Efficient Meshflow and Optical Flow Estimation from Event Cameras

Xinglong Luo[1,4]*, Ao Luo[2,4]*, Zhengning Wang[1]†, Chunyu Lin[3], Bing Zeng[1], and Shuaicheng Liu[1,4]†
[1]University of Electronic Science and Technology of China
[2]Southwest Jiaotong University [3]Beijing Jiaotong University [4]Megvii Technology
{luoboom@std.,zhengning.wang@,eezeng@,liushuaicheng@}uestc.edu.cn
aoluo@swjtu.edu.cn cylin@bjtu.edu.cn

Abstract

In this paper, we explore the problem of event-based meshflow estimation, a novel task that involves predicting a spatially smooth sparse motion field from event cameras. To start, we generate a large-scale High-Resolution Event Meshflow (HREM) dataset, which showcases its superiority by encompassing the merits of high resolution at 1280×720, handling dynamic objects and complex motion patterns, and offering both optical flow and meshflow labels. These aspects have not been fully explored in previous works. Besides, we propose Efficient Event-based MeshFlow (EEMFlow) network, a lightweight model featuring a specially crafted encoder-decoder architecture to facilitate swift and accurate meshflow estimation. Furthermore, we upgrade EEMFlow network to support dense event optical flow, in which a Confidence-induced Detail Completion (CDC) module is proposed to preserve sharp motion boundaries. We conduct comprehensive experiments to show the exceptional performance and runtime efficiency (39× faster) of our EEMFlow model compared to recent state-of-the-art flow methods. Our code is available at https://github.com/boomluo02/EEMFlow.

1. Introduction

Meshflow, a spatially smooth sparse motion field, represents motion vectors exclusively at mesh vertices [28, 49], which has been widely applied in various vision applications, such as image alignment [29, 30, 34], video stabilization [26, 27, 45] and high dynamic range (HDR) imaging [31, 46]. This motion representation combines the benefits of optical flow and global homography, effectively reducing redundancy in motion information and computational costs, while also accommodating non-rigid motions beyond single-plane movements. However, meshflow estimation on RGB images often encounter challenges under scenarios such as low-light and rapid motions. This is due to the loss of fine image texture details and motion blurs.

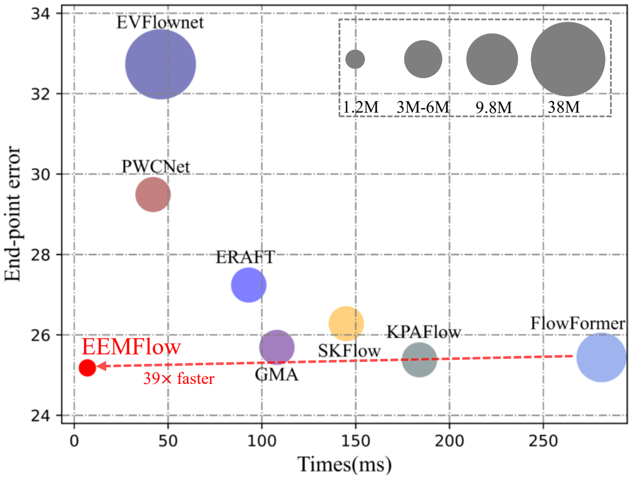

Figure 1. Comparison of computational overhead and accuracy metrics. The x-axis represents inference time, while the y-axis corresponds to the end-point error. The size of each circle indicates the number of model parameters. Lower values for all metrics are considered better.

In contrast, event cameras are well-suited for motion estimation under such situations [7, 11]. Equipped with bio-inspired vision sensors, event cameras can generate sequences of events with microsecond accuracy triggered by changes in log intensity. In particular, when a change is detected in a pixel, the camera returns an event in the form $e = (x, y, t, p)$ immediately, where x, y stands for the spatial location, t refers to the timestamp in microseconds, and p is the polarity of the change, either positive or negative. The advantages of high temporal resolution and high dynamic range make event cameras highly effective for analyzing dynamic scenes.

In this work, we study a new problem that estimates meshflow from event camera data. To start, we create a large-scale **H**igh-**R**esolution **E**vent **M**eshflow (**HREM**) dataset, which contains 20k train and 8k test samples. We build 100 virtual scenes in Blender to render the dataset,

[*] Equal contribution. [†] Corresponding Author.

which can provide accurate physically-based events along with dense meshflow label pairs. Based on the dataset, we further propose an **E**fficient **E**vent-based **M**esh**F**low (**EEMFlow**) network to estimate high-resolution meshflow from event data. Unlike recent flow networks relying on recurrent refinement structure [11, 17, 42], our network is developed on an encoder-decoder architecture with multi-scale global optimizing scheme, which can produce full-resolution meshflow with minimal computational overhead. Specifically, our EEMFlow achieves efficiency by employing the lightweight encoder, building cost volume with dilated feature correlation, and using group shuffle convolutions during decoding. We select recent top-ranked flow networks, including ERAFT [11], SKFlow [41], KPAFlow [32], and FlowFormer [15], which is trained on our HREM dataset. The results show that our approach achieves state-of-the-art performance while maintaining a fast inference speed of 142.9 FPS, outperforming previous works by a relatively large margin (see Fig. 1).

Additionally, we empirically demonstrate that the proposed new pipeline has the capability to effectively handle various motion patterns. Its lightweight design and runtime efficiency further contribute significantly to the field of optical flow estimation. Specifically, we refine the optical flow progressively during decoding using the coarse to fine residual approach. A **C**onfidence-induced **D**etail **C**ompletion (**CDC**) module is proposed to preserve motion boundary details during flow upsampling. We also perform comparative experiments with recent event flow networks [21, 25, 33] to illustrate its superiority. The enhanced flow network is referred to as **EEMFlow+**, demonstrating impressive performance on the reputable DSEC [11] online test benchmark, with the fastest inference speed reaching 39.2 FPS. Our contributions are summarized as:

- We build **HREM**, the first event-based meshflow dataset, superior in the highest resolution at 1280×720, dynamic scenes, complex motion patterns, as well as physically correct accurate events paired with meshflow and optical flow labels.
- We propose **EEMFlow**, which achieves SOTA performances when compared to top-ranked optical flow networks when trained on our meshflow dataset. Moreover, it achieves inference speed of 142.9 FPS, which is 25.5 to 38.7 times faster than compared methods.
- We propose **CDC**, a confidence-induced detail completion module that empowers EEMFlow to make a meaningful contribution to the optical flow community. The upgraded model achieving SOTA performance when compared to representative methods, while also boasting the fastest inference speed to date.

