Robust Event-based Angular Velocity Estimation in Dynamic Environments

Sangil Lee

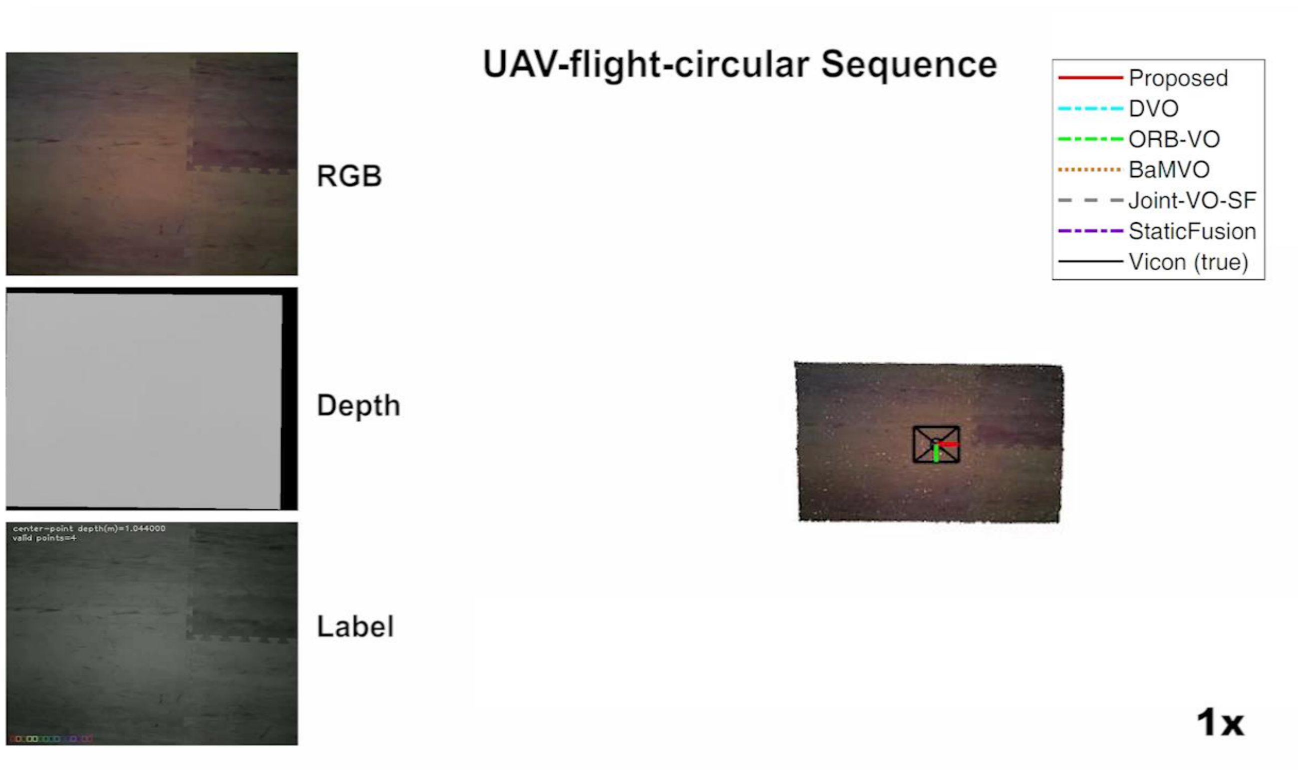
- **Affiliation**: Laboratory for Autonomous Robotics Research (LARR),
  Seoul National University
- **Advisor**: H. Jin Kim
- **Committee**: Youdan Kim
  Chan Gook Park
  Ayoung Kim
  Pyojin Kim

Contents

Chapter 1

Motivation

- Real work environment is apparently different from well-defined environment.
- RGB-D cameras have a narrow range of activity (indoor) and have the same limitation as the conventional frame-based cameras. (motion blur, low dynamic range, etc.)



Video 1. Robust RGB-D visual odometry in dynamic environment [1]

Video 2. Real-time Object-aware Monocular Depth Estimation in Onboard Systems [2]

[1] Lee, Sangil, Son, Clark Youngdong, and Kim, H Jin, “Robust Real-time RGB-D Visual Odometry in Dynamic Environments via Rigid Motion Model”, IROS, 2019.
[2] Lee, Sangil, Lee, Chungkeun, Kim, Haram, and Kim, H Jin, “Real-time Object-aware Monocular Depth Estimation in Onboard Systems,” IJCAS, 2021.

