Secrets of Edge-Informed Contrast Maximization for Event-Based Vision – Supplementary Material –

Pritam P. Karmokar*, Quan H. Nguyen*, and William J. Beksi
The University of Texas at Arlington
Arlington, TX, USA
{pritam.karmokar,quan.nguyen4}@mavs.uta.edu, william.beksi@uta.edu

A. Full Results on the DSEC Test Sequences

We provide a full report of our accuracy evaluation results on the DSEC benchmark in Tab. 1. In addition, a complete overview of the sharpness results in terms of flow warp loss (FWL) scores on the DSEC test set are shown in Tab. 2. At the time of this writing, Liu *et al*. [5] had the best-known supervised learning (SL) method on the DSEC-Flow benchmark in terms of accuracy. However, Liu *et al*. [5] did not report their FWL scores. Conversely, Gehrig *et al*. [3] had the best-known SL method in terms of FWL scores.

Tab. 1 provides a summary of the accuracy comparisons against these SL techniques as well as the best-known model-based (MB) methods. Similarly, Tab. 2 summarizes the comparisons of the FWL scores (sharpness). We note that no MB method, including ours, produces accuracy scores comparable to state-of-the-art SL approaches on the DSEC test set. Nonetheless, when compared to other state-of-the-art MB methods our approach provides comparable average endpoint error (AEE) and percentage 3-pixel error (%3PE). Additionally, our percentage 1-pixel error (%1PE) scores are consistently better than other MB methods. Interestingly, for `zurich_city_12_a` (noisy), [1] performed better than others due to its event denoising component.

		All			interlaken_00_b			interlaken_01_a			thun_01_a		
		AEE ↓	%1PE ↓	%3PE ↓	AEE ↓	%1PE ↓	%3PE ↓	AEE ↓	%1PE ↓	%3PE ↓	AEE ↓	%1PE ↓	%3PE ↓
SL	TMA [5]	**0.74$_3$**	**10.86$_3$**	**2.30$_1$**	**1.38$_5$**	**18.12**	**5.78$_5$**	**0.80$_9$**	**12.89$_4$**	**3.10$_8$**	**0.61$_6$**	**8.84$_4$**	**1.60$_5$**
	E-RAFT [3]	0.78$_8$	12.74$_2$	2.68$_4$	1.39$_4$	20.41$_5$	6.18$_9$	0.89$_9$	15.48$_3$	3.90$_7$	0.65$_4$	10.95$_4$	1.87
MB	Brebion *et al.* [1]	4.88$_1$	82.81$_2$	41.95$_2$	8.58$_8$	90.12	59.84$_1$	5.94	86.63	47.33	3.01	71.66$_3$	29.69$_7$
	Shiba *et al.* [7]	$\underline{3.47_2}$	76.57	$\underline{30.85_5}$	$\underline{5.74}$	78.08$_6$	$\underline{38.92_5}$	$\underline{3.74}$	75.40$_2$	$\underline{31.36_6}$	2.12	64.73	17.68$_4$
	Ours (EINCM)	5.00$_3$	$\underline{68.66_8}$	35.87$_2$	6.39$_6$	$\underline{72.63}$	43.6	5.48$_2$	$\underline{70.00_8}$	41.32$_8$	$\underline{2.01_5}$	$\underline{51.83_2}$	$\underline{16.17_4}$

		thun_01_b			zurich_city_12_a			zurich_city_14_c			zurich_city_15_a		
		AEE ↓	%1PE ↓	%3PE ↓	AEE ↓	%1PE ↓	%3PE ↓	AEE ↓	%1PE ↓	%3PE ↓	AEE ↓	%1PE ↓	%3PE ↓
SL	TMA [5]	**0.55$_2$**	**7.44$_9$**	**1.31**	**0.57$_2$**	**9.6**	**8.66**	**0.65$_7$**	**14.10$_7$**	**1.99**	**0.55$_4$**	**6.95$_4$**	**1.07$_9$**
	E-RAFT [3]	0.57$_7$	8.322	1.52	0.61$_2$	11.24	1.05$_7$	0.71$_3$	15.5	1.91$_3$	0.58$_9$	8.74$_8$	1.30$_3$
MB	Brebion *et al.* [1]	3.91$_3$	77.56$_7$	34.69	$\underline{3.13_9}$	80.27$_7$	34.07$_8$	3.99$_8$	88.30$_4$	45.67	3.78$_1$	81.35$_3$	37.98$_7$
	Shiba *et al.* [7]	$\underline{2.48}$	73.63$_2$	$\underline{23.56_4}$	3.86	86.39$_8$	$\underline{43.96_1}$	$\underline{2.72}$	76.85$_1$	$\underline{30.53}$	$\underline{2.35}$	72.86$_4$	$\underline{20.98_7}$
	Ours (EINCM)	2.77$_8$	$\underline{63.63_3}$	26.56	8.37	$\underline{79.59_7}$	45.78$_6$	3.15$_3$	$\underline{64.68_7}$	30.87$_9$	3.00$_5$	$\underline{62.19_9}$	26.63$_3$