2. Related Work

2.1. Image-based Meshflow Warping

Meshflow is a lightweight and spatially smooth sparse motion field with motions only located at mesh vertices [28]. Meshflow is different from dense optical flow, where optical flow estimates motions of every pixel of an image while meshflow only concentrates on the global motion, rejecting motions of any dynamic contents. Meshflow is also different from a global homography, where local motions from nonplanar depth variations can be well reflected. Mesh-based methods proofs to be effective in various applications, such as high dynamic range (HDR) imaging [46], burst image denoising [47], video denoising [38], image/video stitching [23, 35] and video stabilization [45]. It is worth noting that directly downsampling the optical flow may yield flow fields of the same resolution as meshflow, but they differ significantly from meshflow and perform poorly in terms of warping effects [49]. Direct downsampling ignores motion outliers from different dynamic objects and the global consistency, which can be effectively addressed by meshflow through motion propagation.

2.2. Event-based Optical Flow Estimation

Optical flow estimation from event cameras has received significant attention in recent years. Early approaches, such as [1], could only estimate optical flows at the regions where events are triggered. Recently, deep methods can estimate optical flows from event data, even for the regions without triggered events. For example, EV-FlowNet [52] learns event and flow labels in a self-supervised manner by minimizing photometric distances of grey images acquired by DAVIS [2]. Various event representations, including EST [8] and Matrix-LSTM [3], have been explored, and different network structures, such as Spike-FlowNet [19], LIF-EV-FlowNet [13], STE-FlowNet [5], Li *et al.* [22], ERAFT [11], Yang *et al.* [48], EVA-Flow [50], ADMFlow [33], E-FlowFormer [21], and TMA [25] have been proposed to improve performances. Some methods even use both events and images as input for flow estimation, such as Fusion-FlowNet [20], Pan *et al.* [36], DCEI-Flow [43] and RPEFlow [44]. In this work, we study a new problem of event-based meshflow estimation, proposing an efficient event-based meshflow network.

2.3. Event-based Optical Flow Dataset

Early works synthesize events by thresholding rgb images [18] and applying interpolation for high framerate [9]. However, the timestamp of synthesized events is inaccurate, let alone interpolation artifacts. The DAVIS event camera [2] can capture both images and real events, resulting in some event datasets: DVSFLOW [24], MVSEC [53] and DSEC [10], based on which EV-Flownet [52] and ER-

Table 1. Comparison of available datasets.'Dense OF' and Meshflow' indicate whether the dataset has dense optical flow labels and meshflow labels.

Dataset	Resolution	Dynamic Objects	Extreme Conditions	Dense OF	Meshflow	Motion Pattern
DVSFLOW [24]	180×240	✘	✘	✘	✘	Rotation
MVSEC [53]	260×346	✘	✔	✘	✘	Drone
DSEC [10]	480×640	✘	✔	✘	✘	Car
MDR [33]	260×346	✘	✘	✔	✘	Car
BlinkFlow [21]	480×640	✔	✘	✔	✘	Random
Ekubric [44]	720×1280	✔	✘	✔	✘	Falling
HREM(Ours)	720×1280	✔	✔	✔	✔	Random

AFT [11] compute sparse optical flow. In this way, however, flows can only locate on sparse event regions [43]. Recently, Luo *et al.* [33] proposed to render the MDR dataset from graphics, but it only contains static scenes. Wan *et al.* [44] synthesized the Ekubric dataset based on the Kubirc toolbox [12], which only includes a single falling motion pattern. Li *et al.* [21] considered non-rigid motions simulating dancing but with lower resolution and without extreme condition scenarios. None of the aforementioned datasets involve the estimation of meshflow. In this work, we render a comprehensive dataset that can support both meshflow and optical flow estimation, with a higher resolution, rich dynamic scenes, as well as extreme conditions like relatively low light and motion blur, as shown in Table 1.

3. Algorithm

3.1. High-Resolution Event Meshflow Dataset

We create the high-resolution event meshflow dataset (HREM), consisting of 100 virtual scenes that accurately mimic real-world environments, both indoors and outdoors. In these scenes, we put dynamic objects to simulate intricate object motions. Camera is programmed to track these movements, ensuring a realistic portrayal of motion. The Blender rendering engine was utilized to create high frame rate videos and dense optical flow labels. For event data generation, we employed three advanced simulators: ESIM [37], V2E [14], and DVS-Voltmeter [24]. The simulator that provided the highest contrast in warped events images was chosen for our dataset. We also process the dense optical flow using motion propagation and median filters, enabling the generation of meshflow labels. Our comparisons with existing datasets is shown in Fig. 2, emphasize the superiority of HREM in high-resolution, dynamic scenes, complex motion patterns, and comprehensive labeling. Following [28], we generate meshflow from dense optical flow, as depicted in Fig. 3.

Motion Propagation. Given a dense optical flow $\boldsymbol{F}$, we place a uniform mesh of 16×16 regular cells on its image plane and then select the motion of the middle point p in each cell as the local motion v_p. Since the vertices of the mesh near point p should have a similar motion to v_p, we define a rectangle that covers 3×3 cells centered at p, and assign to all the vertices within the rectangle, ensuring the propagation of the local motion v_p across the image plane.

Median Filters. The local motions of the middle points in all cells are propagated to their nearby mesh vertices, resulting in each vertex potentially receiving multiple motion vectors. To select the most appropriate motion vector for a given vertex, we apply a median filter f_1 to filter the candidate motions. The response of the filter is then assigned to the corresponding vertex. The median filter is a widely used technique in optical flow estimation and has been shown to produce high-quality flow estimates [39]. Therefore, we use the median filter for sparse motion regularization. However, due to dynamic objects, the motion field may contain noise and needs to be spatially smoothed. To address this issue, we apply another median filter f_2 that covers 3×3 cells neighborhood to suppress the noise in the motion field. This second median filter produces a spatially-smooth sparse motion field, which is what we called as meshflow. Ultimately, we use the meshflow (generally 16×16) as the label, which contains global motion information. However, we upsample the meshflow to the full image resolution for intuitive display and alignment applications.

3.2. Estimation for Meshflow and Optical Flow

Following [11] which estimates optical flow from two consecutive event sequences, we estimate the meshflow $MF_{k\to k+1}$ and optical flow $F_{k\to k+1}$ from event sequences $E(t_{k-1}, t_k)$ and $E(t_k, t_{k+1})$, and overall architecture of our network is shown in Fig. 4. Fellow [54], we convert the inputs $E(t_{k-1}, t_k)$ and $E(t_k, t_{k+1})$ to the 3D volumns $V_{k-1\to k}$ and $V_{k\to k+1}$.

