accurately predict visual motion. We anticipate this and propose a neural network architecture with three fully connected layers (two ReLU layers followed by a linear readout; Figure 1).

Confidence network *g* Subnetwork g will be trained to identify whether the prediction from f on the same window can be expected to be accurate. As with f, we can summarize this network as a function:

$$g(N_{xy}(t)) = [c^{u_p}_{xy}(t), c^{v_p}_{xy}(t)],$$

where $c^{u_p}_{xy}(t), c^{v_p}_{xy}(t) \in [0, 1]$ are the confidences for visual motion predictions $\hat{u}^p_{xy}(t)$ and $\hat{v}^p_{xy}(t)$ respectively. Importantly, this means confidence is considered for each projection separately.

g is a two layer fully connected network, with the first nonlinearity being a ReLU and the second a sigmoid. The sigmoid function enables a binary interpretation of the output of this network while maintaining differentiability. The schematic of this network is shown in Figure 1.

3.5 Supervised training local networks from global signal

The DVS-COCO dataset To train the aforementioned pair of networks, we use the DVS-COCO pan-tilt dataset. In this dataset, the Samsung Gen 3 HVGA (320x480 resolution) dynamic vision sensor (DVS) has been mounted on a motorized pan-tilt stage and set in front of a computer monitor (see supplement for schematic). Random saccade-like velocities move the stage up to 75 degrees/second while the images of the Microsoft COCO dataset are presented on the screen.[33] Each one of the 10,000 selected images are presented for 15 seconds and for an average of 30 saccades. For a full description of the DVS-COCO dataset, please see the supplementary material.

We will refer to the angular velocity of the camera as $[\omega_{\text{pan}}, \omega_{\text{tilt}}]$. With our approximation that pure pan/tilt motion produces globally uniform visual motion, it follows that our goal is to train f to accurately predict $[\omega_{\text{pan}}, \omega_{\text{tilt}}]$.

Mixture of Inputs training Even with ground truth visual motion, we do not know a priori which spatial neighborhoods contain sufficient information to predict that visual motion. For example, regions without events cannot predict flow. In traditional computer vision, this is analogous to training a Region Proposal Network (RPN) without ground truth bounding boxes. Without labelled data, we can't use approaches like Faster RCNN [24, 26, 25]. We therefore developed a novel approach, which we will call Mixture of Inputs training.

We define the loss function $\mathcal{L}_f$ of the network f at time t on spatial region $N_{xy}(t)$ to be the squared error of predictions weighted by confidence g:

$$\mathcal{L}_f\left(f(N_{xy}(t)), g(N_{xy}(t)), \omega_{\text{pan}}(t), \omega_{\text{tilt}}(t)\right) = \sum_p \left(\begin{bmatrix} c_{xy}^{u_p}(t) \\ c_{xy}^{v_p}(t) \end{bmatrix}^T \left(\begin{bmatrix} \hat{u}_{xy}^p(t) \\ \hat{v}_{xy(t)}^p \end{bmatrix} - R_{\theta_p} \begin{bmatrix} \omega_{\text{pan}}(t) \\ \omega_{\text{tilt}}(t) \end{bmatrix} \right) \right)^2 \tag{3}$$

where R_{θ_p} is the Euclidean rotational matrix for angle θ_p.

Our convolutional architecture then batches all spatial neighborhoods in an image, and thus the local loss over a single image, $\mathcal{L}_{\text{local}}$ can be computed:

$$\mathcal{L}_{\text{local}} = \sum_{x,y} \mathcal{L}_f\left(f(N_{xy}(t)), g(N_{xy}(t)), \omega_{\text{pan}}(t), \omega_{\text{tilt}}(t)\right)$$