Table 1. DSEC test set accuracy results. *Bold* and *underline* typefaces indicate the best among supervised learning and model-based methods, respectively.

* Indicates equal contribution.

		All	int_00_b	int_01_a	thu_01_a	thu_01_b	zur_12_a	zur_14_c	zur_15_a
		FWL ↑	FWL ↑	FWL ↑	FWL ↑	FWL ↑	FWL ↑	FWL ↑	FWL ↑
SL	E-RAFT [3]	1.29	1.32	1.42	1.20	1.18	1.12	1.47	1.34
MB	Shiba *et al.* [7]	1.36	1.50	1.51	1.24	1.24	1.14	1.50	1.41
	Ours (EINCM)	**1.61$_5$**	**1.94**	**1.86$_8$**	**1.40**	**1.39$_6$**	**1.28$_9$**	**1.60$_5$**	**1.60$_3$**

Table 2. DSEC test set sharpness results (FWL scores). *Bold* typeface is used to indicate the **best**.

B. Additional Sharpness Results on MVSEC

For the $dt = 1$ setting on MVSEC, each data sample contains very few events ($\approx$ 6.5 K, 9.4 K, 7.8 K, and 8.7 K on average in `indoor_flying1`, `indoor_flying2`, `indoor_flying3`, and `outdoor_day1`, respectively). In this scenario, MultiCM [7] reported (sharpness) FWL scores of $\approx$1 for each sequence. We report further comparisons for the MVSEC $dt = 1$ case with exact FWL scores in Tab. 3. The FWL scores of MultiCM were obtained using the open-source code provided by the authors. We observe that although small, the FWL scores for both indoor and outdoor sequences were all > 1 and better than MultiCM. We also note that the average FWL score for `indoor_flying2` is higher than other sequences, which can be correlated with it comprising a larger average number of events.

	MVSEC (*dt*=1)			
	indoor_flying1	indoor_flying2	indoor_flying3	outdoor_day1
Ground truth	1.02$_6$	0.98$_6$	1.00$_6$	0.99$_6$
Shiba *et al.* [7]	1.01$_9$	0.96$_8$	0.98$_9$	0.98$_5$
Ours (EINCM)	**1.03$_4$**	**1.16$_1$**	**1.03$_8$**	**1.00$_3$**

Table 3. Flow warp loss (FWL) for MVSEC sequences with $dt = 1$ on grayscale frames. *Bold* typeface indicates the **best**.

C. MVSEC Outdoor Evaluations

The MVSEC outdoor sequence `outdoor_day1` consists of 11,440 image frames. Yet, optical flow is only evaluated on a small subset of this sequence. To compare their results with UnFlow [6], Zhu *et al*. [8] evaluated on 800 frames from `outdoor_day1` spanning a time window from 222.4 s to 240.4 s. These start and end times, interpreted as image timestamps, correspond to $1,506,118,124.7330644$ s and $1,506,118,142.7177844$ s, respectively. Equivalently, interpreted as image indices, they correspond to the 10,138$^{\text{th}}$ and the 10,958$^{\text{th}}$ (with starting index 0), respectively. Following Zhu *et al*. [8], other works that benchmarked their evaluations on `outdoor_day1` fall short on consistently reporting and/or using the same evaluation points. To the authors' knowledge, there are at least two sets of evaluation points for the MVSEC `outdoor_day1` sequence in the literature.