Motivation

- The frame-based camera produces images at a fixed frame rate.
- Thus, there exist blind time and exposure time.
- Every pixel of the event-based camera detects the change in light intensity individually.

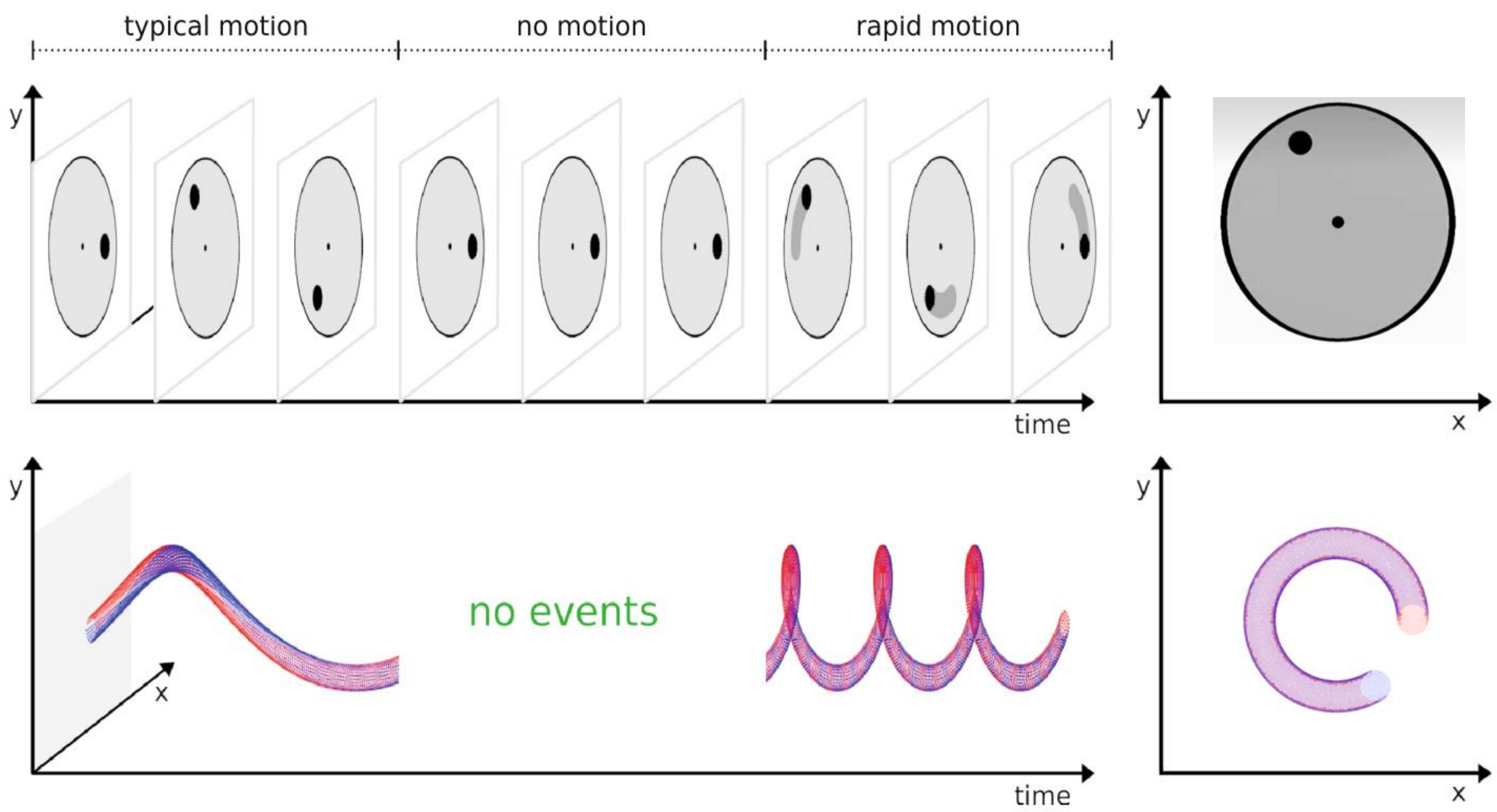


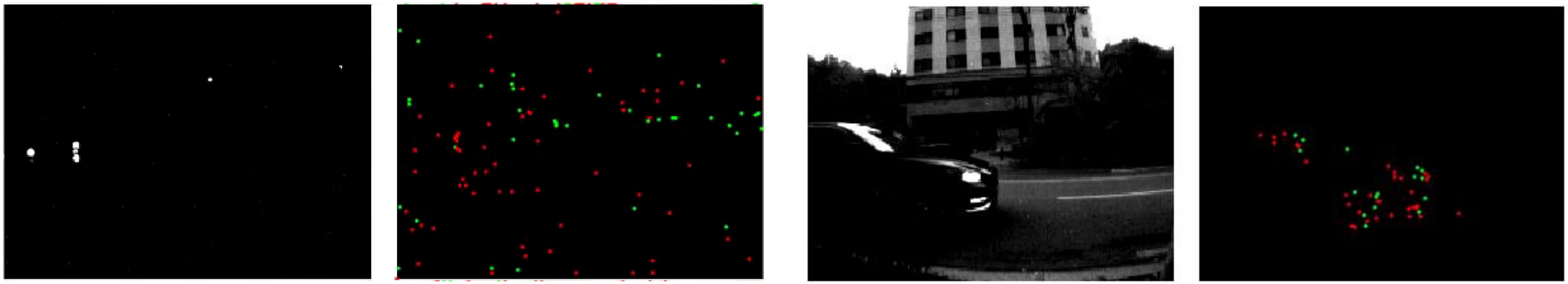
Figure 1. Frames vs. Events. Adopted from [3,4].

Video 3. Frames vs. Events in HDR scene.

[3] E. Mueggler, et al., “Event-based, 6-dof pose tracking for high-speed maneuvers,” IROS, 2014.
[4] H. Kim, et al., “Real-time 3d reconstruction and 6-dof tracking with an event camera,” ECCV, 2016.

Challenges

- Due to these characteristics of the event camera, it has a large potential to be performed well in a dynamic environment including vehicle and human's interaction.
- Challenge 1: Since event camera is a kind of data-driven sensors, it is difficult to perceive scene in event stream. ⇒ Ch. 3 and 4
- Challenge 2: Because of the same reason, it is hard to distinguish between background and foreground object in the stationary camera. ⇒ Ch. 5



(a) Road at night (related to **C1**) (b) Car observed from stationary camera (related to **C2**)

Video 4. Frames vs. Events in various environments

Research Objectives

- **Objective 1:** We propose an asynchronous optical flow stream that accurately estimates angular velocity with a low latency.
- **Objective 2:** We robustly estimate angular velocity in a dynamic environment using optical flow stream and dual-mode motion model.