3.2.1 Efficient Event-based Meshflow Network

We propose EEMFlow to directly output results of the same resolution as the ground truth meshflow MF_{GT} for supervised regression, fully leveraging the advantage about low-parameter and high-motion-information of meshflow.

Overall Structure of Meshflow Estimation. Since meshflow focuses more on global large motion rather than local detailed motion, the EEMFlow we designed does not require excessively deep network layers or refinement operations from coarse to fine. Firstly, EEMFlow employs a three-level pyramid feature encoder to extracting features $(V^i_{t_{k-1}})^N$ and $(V^i_{t_k})^N$ from $V_{k-1\to k}$ and $V_{k\to k+1}$, the convolutional layers within the i-th level share weights for

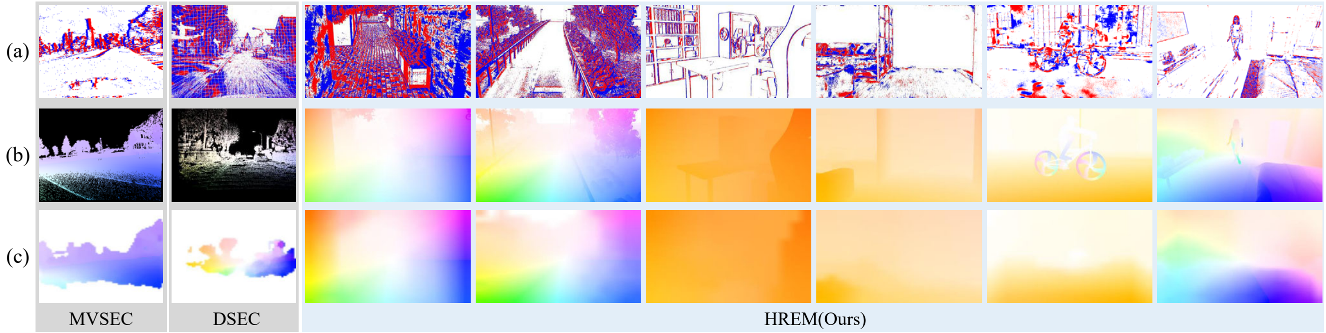

Figure 2. Examples from MVSEC [53], DSEC [10] and our dataset. (a): Event data, (b): Optical flow, (c): Upsampled meshflow. The meshflows of MVSEC and DSEC contain a significant number of ineffective areas due to the sparsity of optical flow. In contrast, our meshflow provides a complete global motion field. Best viewed on a color screen in high resolution.

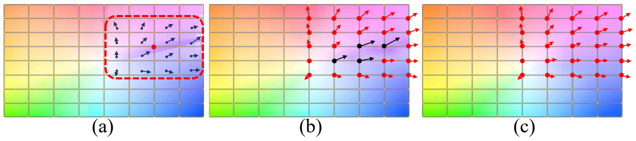

Figure 3. The process of generating meshflow from dense optical flow. (a) Propagate the motion vector of each grid center to the grid vertices. (b) Apply median filter f_1 to multiple motion vectors of each vertex to select the most appropriate motion for that vertex. (c) Use median filtering f_2 to smooth the motion field in the mesh grid. For ease of visualization, we present the 8×8 grid mesh in this paper.

$V^i_{t_{k-1}}$ and $V^i_{t_k}$. Secondly, EEMFlow utilizes multi-scale feature correlation to builds the cost volumes for meshflow estimation. Features $(V^i_{t_{k-1}})^N$ and $(V^i_{t_k})^N$ undergo average pooling operation $\mathcal{P}$ to the same resolution (1/64 resolution of $V_{k\rightarrow k+1}$), and then use correlation to capture relative motion information and output cost volumes $(C^i)^N$. Specifically, we employ the dilated feature correlation (DFC) to increase the search area while reducing computational parameters. Finally, we stack the cost volumes $(C^i)^N$ and features $(V^i_{t_{k-1}})^N$ after pooling, fuse them with a weighted-sum operation, and then feed them into the decoders to regress the meshflow $MF_{k\rightarrow k+1}$ at the same resolution as MF_{GT}. Inspiring by [51], we replace the conventional convolutions with group shuffle convolutions, which leads to efficient computation while maintaining high accuracy.

Dilated Feature Correlation. We use the inner product to calculate correlation between $V^i_{t_{k-1}}$ and $V^i_{t_k}$ for meshflow estimation :

$$C^i(\boldsymbol{u}, \boldsymbol{d}) = V^i_{t_k}(\boldsymbol{u}) \cdot V^i_{t_{k-1}}(\boldsymbol{u}+\boldsymbol{d})/M, \boldsymbol{d} \in \mathcal{N}, \tag{1}$$

where $\boldsymbol{u}$ represents the spatial coordinates on $V^i_{t_k}$, $\mathcal{N}$ represents the search grid of coordinate $\boldsymbol{u}$ in feature $V^i_{t_{k-1}}$, M represents the number of elements in $\mathcal{N}$, $\boldsymbol{d}$ represents the offset coordinates of the elements in $\mathcal{N}$, and $\cdot$ represents the inner product calculation. Many methods like [16, 40] simply define $\mathcal{N}$ as a square range of size $(2r+1) \times (2r+1)$ and observe that increasing the radius r can reduce errors but increase computational overhead. We propose dilated feature correlation that samples densely around the center and sparsely at farther distances, thereby reducing computation while enabling larger radius cost volumes, as Eq. 2:

$$\mathcal{N}(d_x, d_y) = \begin{cases} 0, \text{if } |d_x| + |d_y| = 2k, k \in [2, r], \\ 1, \text{others}, \end{cases} \tag{2}$$

where $\boldsymbol{d} = (d_x, d_y)$ represents the relative position coordinates within the neighborhood $\mathcal{N}$, where $\mathcal{N}(d_x, d_y) = 1$ indicates the computation of correlation, and $\mathcal{N}(d_x, d_y) = 0$ signifies no computation.