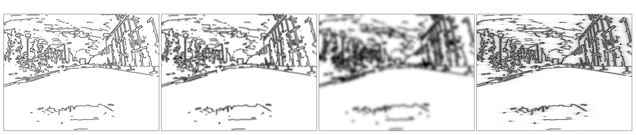
C.1. Discrepancies

We summarize discrepancies in prior works as follows.

- Although Zhu *et al*. [8] reported a usage of 800 frames, the provided timestamps indicate 820 frames instead. On the other hand, their publicly available code and assets suggest the use of exactly 800 frames.
- Lee *et al*. [4] and Ding *et al*. [2] used two sets of 401 frames, one between the image indices $[9200, 9600]$ and the other between $[10500, 10900]$.
- Shiba *et al*. [7] mentioned using the same 800 frames as [8]. However, the reported results were not reasonably reproducible on our local machine. Therefore, in Tab. 1 of the main paper, the accuracy scores for [7] were obtained by running their code locally on the 800 frames as suggested by [8]. This corresponds to image indices 10,148 to 10,948.

Our evaluations on MVSEC `outdoor_day1` were performed on the 800 frames corresponding to the image indices $[10148, 10948]$ (starting at 0).

D. Edge Smoothing Sensitivity Analysis



(a) Original (b) $k = 1$ (c) $k = 5$ (d) IEDT

Figure 1. Edge smoothing operations.

In Tab. 4, we present a sensitivity analysis on the choice of edge smoothing methods. Observe that we obtained the best performance by using a Gaussian kernel size of $k = 1$ (Fig. 1). Increasing the kernel size to $k = 5$ resulted in enlarging the reach of an edgel (edge pixel) to non-edge pixel regions. Yet, it also simultaneously increased the softness of the edgels, which resulted in performance degradation. The inverse exponential distance transform (IEDT) [1] can smooth edges in a manner where the reach of edgels can be extended to the non-edge pixel regions without softening the edgel itself. Edges smoothed using the IEDT yielded better performance when compared to Gaussian blurring with $k = 5$. Note that the IWEs for all three settings were consistently obtained using $k = 1$. Nevertheless, the IEDT is computationally expensive (Tab. 5). Consequently, we used a Gaussian blur with $k = 1$ for edge smoothing.

	`outdoor_day1` ($dt = 4$)		
	$k = 1$	$k = 5$	IEDT
AEE ↓	1.70_4	1.76_7	1.73_6
%3PE ↓	16.01_3	16.93	16.71_9
FWL ↑	1.23	1.20_6	1.21_1

Table 4. Edge smoothing sensitivity analysis results. We report the accuracy and sharpness scores on the MVSEC sequence `outdoor_day1` ($dt = 4$). The first two columns depict a Gaussian blur with kernel size $k = 1$ and $k = 5$. The third column shows results using the inverse exponential distance transform (IEDT).

E. Hyperparameters

As discussed in the main paper, all the experiments used five pyramid levels to take advantage of multiscaling. With regards to multiple references for MVSEC $dt = 1$, reference times t_0, t_{mid}, and t_1 were used to compute contrasts, while the image timestamps $\mathcal{T}^{(i)}$ were utilized to compute correlations. In the MVSEC $dt = 4$ case, there were three images within the duration of each data sample. Therefore, the image timestamps $\mathcal{T}^{(i)}$ were used as reference times to compute both contrasts and correlations. For the ECD sequence `slider_depth`, $dt = 2$ was chosen (with on average ≈ 24 K events per data sample) for the evaluations. Each data sample consisted of three images: two at the boundaries and one in between. Contrasts and correlations were calculated at the three image timestamps $\mathcal{T}^{(i)}$. Similarly, in the DSEC sequences each data sample consisted of three images and the timestamps $\mathcal{T}^{(i)}$ served as reference times for computing both contrasts and correlations.

The accuracy and FWL scores were evaluated for each sequence using the corresponding events within a data sample. However, for optimization we ensured a fixed number of events per data sample $\mathcal{D}^{(i)}$. Specifically, we used 30 K and 40 K events for the indoor and outdoor sequences from MVSEC, respectively. For DSEC and ECD, we used 1.5 M and 30 K events, respectively. For the MVSEC sequences, we set $\alpha = 20, \beta = 35$, for ECD we used $\alpha = 60, \beta = 60$, and for DSEC $\alpha = 2000, \beta = 4000$ were used.