Note that $\mathcal{L}_{\text{local}}$ cannot be used to train g. This is because $\mathcal{L}_f$ has a trivial global minimum for where $g(N_{xy}(t)) = 0$. Therefore, to train g we use a separate loss function which takes global information into account. We define confidence normalization terms $Z_{u_p}(t) = \sum_{x,y} c_{xy}^{u_p}(t)$ and $Z_{v_p}(t) = \sum_{x,y} c_{xy}^{v_p}(t)$. We will refer to the weighted average of optical flow projection predictions as the global prediction of the camera angular velocity in reference frame p, $[\hat{\omega}_{\text{pan}}^p(t), \hat{\omega}_{\text{tilt}}^p(t)]$:

$$\begin{bmatrix} \hat{\omega}_{\text{pan}}^p(t) \\ \hat{\omega}_{\text{tilt}}^p(t) \end{bmatrix} = \sum_{x,y} R_{\theta_p}^T \begin{bmatrix} \frac{c_{xy}^{u_p} \hat{u}_{xy}^p}{Z_{u_p}}(t) \\ \frac{c_{xy}^{u_p} \hat{v}_{xy}^p}{Z_{v_p}}(t) \end{bmatrix}$$

Now we define the loss function for g, $\mathcal{L}_g$, to be the squared error between this global optical flow prediction and the rotational velocity:

$$\mathcal{L}_g = \sum_p \left(\left(\hat{\omega}_{\text{pan}}^p(t) - \omega_{\text{pan}}(t)\right)^2 + \left(\hat{\omega}_{\text{tilt}}^p(t) - \omega_{\text{tilt}}(t)\right)^2 \right)$$

Mathematically, $\mathcal{L}_g$ is similar to the gating network in Mixture of Expert (MoE) models [22]. In MoE, many networks compete to make predictions and the gating function selects the best predictors. Here, however, we have the same network and many different samples. Our "gating" network (the confidence network g), selects the best inputs, not the most suitable expert. We therefore call this approach *Mixture of Inputs*. Another distinction is that we do not have any explicit competition between inputs, although there is implicit competition for the training error signal through $\mathcal{L}_g$.

Training protocol We subsample each image by taking a randomly located 150×150 window from the full 320×480 Samsung Gen3 HVGA DVS image. We randomly select a batch of 80 time points from our training set (full description of training and testing set in supplementary material). For each time and corresponding 150×150 window, the four leaky integrated images $\mathcal{V}^{\tau p}$ are calculated with events histories of 3τ. Parameters of f are updated according to $\mathcal{L}_f$, and the parameters of g according to $\mathcal{L}_g$. We schedule our training rate with ADAM [34], using parameters $\beta_1 = 0.9, \beta_2 = 0.999, \eta = .01$.

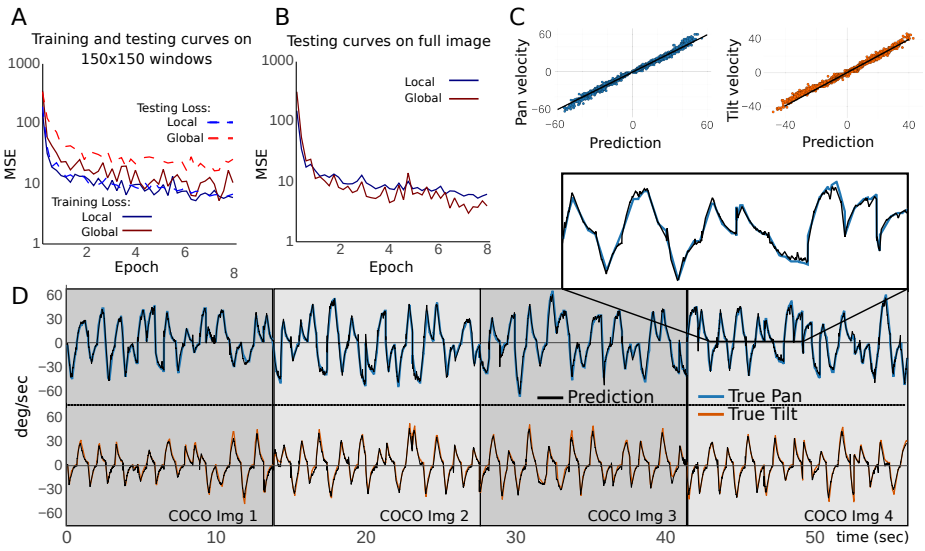

Fig. 2: Training and Testing on the DVS-COCO Benchmark. (A) Global, $\mathcal{L}_g$, and local, $\mathcal{L}_f$, loss functions over the course of training using 150×150 windows. (B) Test set mean squared error (MSE) of global predictions (red), local predictions (blue) using the full 320×480 image (C) Scatter plots between predicted and true velocities over the whole test set. (D) Continuous predictions on four testing videos.