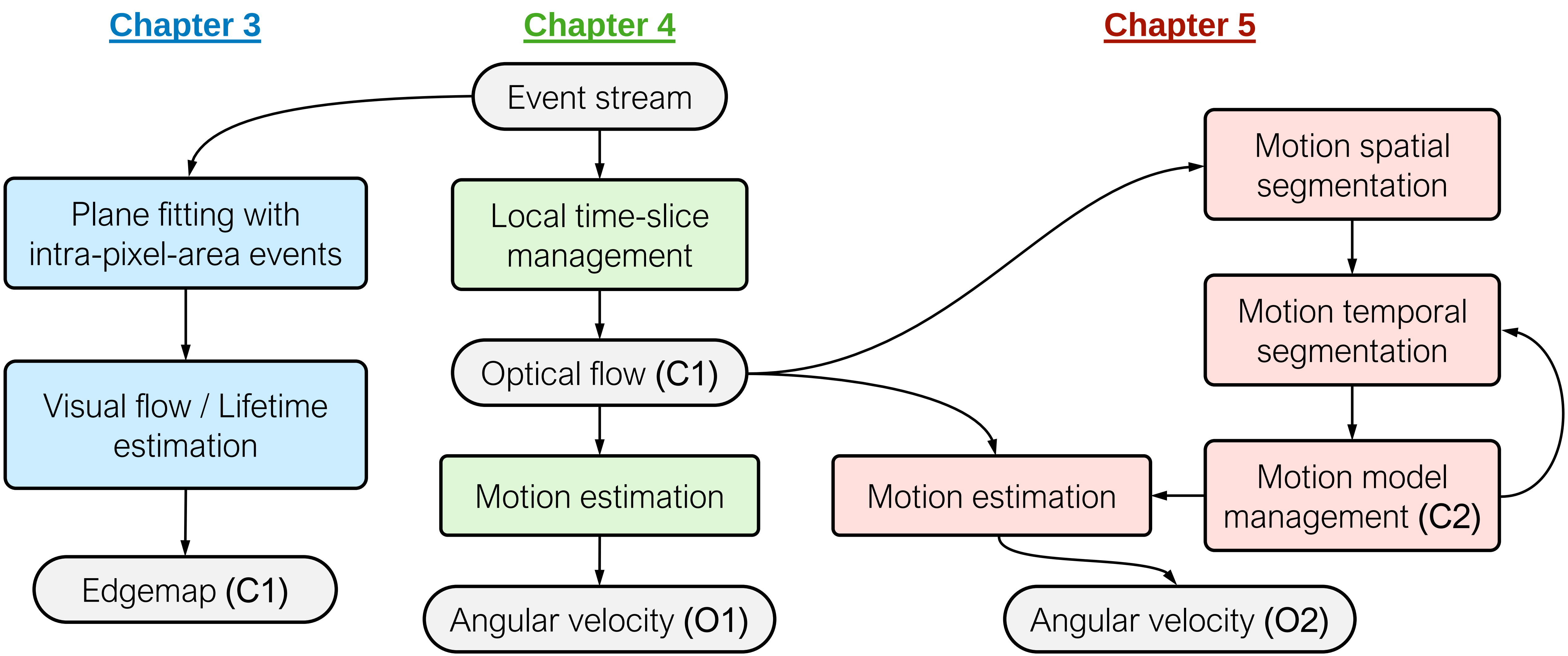


Figure 2. Flowchart of the dissertation

Representative Papers (Cont'd)

Optical Flow Estimation

- R. Benosman, et al., "Event-based Visual Flow." IEEE Trans. Neural Netw. Learning Syst., 2014.
- E. Mueggler, et al., "Lifetime Estimation of Events from Dynamic Vision Sensors," ICRA, 2015.

  ⇒ These visual flows are vulnerable to timestamp noise and aperture problem.

- M. Liu and T. Delbruck, "Adaptive Time-Slice Block-Matching Optical Flow Algorithm for Dynamic Vision Sensors," BMVC, 2018.
- C. Lee, et al., "Spike-Flownet: Event-based Optical Flow Estimation with Energy Efficient Hybrid Neural Networks," ECCV, 2020.
- A. Z. Zhu, et al., "EV-Flownet: Self-supervised Optical Flow Estimation for Event-based Cameras," arXiv preprint, 2018.
- A. Z. Zhu, et al., "Unsupervised Event-based Learning of Optical Flow, Depth, and Ego motion," CVPR, 2019.

  ⇒ Event-frame-based flow map estimation algorithms have a large latency.

Representative Papers

Angular Velocity Estimation

- G. Gallego, et al., “A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation,” CVPR, 2018.
- M. Gehrig, et al., “Event-based Angular Velocity Regression with Spiking Networks,” ICRA, 2020.

  ⇒ Event-frame-based angular velocity estimation algorithms have a large latency and their parameter configuration highly depends on a scene.

Motion Segmentation/Estimation in Dynamic Environments

- T. Stoffregen, et al., “Event-based Motion Segmentation by Motion Compensation,” ICCV, 2019.
- D. Falanga, et al., "Dynamic Obstacle Avoidance for Quadrotors with Event Cameras," Science Robotics, 2020.

  ⇒ Motion-compensated image of warped events is easily corrupted by outliers.

- A. Mitrokhin, et al., "EV-IMO: Motion Segmentation Dataset and Learning Pipeline for Event Cameras," IROS, 2019.

  ⇒ Moving objects occupy a small portion of the image frame, and the camera has to keep moving.

Contributions and Outlines

- Chapter 3: Visual Flow with Intra-pixel-area Events
  I. Intra-pixel-area Events for visual flow to be estimated robustly to data noise
  II. Lifetime estimation and edge map detection with accurate visual flow
  ⇒ "Edge Detection for Event Cameras using Intra-pixel-area Events," BMVC, 2019.

- Chapter 4: Low-latency and Scene-robust Optical Flow
  I. Low-latency optical flow estimates angular velocity accurately
  II. Scene-robust optical flow shows stable performance with consistent latency under 15ms
  ⇒ "Low-latency and Scene-robust Optical Flow Stream and Angular Velocity Estimation," Access, 2021.