Extracting image edges via OpenCV's `Canny`[1] involves using a pair of threshold values (`thresh_1`, `thresh_2`).

[1] https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html

We used $(100, 200)$ and $(30, 80)$ for the MVSEC indoor and outdoor sequences, respectively. For ECD, $(100, 200)$ was used. Finally, for DSEC $(30, 80)$ was used for all sequences except for `zurich_city_12_a` (night-time images with extremely noisy events), where the thresholds $(10, 60)$ were used. The coefficient γ for the regularizer term in our objective function was fixed to 0.0025 for the MVSEC sequences, while it was set to 0.0 for both the ECD and DSEC sequences.

F. EINCM Multiscale Pseudocode

In Alg. 1, we present the high-level pseudocode of the multiscaling scheme used by our method. The i-th input data sample $\mathcal{D}^{(i)}$ consists of the corresponding events $\mathcal{E}^{(i)}$, edge images $\mathcal{I}^{(i)}$, and image timestamps $\mathcal{T}^{(i)}$. The outer loop (lines 5-19) reflects the fact that we used five scales in the multiscale scheme. The number of scales as well as the resolution of the motion parameters at each scale are preset and can be adjusted. The main contrast and correlation maximization (CCM, line 6), where we optimize for the motion parameters, requires a loss function and an initial ${}_{l}\mathbf{\Theta}_i$ (*i.e.*, the first argument). To solve for *handovers* (line 12), we essentially solve for the coefficient w_{ho}. This coefficient linearly combines the optimized parameters at the current index and scale (result of line 6), and the downsampled optimized parameters from the previous index at the current scale (result of line 7). We optimize for w_{ho} in the same manner as the main CCM optimization where we replace ${}_{l}\mathbf{\Theta}_i$ by the aforementioned weighted sum.

Algorithm 1: EINCM Multiscale Pseudocode

Data: $\mathcal{E}^{(i)}, \mathcal{I}^{(i)}, \mathcal{T}^{(i)}$, and optionally ${}_{0}\mathbf{\Theta}^*_{i-1}$
Hyperparameters: $\boldsymbol{a}$
Result: ${}_{0}\mathbf{\Theta}^*_i$

1 **if** ${}_{0}\mathbf{\Theta}^*_{i-1}$ is available **then**
2 $\quad {}_{4}\mathbf{\Theta}^0_i \leftarrow \texttt{downscale}({}_{0}\mathbf{\Theta}^*_{i-1})$
3 **else**
4 $\quad {}_{4}\mathbf{\Theta}^0_i \leftarrow$ **zero**
5 **for** $\texttt{lvl} = 4$ **to** 0 **do**
6 $\quad {}_{\texttt{lvl}}\mathbf{\Theta}^*_i \leftarrow \arg\max_{{}_{\texttt{lvl}}\mathbf{\Theta}_i} \texttt{loss}({}_{\texttt{lvl}}\mathbf{\Theta}^0_i; \mathcal{E}^{(i)}, \mathcal{I}^{(i)}, \mathcal{T}^{(i)})$
7 $\quad {}_{\texttt{lvl}}\mathbf{\Theta}^{\downarrow}_{i-1} \leftarrow \texttt{downscale}({}_{0}\mathbf{\Theta}^*_{i-1})$
8 $\quad w_{\text{ho}} \leftarrow 0$
9 $\quad$ **if** $\texttt{handover\_flag}_{\texttt{lvl}}$ **then**
10 $\quad\quad$ **if** $\texttt{solve\_flag}_{\texttt{lvl}}$ **then**
11 $\quad\quad\quad w^0_{\text{ho}} \leftarrow 0.5$
12 $\quad\quad\quad w^*_{\text{ho}} \leftarrow \arg\max_{w_{\text{ho}}} \texttt{loss}_{w_{\text{ho}}}(w^0_{\text{ho}}; {}_{\texttt{lvl}}\mathbf{\Theta}^*_i, {}_{\texttt{lvl}}\mathbf{\Theta}^{\downarrow}_{i-1}, \mathcal{E}^{(i)}, \mathcal{I}^{(i)}, \mathcal{T}^{(i)})$
13 $\quad\quad\quad w_{\text{ho}} \leftarrow w^*_{\text{ho}}$
14 $\quad\quad$ **else**
15 $\quad\quad\quad w_{\text{ho}} \leftarrow \boldsymbol{a}$
16 $\quad {}_{\texttt{lvl}}\mathbf{\Theta}^*_i \leftarrow w_{\text{ho}} \cdot {}_{\texttt{lvl}}\mathbf{\Theta}^{\downarrow}_{i-1} + (1 - w_{\text{ho}}) \cdot {}_{\texttt{lvl}}\mathbf{\Theta}^*_i$
17 $\quad$ **if** $\texttt{lvl} \neq 0$ **then**
18 $\quad\quad {}_{\texttt{lvl}-1}\mathbf{\Theta}^0_i \leftarrow \texttt{upscale}({}_{\texttt{lvl}}\mathbf{\Theta}^*_i)$
19 **end for**
20 **return** ${}_{0}\mathbf{\Theta}^*_i$