3.6 Evaluation Metrics and Comparisons

DVS COCO test set performance We evaluate our network by both its local and global predictions. Because we train f and g on 150×150 windows, we also compute the test loss on 150×150 windows. We evaluate our network's best possible global prediction using the whole image, which will be referred to as the global prediction with global error $\text{MSE}_{\text{global}}$.

Our local MSE is computed with the squared errors of each local prediction, $\mathcal{E}^p_{xy}$:

$$\mathcal{E}^p_{xy} = \left(\begin{bmatrix} \hat{u}^p_{xy} \\ \hat{v}^p_{xy} \end{bmatrix} - R_{\theta_p} \begin{bmatrix} \omega_{\text{pan}} \\ \omega_{\text{tilt}} \end{bmatrix} \right)^2, \text{MSE}_{\text{local}} = \sum_{x,y,p} \frac{1}{C} \begin{bmatrix} c^{u_p}_{xy} \\ c^{v_p}_{xy} \end{bmatrix}^T \mathcal{E}^p_{xy} \tag{4}$$

Where $C = \sum_{x,y,p} (c^{u_p}_{xy} + c^{v_p}_{xy})$

All networks are trained for 10 epochs. We report the average MSE over the last epoch for both $\text{MSE}_{\text{local}}$ and $\text{MSE}_{\text{global}}$.

Ablations Mixture of Inputs training utilizes two loss functions, $\mathcal{L}_g$ and $\mathcal{L}_f$. While the confident local loss $\mathcal{L}_f$ has a trivial global minimum for g, $\mathcal{L}_g$ can be used to train f. To understand the contribution of $\mathcal{L}_f$ we train a network this way and refer to it as the "$\mathcal{L}_g$ only" network.

To evaluate the contribution of our confidence network, we train a network using a heuristic confidence instead. This heuristic provides a binary confidence measure, 1 for predictions with above average number of events in the last 3τ and 0 for those below. Intuitively, this heuristic will identify neighborhoods that contain more than just events from noise. Visual motion networks trained with this confidence signal will be referred to as "Mean confidence network".

Contrast Maximization (CM) We also compare our network's performance with that of an optimization approach called Contrast Maximization (CM) [19, 20]. We briefly describe this in the supplementary material for completeness. CM is expected to provide strong results for pan/tilt conditions. We performed a brute force search to optimize the time window size for CM on the DVS-COCO dataset and found 60ms.

To compare local predictions, we use CM with a 15×15 spatial window size. We also extend their method to use both our network's confidence scores as well as heuristic confidence metrics designed to utilize the information available to their optimization. In particular, we use the mean and variance of the 15×15 windows. We include a full description of these confidence models in the supplementary material.

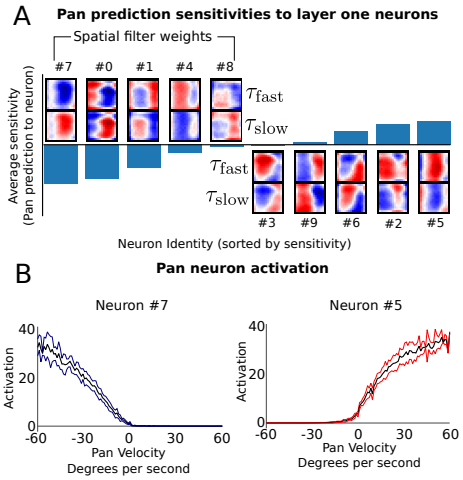

Fig. 3: Analysis of the visual motion network f. Panel A shows the sensitivity of pan predictions to the activity of all ten neurons in layer 1. Accompanying each sensitivity is a heatmap of the spatial weights of that neuron. There are two spatial filters for each neuron, one for each time constant. Panel B shows the activation of individual neurons to pan velocity (see supplementary material for tilt) in regions which have confidence greater than 0.1. Mean activation across the DVS-COCO testing set is shown in black, with colored lines to show one standard deviation above and below the mean. Color indicates neurons which are selective for negative velocity (blue) or positive velocity (red).