3.2.2 Event-based Optical Flow Network

Based on EEMFlow, we make some improvements for accurate estimation of optical flow $F_{k\rightarrow k+1}$, upgrading it to EEMFlow+. Since optical flow focuses more on local motion and pays attention to object edge details, we employ the coarse to fine residual approach to progressively refine the flow, which can be expressed as Eq. 3.

$$F^{i+1} = \text{Conv}^i(\mathcal{C}(V^i_{t_{k-1}}, \mathcal{W}(V^i_{t_k}, F^i_{\uparrow}))) + F^i_{\uparrow}, \tag{3}$$

where $\mathcal{C}(\cdot,\cdot)$, $\mathcal{W}(\cdot,\cdot)$ and $\uparrow$ donates our dilated feature correlation, the warping operation and upsampling, respectively. We employ the pyramid decoders for optical flow, thus F^i is the output flow of the i-th decoder, F^{i+1} and Conv^i are respectively the output flow and convolutions in the $i+1$-th decoder. The most noteworthy aspect is upsampling F^i to $F^i_{\uparrow}$. Many methods [6, 16, 40, 52] use bilinear interpolation for upsampling, but this can lead to the mixing of incorrect motions at object edges, resulting in blurring. Therefore, we propose the confidence-induced detail completion module for upsampling to enhance edge details.

Confidence-induced Detail Completion Module. We propose the confidence-induced detail completion module

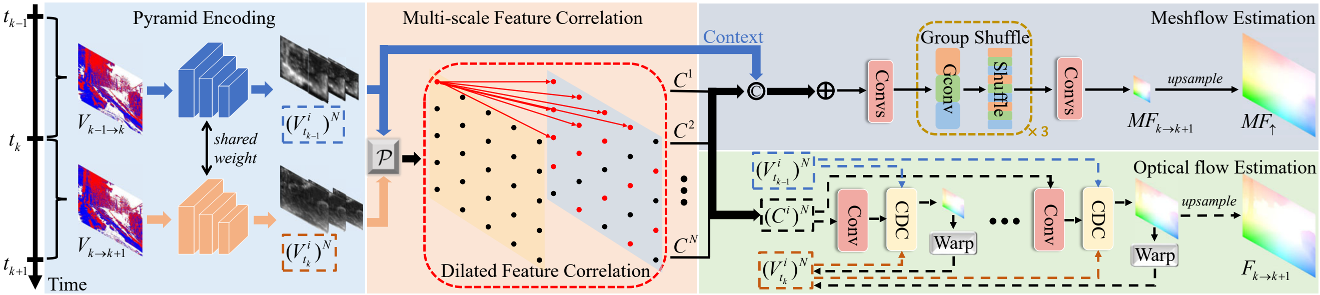

Figure 4. Our proposed network architecture. We employ pyramid encoders to extract multi-scale features from $V_{k-1\to k}$ and $V_{k\to k+1}$, then use dilated feature correlation to compute the cost volume between each layer of features, followed by decoding to output the results. For meshflow estimation, we utilize the decoders with group shuffle convolutions to output predictions $MF_{k\to k+1}$, upsampled to $MF\uparrow_{k\to k+1}$. For optical flow estimation, we refine flows using a coarse-to-fine residual approach and confidence-induced detail completion module, finally outputing the optical flow $F_{k\to k+1}$.

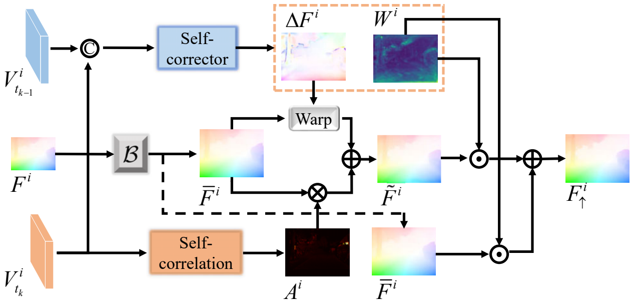

Figure 5. The structure of CDC. CDC employs self-corrector based on a dense convolutional network and self-correlation based on a self-attention mechanism to correct the flow obtained from bilinear upsampling. $\mathcal{B}$ represents bilinear upsampling.

(CDC) to eliminate the blurring of object edges caused by the mixing of multiple motions at the junctions of different movements during upsampling. The detailed structure of our CDC is shown in Fig. 5. Given a small-scale flow F^i from the i-th level, our CDC first generates an initial flow $\bar{F}^i$ for the $i+1$-th level through bilinear interpolation, and then arranges the self-corrector and self-correlation branches to correct it. Self-corrector is based on a dense convolutional network with a five-layer structure. It captures motion information within the edge neighborhood through dense convolution from the concatenated feature $V^i_{t_{k-1}}$ and $V^i_{t_k}$, outputting the corrected flow ΔF^i and the corrected confidence map W^i. Self-correlation is based on a self-attention mechanism, using a large receptive field to find the fine regions in features $V^i_{t_k}$ that are identical to the motion of the error region in $\bar{F}^i$. It outputs self-attention weights A^i, multiplied with the initial flow $\bar{F}^i$. With the corrected flow ΔF^i and self-attention weights A^i, we can generate the fine flow $\tilde{F}^i$:

$$\tilde{F}^i = \alpha\mathcal{W}(\bar{F}^i, \Delta F^i) + (1-\alpha)(A^i \otimes \bar{F}^i), \qquad (4)$$

where $\mathcal{W}(\cdot,\cdot)$ means the warping operation, $\otimes$ donates multiplication and $\alpha \in [0,1]$ is the weight coefficient. We identify error-prone object edge areas based on the corrected confidence map W^i, as weights to fuse the initial flow $\bar{F}^i$ and the fine flow $\tilde{F}^i$, obtaining the final corrected flow $F^i_\uparrow$:

$$F^i_\uparrow = W^i \odot \bar{F}^i + (1 - W^i) \odot \tilde{F}^i, \qquad (5)$$

where $\odot$ donates the element-wise multiplier.

3.2.3 Loss Function

For both meshflow and optical flow estimation, we use L1 loss for supervised regression during training. Our EEMFlow, used for estimating meshflow, directly outputs results at the same resolution as the meshflow GT, allowing for direct loss calculation. Similarly, the results outputted by our EEMFlow+ for optical flow estimation are at the same resolution as the network input, enabling direct calculation with the optical flow GT.