G. Runtime Analysis

In Tab. 5, we present a detailed runtime report of our image preprocessing as well as the optimization (including and excluding the first `jit`[2] compilation) pipeline on the same machine and software suite described in the main paper.

	ECD (176×240)	MVSEC (260×346)	DSEC (480×640)
Preprocessing	$17.4\,\text{ms} \pm 588\,\mu\text{s}$	$33.7\,\text{ms} \pm 1.81\,\mu\text{s}$	$68.7\,\text{ms} \pm 3.59\,\mu\text{s}$
Edge extraction	$146\,\mu\text{s} \pm 27.7\,\mu\text{s}$	$162\,\mu\text{s} \pm 18.62\,\mu\text{s}$	$351\,\mu\text{s} \pm 46.1\,\mu\text{s}$
Gaussian blur	$195\,\mu\text{s} \pm 13.9\,\mu\text{s}$	$395\,\mu\text{s} \pm 29.6\,\mu\text{s}$	$1.6\,\text{ms} \pm 172\,\mu\text{s}$
Inverse exponential distance transform	$755\,\text{ms} \pm 28.8\,\text{ms}$	$1.56\,\text{s} \pm 21.6\,\text{ms}$	$5.36\,\text{s} \pm 86.3\,\text{ms}$
CCM at pyramid level 0 (include first `jit` compilation)	$356\,\text{ms} \pm 1.02\,\text{s}$	$465.59\,\text{ms} \pm 1.354\,\text{s}$	$2.35\,\text{s} \pm 3.65\,\text{s}$
CCM at pyramid level 0 (exclude first `jit` compilation)	$15.96\,\text{ms} \pm 846.4\,\mu\text{s}$	$32.44\,\text{ms} \pm 188.3\,\mu\text{s}$	$1.128\,\text{s} \pm 280.3\,\mu\text{s}$
Downscale from pyramid level 4 to 0	$96.4\,\text{ms} \pm 62.9\,\text{ms}$	$96.4\,\text{ms} \pm 62.9\,\text{ms}$	$96.4\,\text{ms} \pm 62.9\,\text{ms}$
Upscale to sensor size	$47.8\,\text{ms} \pm 39.5\,\text{ms}$	$99.3\,\text{ms} \pm 33.2\,\text{ms}$	$122\,\text{ms} \pm 14.3\,\text{ms}$

Table 5. The runtime details of the edge extraction pipeline (Fig. 2 in the main paper). This includes the following: (i) preprocessing, (ii) edge detection, and (iii) edge smoothing components, the optimization routine, and upscaling/downsampling routines.

References

[1] Vincent Brebion, Julien Moreau, and Franck Davoine. Real-time optical flow for vehicular perception with low-and high-resolution event cameras. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):15066–15078, 2021.

[2] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 525–533, 2022.

[3] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras. In *Proceedings of the International Conference on 3D Vision*, pages 197–206. IEEE, 2021.

[4] 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 *Proceedings of the European Conference on Computer Vision*, pages 366–382. Springer, 2020.

[5] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9685–9694, 2023.

[6] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[7] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Secrets of event-based optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 628–645. Springer, 2022.

[8] Alex Zihao 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] https://jax.readthedocs.io/en/latest/_autosummary/jax.jit.html