Extreme event dataset and motion segmentation We use a set of public test sequences called the Extreme Event Dataset (EED) [6]. The EED features independently moving objects and 6DOF motion in challenging lighting scenarios, including a strobe light. This dataset serves as a benchmark for event-based motion segmentation algorithms. It comes with hand labeled ground truth bounding boxes of independently moving objects in the scene.

To segment a scene, we cluster the confident flows output from our network. We compute a distance between each confident flow, e.g. $u_{x_i y_i}$ and $u_{x_j y_j}$:

$$d_{ij} = \sqrt{(u_{x_i y_i} - u_{x_j y_j})^2 + (x_i - x_j)^2 + (y_i + y_j)^2} \tag{5}$$

We threshold d_{ij} at 10, and cluster all graph connected flows together. This parameter is flexible, and was selected to be on the order of one of our kernel filters. To compare with [6, 21, 35] we use the success rate defined in [6], which is the percentage of bounding boxes in a sequence overlapping 50% or more with a proposed segmentation.

4 RESULTS

4.1 Visual motion and confidence jointly learned

In Figure 2, we demonstrate training and test performance on the DVS-COCO velocity recovery task. In general, our network is able to accurately recovers test set velocities. Over the entire testing set of 6000 clips, our $\text{MSE}_{\text{global}}$ is 4.5 $(\text{degrees/sec})^2$.

As our training and testing sets sample times sparsely from 1000 training and 300 testing videos using truncated histories, we also demonstrate the ability of our network to make continuous predictions with full time histories on four testing videos (Figure 2).

4.2 Network f learns direction selective neurons

Our network learns to combine leaky integrated images by computing their difference, as is expected from biological models [10, 11]. This can be seen qualitatively in Figure 3, which shows learned filters for the two τ are opposite in sign. The median correlation between τ_{fast} weights and τ_{slow} is $-.83$.

The high sensitivity learned kernels for pan prediction shown in Figure 3A are polarized horizontally. This suggests that diagonal edges are being ignored for pan predictions in the unrotated reference frame, which is consistent with a solution to the aperture problem. In Figure 3B, we see those neurons are direction selective, with an approximately linear relationship with speed in their selected direction.

4.3 Confidence network identifies edges and phase

The behavior of our confidence network is shown in Figure 4. For predictions in the unrotated reference frame, our confidence network outputs vertical and horizontal lines. In general, this strategy enables identifying regions x, y in which either $\hat{u}_{xy}$ or $\hat{v}_{xy}$ are confident, but not both. This is unlike corner detectors, such as the Harris detector [36], which can be used for both components of optical

flow. That our network does not look for corners could be due to the relative sparsity of corners compared to edges. Our network is trained on 150×150 windows, and such corners might not always be present.

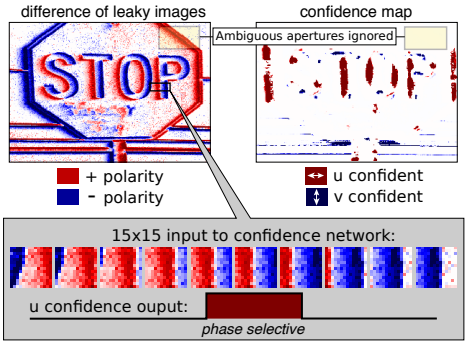

Fig. 4: Confidence network in action. Top left: difference of leaky integrated images is shown for a scene from the DVS-COCO dataset. Top right: the confidence output from the corresponding input. No confidence is shown in white. Bottom: sequential 15x15 windows of the input are shown above the confidence output of that region. The confidence network is selective to the phase of the edge orthogonal to the direction of visual motion.

