

# Lattice-allocated Real-time Line Segment Feature Detection and Tracking Using Only an Event-based Camera

## Supplementary Material

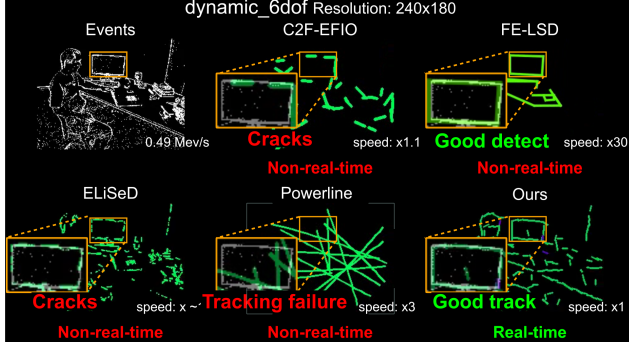


Figure 10. The part of the attached video representing line segments results with ECD dataset dynamic\_6dof from four baselines and our method. Our method outperforms others in terms of processing speed and accuracy.

Baselines	Hyper-parameters	
ELiSeD [4]	tolerance angle [deg]	11.25, <u>22.5</u> , 45
Powerline [9]	None	
C2F-EFIO [7]	None	
FE-LSD [34]	threshold	0.15, 0.3, <u>0.6</u> , 0.9
Ours	$\alpha$	0.7, <u>1.0</u> , 1.3
	threshold $f_{th}$	0.15, <u>0.2</u> , 0.25
	block size $b$ [px]	6, <u>8</u> , 10
	240x180:	
	346x260:	<u>10</u>
	640x480:	<u>14</u>
	perturbation size $\Delta q$ [px]	
	240x180:	<u>0.8</u>
	346x260:	<u>1.1</u>
	640x480:	<u>2.5</u>
	distance threshold $d_{max}$ [px]	<u>0.2</u> $\times b$

Table 5. Hyper-parameters from each baseline for evaluations. All bold values with underlines are default hyper-parameters for qualitative evaluation. Other values are used for quantitative evaluations, such as Precision and Recall distributions of line accuracy and sensitivity evaluations.

## 6. Videos

We encourage readers to inspect the attached video as a supplementary material. As shown in Fig. 10, this video summarizes line segment results from all methods with all datasets. Especially, the demo with a modern online event camera (640x480) is also presented, proving that our method works in real time in the real-world environment.

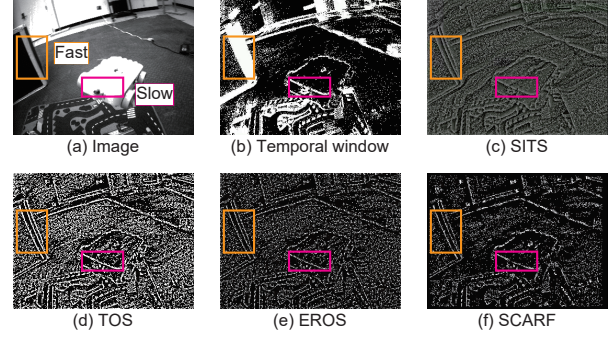


Figure 11. Qualitative comparison in velocity-invariant event-based representations (b) temporal windows [20], (c) SITS [1], (d) TOS [14], (e) EROS [15], and (f) SCARF with the “boxes\_seq.01” in EVIMO [27] dataset. As shown in the (a) Image, EVIMO includes multiple objects moving separately at different speed, useful for motion segmentation tasks.

## 7. Experimental details

Table 5 summarizes the hyper-parameter settings used for each method. Powerline and C2F-EFIO do not include tunable hyper-parameters for balancing precision and recall in line segment accuracy. In contrast, ELiSeD, FE-LSD, and our method allow control over this trade-off via threshold values, resulting in the PR curves shown in Fig. 8. The units of the block size  $b$ , the perturbation size  $\Delta q$ , and the distance threshold  $d_{max}$  are in pixels; their values are scaled according to the resolution of each dataset.

## 8. Qualitative Evaluation in Velocity-invariant Event-based Representations

Figure 11 compares various velocity-invariant event-based representations. Although the temporal window contains motion blur caused by fast-moving objects, all methods mitigate this effect to some extent. However, (c) SITS, (d) TOS, and (e) EROS introduce artifacts across the entire field of view. In contrast, (f) SCARF effectively removes redundant events and preserves sharp edges of multiple objects moving at different speeds. This is achieved through the FIFO mechanism within the buffer in each block and the “inhibitory” role of inactive events.