4. Experiments

4.1. Implementation Details

4.1.1 Datasets

Our HREM dataset includes 100 indoor and outdoor scenes, with a resolution of 1280×720. We randomly select 70 scenes for training and reserve the remaining 30 for testing. The training set comprises $20,000$ samples, while the test set contains $8,000$ samples. Additionally, we further divide the test set by scene type (outdoor vs. indoor) and camera motion speed during rendering, resulting in four sub-sequences (outdoor_slow, outdoor_fast, indoor_slow, indoor_fast), with mean motion magnitudes ranging from, $0-30$, $30-100$, $0-20$ and $20-100$ pixels, respectively. Moreover, similar to [52], we employ two data input modes: $dt = 1$ and $dt = 4$. $dt = 1$ uses the event sequence between two consecutive frames of RGB images as input, with a meshflow generation frequency of 60 Hz, while $dt = 4$ uses the event sequence between four consecutive frames of RGB images as input, with a meshflow

Table 2. Quantitative comparison of our EEMFlow with other advanced flow networks on our HREM dataset. The evaluation metric used is End-Point Error (EPE). ”Parameters” and ”Time” respectively indicate the network parameter count and inference time. ΔP and ΔT represent the change in parameter count and inference time relative to ERAFT [42] for other networks. Smaller values are desirable for all metrics. We highlight the best results in red and the second-best results in blue.

Method	Parameters	Time	Outdoor		Indoor		Avg
$dt = 1$	(M)	(ms)	Slow	Fast	Slow	Fast	
EVFlownet [52]	38.2	46	3.55	16.16	2.93	11.65	8.57
PWCNet [40]	3.36	42	3.91	14.49	2.86	11.89	8.29
ERAFT [11]	5.27	93	4.15	13.32	2.91	10.34	7.68
SKFlow [41]	6.28	145	3.76	11.78	7.24	8.81	7.24
GMA [17]	5.89	108	2.18	12.07	2.02	9.34	6.40
KPAFlow [32]	6.00	184	2.03	12.25	1.95	9.02	6.31
FlowFormer [15]	9.87	281	2.06	11.71	1.88	8.66	6.08
EEMFlow(Ours)	1.24	7	2.42	9.09	2.00	8.46	5.50
Method	ΔP	ΔT	Outdoor		Indoor		Avg
$dt = 4$	(M)	(ms)	Slow	Fast	Slow	Fast	
EVFlownet [52]	+624%	+51%	18.25	49.32	16.16	47.19	32.73
PWCNet [40]	-36%	-55%	16.40	46.17	14.49	40.90	29.49
ERAFT [11]	0%	0%	15.21	40.83	13.32	39.61	27.24
SKFlow [41]	+19%	+56%	14.93	39.24	11.71	39.22	26.28
GMA [17]	+11%	+16%	14.13	38.89	12.07	37.68	25.69
KPAFlow [32]	+14%	+99%	14.04	38.03	12.25	37.20	25.38
FlowFormer [15]	+88%	+202%	13.89	38.55	10.77	38.53	25.44
EEMFlow(Ours)	-76%	-92%	13.97	37.33	12.09	34.39	24.45

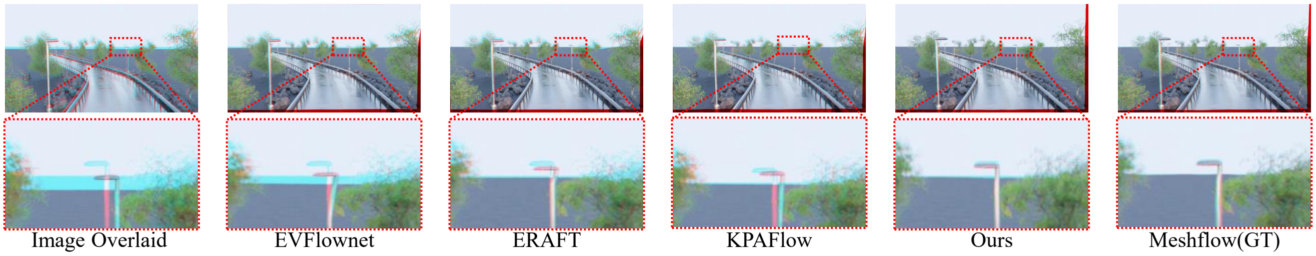

Figure 6. Subjective results of image registration using meshflow estimated by other methods and ours. We estimate meshflow from the event sequences and then warp image I_{t_1} onto image I_{t_2} for fusion, showcasing the fused result. The fewer blue or red ghosting artifacts indicate better alignment performance.

generation frequency of 15 Hz. DSEC [10] is a commonly used dataset for event-based optical flow estimation, which consists of real-world data captured by real event cameras mounted on cars. We also conduct experiments on DSEC to compare the performance for optical flow estimation.

4.1.2 Training details

We conduct experiments using the PyTorch framework on two NVIDIA 2080Ti GPUs. We train all networks with the same parameters on our HRDM dataset. We employ the AdamW optimizer and OneCycle policy with a learning rate of 5×10^{-4}, weight decay of 5×10^{-5}, and other default parameters set to $\beta_1 = 0.9, \beta_2 = 0.99, \epsilon = 1 \times 10^{-4}$. We train all networks up to $100k$ iterations to reach convergence.

4.1.3 Evaluation metrics

Following EV-FlowNet [52], we use the average End-point Error (EPE) as the metric. When evaluating meshflow, we upsample both the prediction and the ground truth to the input resolution to calculate metrics. In addition, the DSEC dataset uses NPE to measure the percentage of flow errors higher than N pixels in magnitude (e.g., 3PE, 2PE, 1PE) for flow outliers analysis, and employs the Angular Error (AE) to assess the directional accuracy.

4.2. Comparison with State-of-the-Arts

4.2.1 Results for Event-based Meshflow Estimation

In Table 2, We train and test our EEMFlow on our HREM dataset, compared with some advanced networks, e.g., EVFlownet [54], ERAFT [11], PWCNet [40], SKFlow [41], GMA [17], KPAFlow [32], and FlowFormer [15]. Since these networks are all structured to output prediction with the same resolution of input, we upsample the groud truth MF_{GT} to supervise these networks during training. However, EEMFlow can directly output prediction aligned MF_{GT} in resolution. But for fair comparison, we upsample the predictions of all networks to a uniform resolution during evaluation. We train and test all networks using two input modes ($dt = 1$ and $dt = 4$), and