In Figure 4(right), we show the optical flow predictions with confidence greater than zero. Only optical flow predictions with the proper phase, i.e. the phase corresponding to predictions with the same sign as the ground truth, are selected. Thus our optical flow network need only make accurate predictions given a single phase.

4.4 Ablation results

Two τ leaky integrating image ablation Networks trained with a single τ leaky integrating image are unable to learn velocity. Such networks output only zero, and therefore have a MSE of about 700 $(\text{degrees/s})^2$.

Learned confidence ablation Using the heuristic confidence described in section 3.6 learns to predict pan/tilt with significantly higher MSE (47.6 $(\text{degree/s})^2$). This filter is agnostic to the orientation of edges, and thus suffers due to the aperture problem.

Mixture of input ablation Networks trained using only the loss function $\mathcal{L}_g$ learn accurate global optical flow (MSE 9.61), but local predictions are often inaccurate (MSE 54.02). This is because, although the weighted average is constrained by the loss $\mathcal{L}_g$, the variance of that distribution of predictions is not. More surprisingly, constrained global predictions are worse than those trained with the local loss $\mathcal{L}_f$. Together, this suggests Mixture of Inputs training helps in both the identification and learning of accurate predictions.

4.5 Comparison results

Global Contrast maximization Global contrast maximization [20], iterating over all possible optical flow vectors and taking as evidence all events in the image over a time window, can be expected to very accurately recover velocity in our pan-tilt setup. Indeed on the DVS-COCO testing set, this method recovers velocities with a low MSE of 6.0. While our network's globally weighted average prediction made better predictions overall (4.5), our local predictions, each made using only a single 15×15 window, are on average as accurate as the global CM predictions (6.1 vs 6.0).

Table 1: DVS-COCO Velocity prediction

	Global MSE $(\text{degrees/sec})^2$	Local MSE $(\text{degrees/sec})^2$
Ours	**4.5**	**6.1**
Mean Confidence	47.1	48.52
$\mathcal{L}_g$ Only	11.6	54.05
CM Global	6.0	N/A
CM Local (Mean)	14.9	(69.7)
CM Local (Var)	10.4	(129.7)
CM Local (Our confidence)	28.6	193.7

Local Contrast maximization Local contrast maximization, using the same 15×15 windows as our network, provides comparisons to our local predictions. Without any kind of confidence metric, the average of all such local predictions are, as one would expect, not very accurate. Therefore we extend their method to include heuristic confidences which aim to cover the information available to the CM calculation (see supplementary material for more detail).

Using global averages of local CM weighted by the mean, variance, and our network's own confidence, improves CM prediction accuracy (10.4 MSE for variance weighting, 14.9 for mean, and 28.6 using our confidence weights). Identifying accurate local CM predictions is difficult, resulting in higher local predictions errors (ours: 6.1, contrast maximization with mean: 69.7, variance: 129.7, our confidence: 193.7). The challenge of identifying accurate local predictions in other approaches demonstrates the importance of joint training in our network.

Motion segmentation of Extreme Event Dataset The Extreme Event Dataset [6] is comprised of several scenarios designed to be challenging for traditional cameras. In particular, it features a moving DAVIS240B with independently moving, small objects with speeds around 600 pixels/s in dark, uneven, and in stobe lighting. The task for this dataset is to segment these moving objects, despite possible occlusion from netting or other objects.

Our network relies on identifying regions for which cheap visual motion calculations can be reliable. There is no guarantee that every object will contain

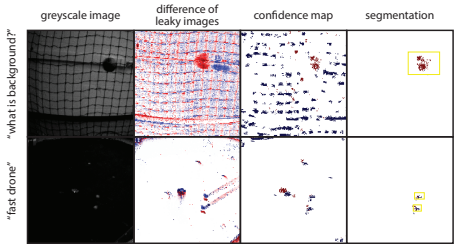

Fig. 5: Motion segmentation examples from the extreme event dataset [6].