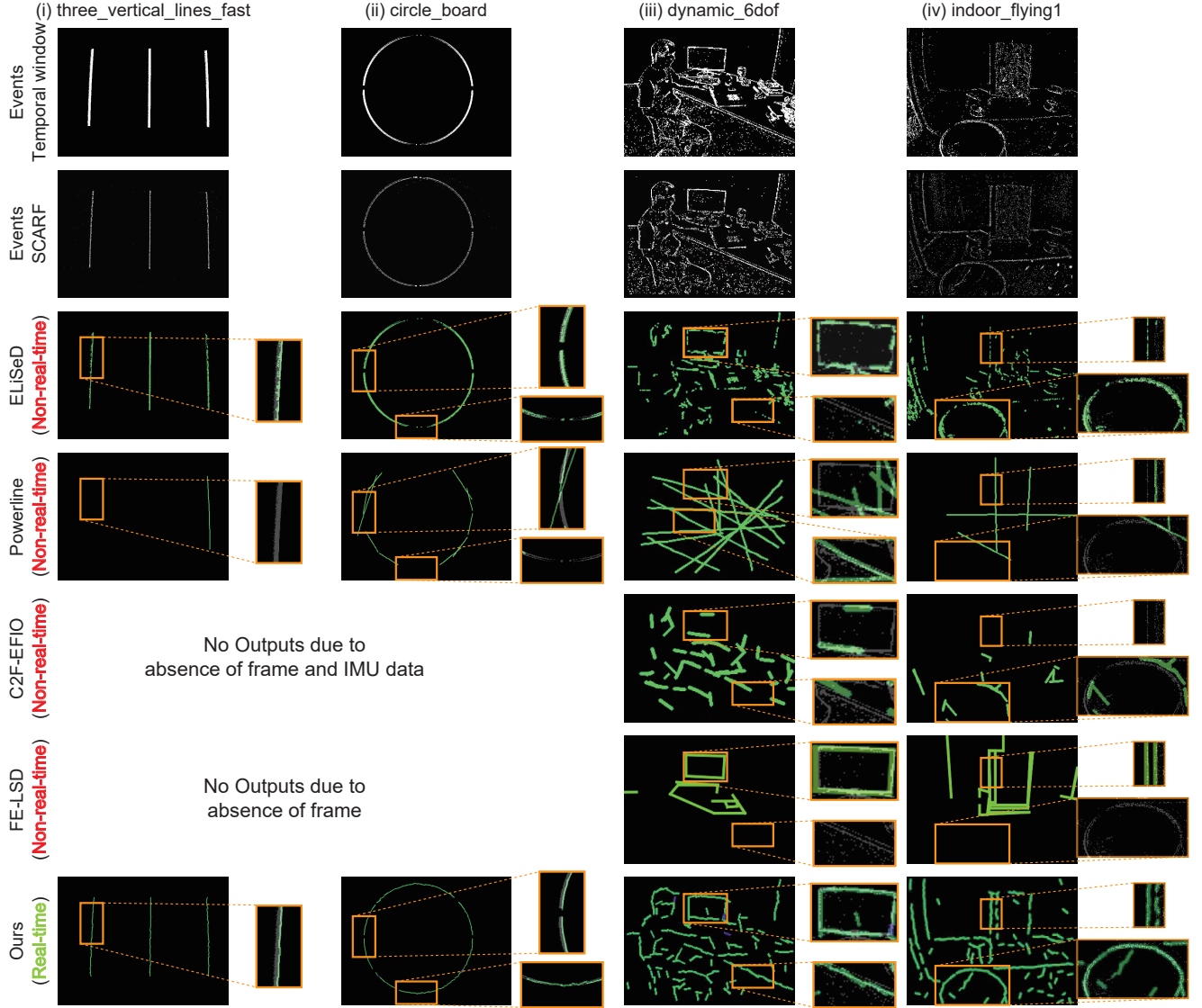


Figure 12. Detailed visualized results of line segments from all methods with datasets used in Sec. 4.3. The zoomed-in images illustrate the overlay of events (gray) and the extracted line segments (green: tracked, blue: detected only in ours).