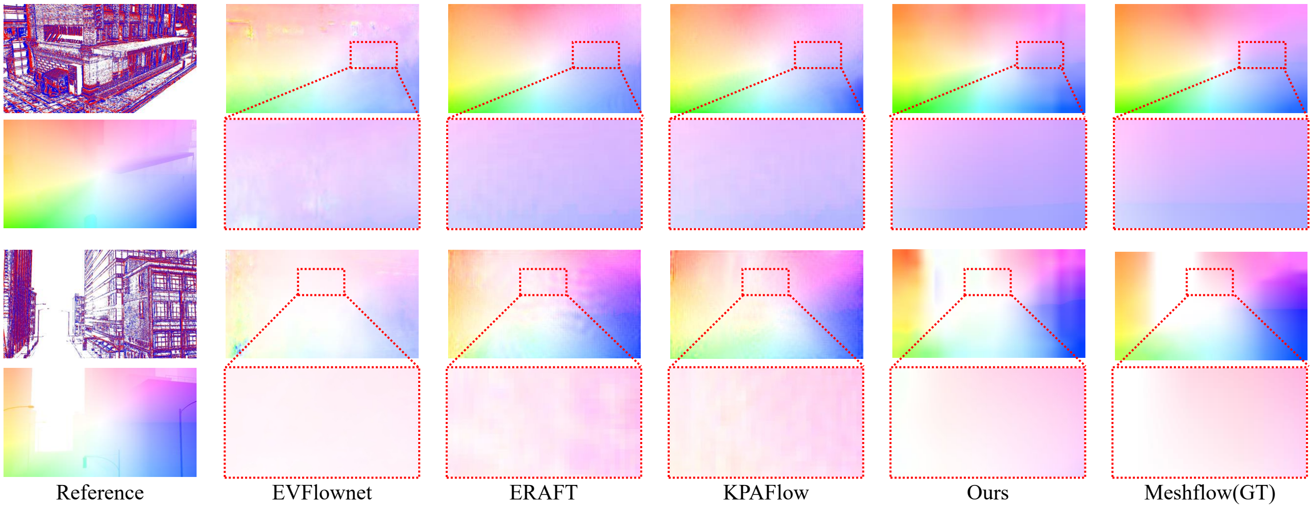

Figure 7. Qualitative comparison of our proposed EEMFlow with other advanced flow networks on our HREM dataset. The subjective images of events and dense optical flow on the left side serve as references. The areas enclosed by red rectangles are zoomed in.

Table 3. Results on DSEC dataset. All networks are trained and tested on the DSEC dataset for optical flow esimation.

Methods	FPS↑	1PE↓	2PE↓	3PE↓	EPE↓	AE↓
MutilCM [39]	-	76.6	48.5	30.9	3.47	14.0
EV-Flownet [52]	22.3	55.4	29.8	18.6	2.32	8.12
OF-EV-SNN [4]	-	53.7	20.2	10.3	1.71	6.34
EVA-Flow [50]	-	15.9	-	3.20	0.88	3.31
ERAFT [11]	11.4	12.7	4.74	2.68	0.79	2.85
ADMFlow [33]	9.88	12.5	4.67	2.65	0.78	2.84
EFlowformer [21]	-	11.2	4.10	2.45	0.76	2.68
TMA [25]	7.55	10.9	3.97	2.30	0.74	2.68
EEMFlow+(Ours)	39.2	11.4	3.93	2.15	0.75	2.67

present their performance metrics in four test sub-sequences (outdoor_slow, outdoor_fast, indoor_slow, indoor_fast). In addition, we average the metric scores across all four test sub-sequences as shown in the "Avg" column.

Table 2 shows that our EEMFlow achieves the lowest EPE score for both input modes $dt = 1$ and $dt = 4$ in the "outdoor_fast" and "indoor_fast" test sub-sequences, demonstrating its superior performance on high-speed and large-movement sequences. EEMFlow also exhibits great potential to outperform other flow networks on the "outdoor_slow" and "indoor_slow" test sub-sequences. Notably, EEMFlow achieves the lowest EPE scores in the "Avg" column for both input modes $dt = 1$ and $dt = 4$. Besides, we also achieves the least number of parameters and the fastest inference speed. Compared to ERAFT [11], our EEMFlow reduces the parameter count by 76% (from 5.27M to 1.24M), reduces the inference time by 92% (from 93ms to 7ms), and improves average EPE in $dt = 4$ by 8% (from 27.24 to 25.18). EEMFlow achieves comparable EPE performance to FlowFormer [15] but exhibits a 38.7× increase in inference speed.

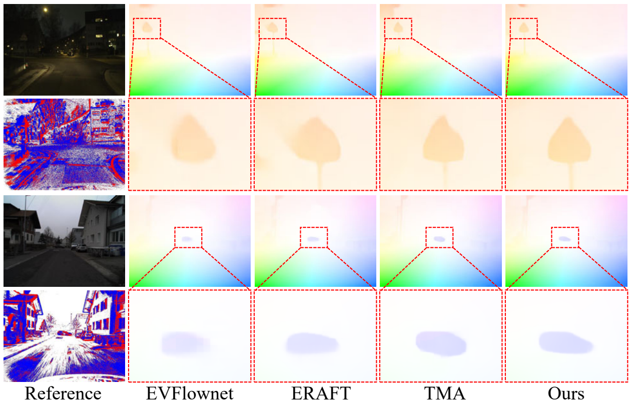

Figure 8. Qualitative comparisons on the DSEC test set. We visualize the dense predictions and zoom in the areas where are apparent differences.

In Fig. 7, we qualitatively compare our proposed EEMFlow with other flow networks, e.g., EVFlownet [54], ERAFT [11], and KPAFlow [32]. To facilitate comparison, we upsample the meshflow estimation and ground truth to the same resolution of input and zoom in on the areas with the apparent differences. EVFlownet shows the worst performance, with many holes and color mixing. ERAFT and KPAFlow would exhibit block artifacts and appear coarse in nature. In contrast, our EEMFlow results are smoother with more natural color transitions and greater similarity to the upsampled ground truth. In Fig. 6, we present the subjective results of these networks for image registration. We estimate and upsample meshflow prediction from the event sequences $E(t_1, t_2)$ and then warp image I_{t_1} onto image I_{t_2}. We also zoom in on the challenging areas of alignment in the registered results, clearly demonstrating that our EEMFlow achieves excellent image registration performance.

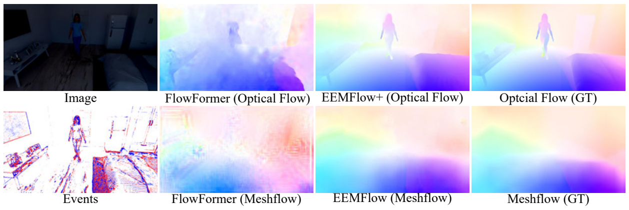

Figure 9. Visualization for meshflow and optical flow results.

Table 4. Results of meshflow and optical flow on HREM for $dt = 1$.