Table 2: EED success percentage

Sequence	SOFAS [35]	Mitrokhin[6]	Stoffregen[21]	Ours
Fast moving drone	87.89	92.78	96.30	**100.00**
Multiple objects	46.15	87.32	**96.77**	93.3
Lighting variation	0.0	84.52	80.51	**97.40**
What is background?	22.08	89.21	**100.00**	**100.00**
Occluded sequence	80.00	90.83	92.31	**100.00**

these confident regions, and the purpose of evaluating on the EED is that it contains small, fast moving drones and occluded objects which will challenge our confidence network. Furthermore, as there is no accompanying training data for this task, our network must generalize from training on a static, well-lit computer screen using a different DVS with pan/tilt motions. The EED, by contrast, contains 3D translational camera motions and Z-axis rotions and challenging lighting.

From Figure 5 and our segmentation results in Table 2, we demonstrate that our network generalizes to these challenging conditions. Our network is able to reliably produce confident flows on the background as well as the often small and quick independently moving objects in the scene. Furthermore, these confident flows are accurate enough to separate the flows of these objects from the flows due to camera motion. Our performance is strong particularly on the strobe light sequence. This is perhaps due to our confidence network rejecting regions and times which are greatly affected by the sudden changes in lighting. We show relatively weak performance on the multiple drones sequences, showing the limitations of our confidence network.

Computational comparisons In Table 3, we show the inference latency using a GeForce GTX 1080 of our network and EV-Flownet, a deep network which produces dense optical flow [15]. From this comparison we see that our network performs significantly faster on low resolutions (5 times speed on 240x180) and saturates with increasing number of pixels to 3.5 times faster on HD resolution.

From [21] the processing latency of Contrast Maximization's image warp is 1ms for 4000events on a scene with egomotion and one independently moving object using a 240x180 DVS. [21] does not mention the number of events used

Table 3: Inference latency

Resolution	240x180	320x480	1080x1920
Ours	12ms	25ms	500ms
EV-Flownet[15]	65ms	125ms	1800ms

for a calculation, only that the set of events span the order of milliseconds. If we assume a motion generating events in three percent DVS pixels per millisecond, then a 240x180 DVS will generate on the order of 10000 events and we approximate the latency of [21] on the order of 10ms. Importantly, the latency increases linearly with number of objects moving and with the resolution of the camera. Our approach does not use a motion model and is invariant to the number of objects in the scene. Thus, in the absence of moving objects and with low resolution cameras, [21] will have a lower latency than our approach.

In our network, we have 19,000 parameters, whereas deep networks producing dense optical flow predictions in [17] and [15] use 2 million and 14 million respectively. In memory limited systems, high parameter networks may require slow, sequential loading of subnetworks from external memory. Deep networks such as [17] and [15] also use skip connections, requiring storage of past activations. In our approach, only one layer's activations are used in any calculation, meaning previous layer's activation can be forgotten. Contrast Maximization based approaches, however, need only store events and are therefore are the lowest memory approach.

5 Discussion

In this work we proposed a low-parameter network which makes accurate motion predictions with low latency for event cameras. Our novel training approach enables the joint learning of a spatially local prediction network and its confidence using a global signal. We show that local predictions generalize well to untrained conditions such as challenging lighting and scenes with ego-motion and independently moving objects for motion segmentation. We suggest our approach is valuable in resource limited systems where accurate motion information is necessary, such as those arising in robotics.

Future work will investigate furthering the computational efficiency of our network, such as using multiple smaller convolutions, or dilation, as our large convolution size is the current computational bottleneck of our approach. The fully-local nature of our predictions also enables the network's stride to be adjusted without compromising predictions. This suggests that not only could the resolution of our predictions be adjusted to meet a systems resources, one could also dynamically adjust the stride to use more or less computation in response to environmental conditions.

While our network was trained on a simple motion task, our global training signal could come from an IMU, where local predictions and depth information are combined in an ego-motion model for training.