## 9. Visualization of Line Segments Results in Quantitative Evaluation

Figure 12 visualizes line segments from all methods with datasets used in quantitative evaluation Sec. 4.3. The fast motion in the dataset “three\_vertical\_lines\_fast” generated a larger number of events, which in turn led to tracking failures in Powerline. In contrast, in the dataset “circle\_board”, the vertical motion is limited, so horizontal line segments are omitted due to the lack of events. This difference in results stems from the fact that conventional storage for event data is velocity-variant. On the other hand, our method, SCARF, stores event data in a velocity-invariant manner,

allowing our approach to consistently extract line segments under both fast and slow motion conditions.

## 10. Limitations

The average lifetime of line segments extracted by ours is shorter than Powerline. As shown in Table 6 and Fig. 13, to evaluate the lifetime of line segments, we computed the average and maximum duration of line segment extracted during the first 10 seconds when processing the public dataset “dynamic\_6dof” and the recorded dataset “dynamic\_6dof”. ELiSeD is based on pixel-wise non-parametric approach, meaning that the line segments become short-lived. Even

dynamic_6dof (240x180)				
	Num. line segments	mean	std	max
ELiSeD	7420	0.08	0.16	2.13
Powerline	43	3.01	3.07	10.16
C2F-EFIO	1425	0.43	0.56	3.53
Ours	2688	0.97	1.11	13.32

monitor_6dof (640x480)				
	Num. line segments	mean	std	max
ELiSeD	1815	0.06	0.08	0.80
Powerline	21	6.69	5.60	14.85
Ours	8989	0.40	0.57	11.25

Table 6. Lifetime of line segment extracted by four methods including tracking during the first 10 seconds with the dataset “dynamic\_6dof”

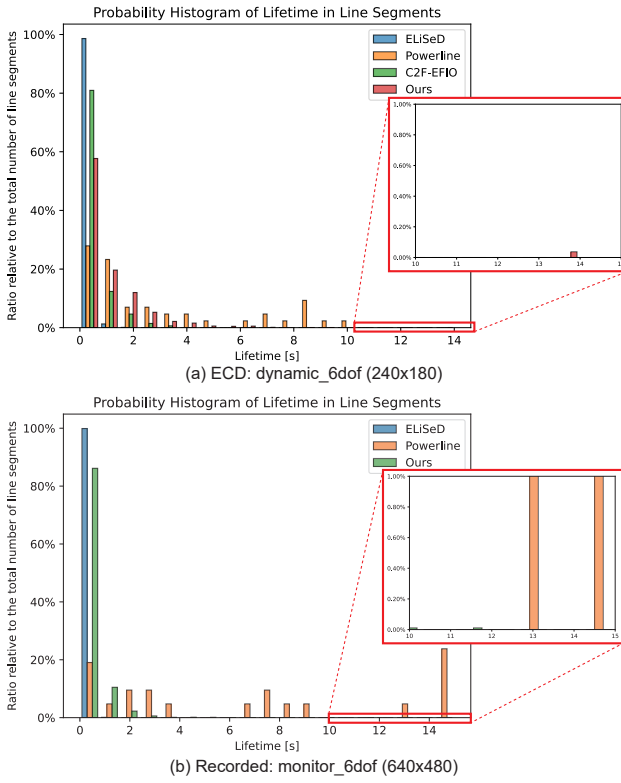


Figure 13. Probability histograms of line segment lifetime from ELiSeD, Powerline, C2F-EFIO, and ours. The zoomed-in graph demonstrates that our method successfully extracts long-lived line segments, corresponding to Powerline.

the maximum lifetime of line segments in ELiSeD was about 2 seconds in “dynamic\_6dof” and shorter than 1 second in “monitor\_6dof”. On the other hand, Powerline enabled to keep long-lived line segments due to spatio-temporal based approach. Compared to the three baselines, our method extracted a larger number of line segments, resulting in a longer average lifetime than ELiSeD and C2F-EFIO but shorter than Powerline. However, on the benefit

of tracking, the same line segment ID could be maintained for a longer duration, bringing the maximum lifetime longer than C2F-EFIO and closer to Powerline. Furthermore, clustering multiple line segments from blocks into one “strong” line segment would improve temporal consistency.