Task	Method	Outdoor		Indoor		Avg
		Slow	Fast	Slow	Fast	
Optical Flow	FlowFormer	6.20	16.06	5.99	15.27	10.88
	EEMFlow+	3.88	11.02	4.03	10.92	7.46
Mesh-flow	FlowFormer	5.99	15.12	5.74	14.95	10.45
	EEMFlow	2.42	9.09	2.00	8.46	5.50

4.2.2 Results for event-based optical flow estimation

Table 3 presents the comparative results on DSEC datset of our optical flow estimation network, EEMFlow+, against other event-based optical flow estimation networks. EEMFlow+ achieves state-of-the-art performance in the 2PE, 3PE, and AE metrics, and is on par with the best in the EPE metric. Notably, EEMFlow+ still maintains a significant advantage in inference speed. Compared to TMA [25], our EEMFlow+ increases the inference speed by 419% (from 7.55FPS to 39.2FPS). Fig. 8 displays the qualitative results of our optical flow estimation on the DSEC test set, compared with other advanced methods, including EVFlownet [52], ERAFT [42], and TMA [25]. By zooming into the object areas, it is clearly observable that our results exhibit more regular shapes and clearer boundaries, highlighting the detail enhancement capability of our proposed CDC module.

4.3. Ablation Studies

4.3.1 The Advantages of Event-Meshflow Estimation

According to the ERAFT [42], image-based approaches face challenges in handling difficult images due to the limited dynamic range of image sensors. As shown in Fig. 9, even advanced image-based techniques like FlowFormer struggle, while our event-based methods (EEMFlow, EEMFlow+) exhibit notably superior performance. Additional quantitative comparisons are provided in Table 4, our network of event-based meshflow estimation EEMFlow achieves the lowest EPE scores in HREM datset for the $dt = 1$ input mode, which demonstrates the remarkable efficiency of our EEMFlow makes it suitable for real-time applications such as online video stabilization and autonomous driving.

Table 5. Ablation studies about CDC of EEMFlow+. CDC consists of two branches, the self-corrector and self-correlation.

Model	Self-corrector	Self-correlation	FPS↑	3PE↓	EPE↓	AE↓
(a)	✘	✘	60.4	3.52	0.89	3.11
(b)	✔	✘	55.6	2.77	0.81	2.92
(c)	✘	✔	46.3	2.65	0.79	2.78
(d)	✔	✔	39.2	2.15	0.75	2.67

4.3.2 Experiments for CDC of EEMFlow+

In Table 5, we also conduct ablation experiments on the CDC module of EEMFlow+ used for optical flow estimation, including its two branches, the self-corrector and self-correlation. We train and evaluate all models using the same settings on the DSEC dataset to show the individual impact of each branch in CDC module. Comparison of (a)&(b) demonstrates that the CDC with only the self-corrector branch can bring a significant increase in accuracy with a minimal loss in speed. Comparison of (b)&(c) shows that self-correlation, compared to self-corrector, can lead to a higher increase in accuracy, albeit at a further reduction in inference speed. Finally, the comparison of (a)&(d) shows that the CDC composed of both the self-corrector and self-correlation branches significantly improves the accuracy of optical flow estimation with an acceptable loss in speed.

5. Conclusion

In this work, we develop the first event-based meshflow dataset, termed HREM, where 100 indoor and outdoor virtual scenes with rich scene contents are rendered using Blender. Our HREM possess physically correct accurate events and meshflow label pairs with so far the highest resolution. Furthermore, we propose an efficient event-based meshflow network (EEMFlow), which achieves state-of-the-art performance on HREM dataset while maintaining high efficiency. Based on EEMFlow, we propose a confidence-induced detail completion module to upgrade it as EEMFlow+ for optical flow estimation, achieving SOTA results on DSEC dataset with the fastest inference speed.

The integration of optical flow and depth data allows for the extension of the proposed framework to address 3D scene flow. Subsequently, our future work will involve expanding this methodology on the HREM dataset by integrating scene flow annotations and investigating scene flow estimation using event cameras.

Acknowledgements. This work was supported by National Natural Science Foundation of China (NSFC) under grant No.62372091, Sichuan Science and Technology Program of China under grants Nos. 2023NSFSC0462, 2022YFQ0079 and 2023YFG0054.

References

[1] Ryad Benosman, Sio-Hoi Ieng, Charles Clercq, Chiara Bartolozzi, and Mandyam Srinivasan. Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37, 2012. 2

[2] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240×180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 2

[3] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. A differentiable recurrent surface for asynchronous event-based data. In *Proc. ECCV*, pages 136–152. Springer, 2020. 2

[4] Javier Cuadrado, Ulysse Rançon, Benoit R Cottereau, Francisco Barranco, and Timothée Masquelier. Optical flow estimation from event-based cameras and spiking neural networks. *Frontiers in Neuroscience*, 17:1160034, 2023. 7

[5] Ziluo Ding, Rui Zhao, Jiyuan Zhang, Tianxiao Gao, Ruiqin Xiong, Zhaofei Yu, and Tiejun Huang. Spatio-temporal recurrent networks for event-based optical flow estimation. In *Proc. AAAI*, pages 525–533, 2022. 2

[6] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. ICCV*, pages 2758–2766, 2015. 4

[7] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020. 1

[8] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Proc. ICCV*, pages 5633–5643, 2019. 2

[9] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *Proc. CVPR*, pages 3586–3595, 2020. 2

[10] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3): 4947–4954, 2021. 2, 3, 4, 6

[11] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras. In *International Conference on 3D Vision (3DV)*, pages 197–206, 2021. 1, 2, 3, 6, 7

[12] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proc. CVPR*, pages 3749–3761, 2022. 3

[13] Jesse Hagenaars, Federico Paredes-Vallés, and Guido De Croon. Self-supervised learning of event-based optical flow with spiking neural networks. *Advances in Neural Information Processing Systems*, 34:7167–7179, 2021. 2

[14] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events. In *Proc. CVPR*, pages 1312–1321, 2021. 3

[15] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer: A transformer architecture for optical flow. *ECCV*, 2022. 2, 6, 7

[16] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *Proc. CVPR*, 2019. 4

[17] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proc. ICCV*, pages 9772–9781, 2021. 2, 6

[18] Jacques Kaiser, J Camilo Vasquez Tieck, Christian Hubschneider, Peter Wolf, Michael Weber, Michael Hoff, Alexander Friedrich, Konrad Wojtasik, Arne Roennau, Ralf Kohlhaas, et al. Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks. In *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 127–134, 2016. 2