References

1. G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *arXiv preprint arXiv:1904.08405*, 2019.
2. D. Drazen, P. Lichtsteiner, P. Häfliger, T. Delbrück, and A. Jensen, "Toward real-time particle tracking using an event-based dynamic vision sensor," *Exp. Fluids*, vol. 51, no. 5, p. 1465, Nov. 2011.
3. P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE_J_JSSC*, vol. 43, no. 2, pp. 566–576, 2008.
4. H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
5. J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6713–6719.
6. A. Mitrokhin, C. Fermuller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018. [Online]. Available: http://dx.doi.org/10.1109/IROS.2018.8593805
7. 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*, vol. 3, no. 3, pp. 2032–2039, 2018.
8. A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. D. Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, J. Kusnitz, M. Debole, S. Esser, T. Delbruck, M. Flickner, and D. Modha, "A low power, fully Event-Based gesture recognition system," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 7388–7397.
9. F. Barranco, C. Fermuller, Y. Aloimonos, and T. Delbruck, "A dataset for visual navigation with neuromorphic methods," *Front. Neurosci.*, vol. 10, p. 49, Feb. 2016.
10. W. E. Reichrdt, "Autocorrelation, a principle for the evaluation of sensory information by the central nervous system," *Sensory Communication*, pp. 303–317, 1961.
11. E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Am. A*, vol. 2, no. 2, pp. 284–299, Feb. 1985.
12. K. H. Britten, M. N. Shadlen, W. T. Newsome, and J. A. Movshon, "Responses of neurons in macaque MT to stochastic motion signals," *Vis. Neurosci.*, vol. 10, no. 6, pp. 1157–1169, 1993.
13. E. P. Simoncelli and D. J. Heeger, "A model of neuronal responses in visual area MT," *Vision Res.*, vol. 38, no. 5, pp. 743–761, Mar. 1998.
14. A. Borst, J. Haag, and D. F. Reiff, "Fly motion vision," *Annu. Rev. Neurosci.*, vol. 33, pp. 49–70, 2010.
15. A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-Supervised optical flow estimation for event-based cameras," Feb. 2018.
16. 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," *CoRR*, vol. abs/1809.08625, 2018. [Online]. Available: http://arxiv.org/abs/1809.08625

17. A. Mitrokhin, C. Ye, C. Fermuller, Y. Aloimonos, and T. Delbruck, "EV-IMO: Motion segmentation dataset and learning pipeline for event cameras," Mar. 2019.
18. R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Trans Neural Netw Learn Syst*, vol. 25, no. 2, pp. 407–417, Feb. 2014.
19. G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," *CoRR*, vol. abs/1804.01306, 2018. [Online]. Available: http://arxiv.org/abs/1804.01306
20. G. Gallego and D. Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 632–639, Apr. 2017.
21. T. Stoffregen, G. Gallego, T. Drummond, L. Kleeman, and D. Scaramuzza, "Event-based motion segmentation by motion compensation," 2019.
22. R. A. Jacobs, M. I. Jordan, S. J. Nowlan, G. E. Hinton *et al.*, "Adaptive mixtures of local experts." *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.
23. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.
24. R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
25. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
26. W. Liu, D. Anguelov, Dragomirand Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.
27. E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *Int. J. Rob. Res.*, vol. 36, no. 2, pp. 142–149, Feb. 2017.
28. A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "HATS: Histograms of averaged time surfaces for robust event-based object classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1731–1740.
29. X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, "HOTS: A hierarchy of Event-Based Time-Surfaces for pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1346–1359, Jul. 2017.
30. W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.
31. S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, Feb. 2004.
32. H. Wallach, "Über visuell wahrgenommene bewegungsrichtung," *Psychologische Forschung*, vol. 20, no. 1, pp. 325–380, 1935.
33. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 740–755.
34. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014.
35. T. Stoffregen and L. Kleeman, "Simultaneous optical flow and segmentation (sofas) using dynamic vision sensor," 2018.

36. C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.