[19] Chankyu Lee, Adarsh Kumar Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy. Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks. In *Proc. ECCV*, pages 366–382. Springer, 2020. 2

[20] Chankyu Lee, Adarsh Kumar Kosta, and Kaushik Roy. Fusion-flownet: Energy-efficient optical flow estimation using sensor fusion and deep fused spiking-analog network architectures. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6504–6510. IEEE, 2022. 2

[21] Yijin Li, Zhaoyang Huang, Shuo Chen, Xiaoyu Shi, Hongsheng Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Blinkflow: A dataset to push the limits of event-based optical flow estimation. *arXiv preprint arXiv:2303.07716*, 2023. 2, 3, 7

[22] Zhuoyan Li, Jiawei Shen, and Ruitao Liu. A lightweight network to learn optical flow from event data. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 1–7. IEEE, 2021. 2

[23] Kaimo Lin, Shuaicheng Liu, Loong-Fah Cheong, and Bing Zeng. Seamless video stitching from hand-held camera inputs. In *Computer Graphics Forum*, pages 479–487. Wiley Online Library, 2016. 2

[24] Songnan Lin, Ye Ma, Zhenhua Guo, and Bihan Wen. Dvs-voltmeter: Stochastic process-based event simulator for dynamic vision sensors. In *Proc. ECCV*, pages 578–593. Springer, 2022. 2, 3

[25] Haotian Liu, Guang Chen, Sanqing Qu, Yanping Zhang, Zhijun Li, Alois Knoll, and Changjun Jiang. Tma: Temporal motion aggregation for event-based optical flow. In *Proc. ICCV*, 2023. 2, 7, 8

[26] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM transactions on graphics (TOG)*, 32(4):1–10, 2013. 1

[27] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Steadyflow: Spatially smooth optical flow for video stabilization. In *Proc. CVPR*, pages 4209–4216, 2014. 1

[28] Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, and Bing Zeng. Meshflow: Minimum latency online video stabilization. In *Proc. ECCV*, pages 800–815. Springer, 2016. 1, 2, 3

[29] Shuaicheng Liu, Nianjin Ye, Chuan Wang, Jirong Zhang, Lanpeng Jia, Kunming Luo, Jue Wang, and Jian Sun. Content-aware unsupervised deep homography estimation and its extensions. 45(3):2849–2863, 2022. 1

[30] Shuaicheng Liu, Yuhang Lu, Hai Jiang, Nianjin Ye, Chuan Wang, and Bing Zeng. Unsupervised global and local homography estimation with motion basis learning. 45(6): 7885–7899, 2023. 1

[31] Zhen Liu, Yinglong Wang, Bing Zeng, and Shuaicheng Liu. Ghost-free high dynamic range imaging with context-aware transformer. In *Proc. ECCV*, pages 344–360. Springer, 2022. 1

[32] Ao Luo, Fan Yang, Xin Li, and Shuaicheng Liu. Learning optical flow with kernel patch attention. In *Proc. CVPR*, 2022. 2, 6, 7

[33] Xinglong Luo, Kunming Luo, Ao Luo, Zhengning Wang, Ping Tan, and Shuaicheng Liu. Learning optical flow from event camera with rendered dataset. In *Proc. ICCV*, 2023. 2, 3, 7

[34] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Depth-aware multi-grid deep homography estimation with contextual correlation. 32(7):4460–4472, 2022. 1

[35] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Parallax-tolerant unsupervised deep image stitching. In *Proc. ICCV*, pages 7399–7408, 2023. 2

[36] Liyuan Pan, Miaomiao Liu, and Richard Hartley. Single image optical flow estimation with an event camera. In *Proc. CVPR*, pages 1669–1678, 2020. 2

[37] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018. 3

[38] Zhihang Ren, Jiajia Li, Shuaicheng Liu, and Bing Zeng. Meshflow video denoising. In *Proc. ICIP*, pages 2966–2970, 2017. 2

[39] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *Proc. CVPR*, pages 2432–2439, 2010. 3, 7

[40] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proc. CVPR*, pages 8934–8943, 2018. 4, 6

[41] Shangkun Sun, Yuanqi Chen, Yu Zhu, Guodong Guo, and Ge Li. Skflow: Learning optical flow with super kernels. *Proc. NeurIPS*, 35:11313–11326, 2022. 2, 6

[42] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proc. ECCV*, 2020. 2, 6, 8

[43] Zhexiong Wan, Yuchao Dai, and Yuxin Mao. Learning dense and continuous optical flow from an event camera. *IEEE Transactions on Image Processing*, 31:7237–7251, 2022. 2, 3

[44] Zhexiong Wan, Yuxin Mao, Jing Zhang, and Yuchao Dai. Rpeflow: Multimodal fusion of rgb-pointcloud-event for joint optical flow and scene flow estimation. In *Proc. ICCV*, pages 10030–10040, 2023. 2, 3

[45] Yiming Wang, Qian Huang, Chuanxu Jiang, Jiwen Liu, Mingzhou Shang, and Zhuang Miao. Video stabilization: A comprehensive survey. *Neurocomputing*, 2022. 1, 2

[46] Qingsen Yan, Jinqiu Sun, Haisen Li, Yu Zhu, and Yanning Zhang. High dynamic range imaging by sparse representation. *Neurocomputing*, 269:160–169, 2017. 1, 2

[47] Min Yang, Jingkun Liang, Jianhai Zhang, Haidong Gao, Fanyong Meng, Li Xingdong, and Sung-Jin Song. Non-local means theory based perona–malik model for image denosing. *Neurocomputing*, 120:262–267, 2013. 2

[48] Yan Yang, Liyuan Pan, and Liu Liu. Event camera data pretraining. In *Proc. ICCV*, pages 10699–10709, 2023. 2

[49] Nianjin Ye, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Jue Wang, and Yongqing Cui. Deepmeshflow: Content adaptive mesh deformation for robust image registration. *arXiv preprint arXiv:1912.05131*, 2019. 1, 2

[50] Yaozu Ye, Hao Shi, Kailun Yang, Ze Wang, Xiaoting Yin, Yaonan Wang, and Kaiwei Wang. Towards anytime optical flow estimation with event cameras. *arXiv preprint arXiv:2307.05033*, 2023. 2, 7

[51] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proc. CVPR*, pages 6848–6856, 2018. 4

[52] Alex Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. In *Proceedings of Robotics: Science and Systems*, 2018. 2, 4, 5, 6, 7, 8

[53] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3 (3):2032–2039, 2018. 2, 3, 4

[54] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proc. CVPR*, pages 989–997, 2019. 3, 6, 7