- Chapter 5: Robust Angular Velocity Estimation in Dynamic Environments
  I. Dual-mode motion model detects moving objects without prior information
  II. Robustness to moving objects for angular velocity estimation
  ⇒ "Real-time Rigid Motion Segmentation using Grid-based Optical Flow," SMC, 2017.
  ⇒ "Robust Real-time RGB-D Visual Odometry in Dynamic Environments via Rigid Motion Model," IROS, 2019.

Chapter 2

Event Camera

- Event camera senses logarithmic scale of intensity, detects the change, and verifies the polarity.
- Event camera produces asynchronous event:

$$e_k = (\mathbf{x}_k, t_k, p_k)$$

with pixel position $\mathbf{x}_k$, timestamp t_k, polarity p_k.

- DAVIS240C produces images and events.

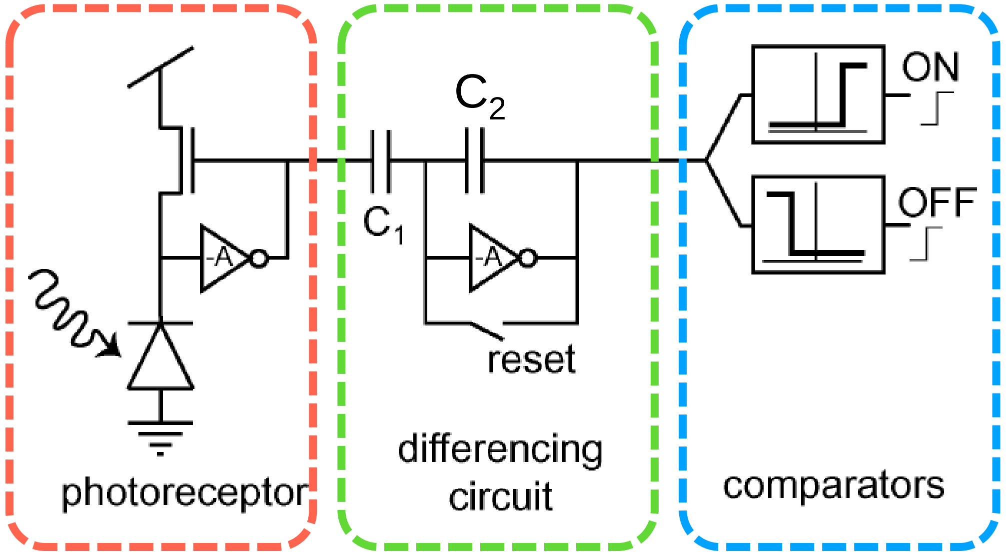
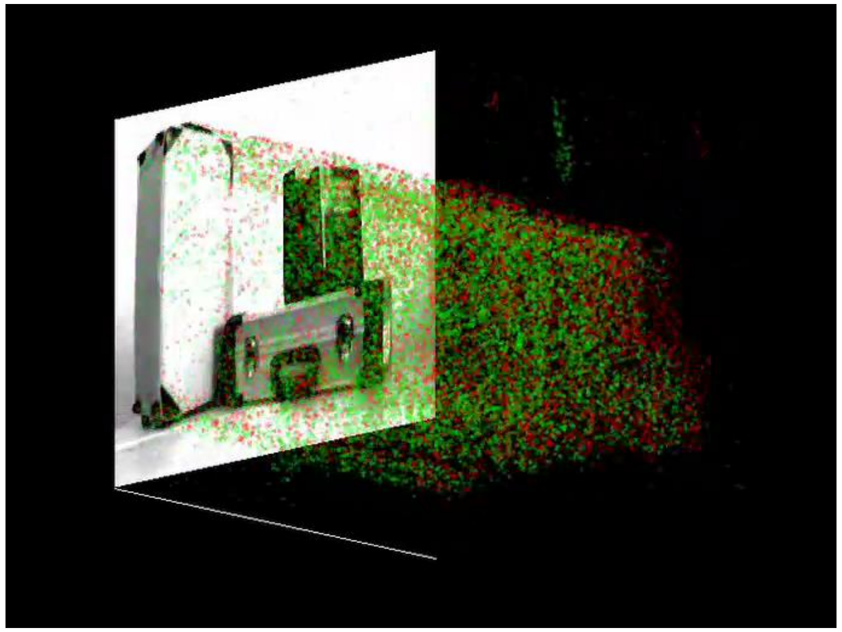


Figure 3. The abstracted schematic of event camera

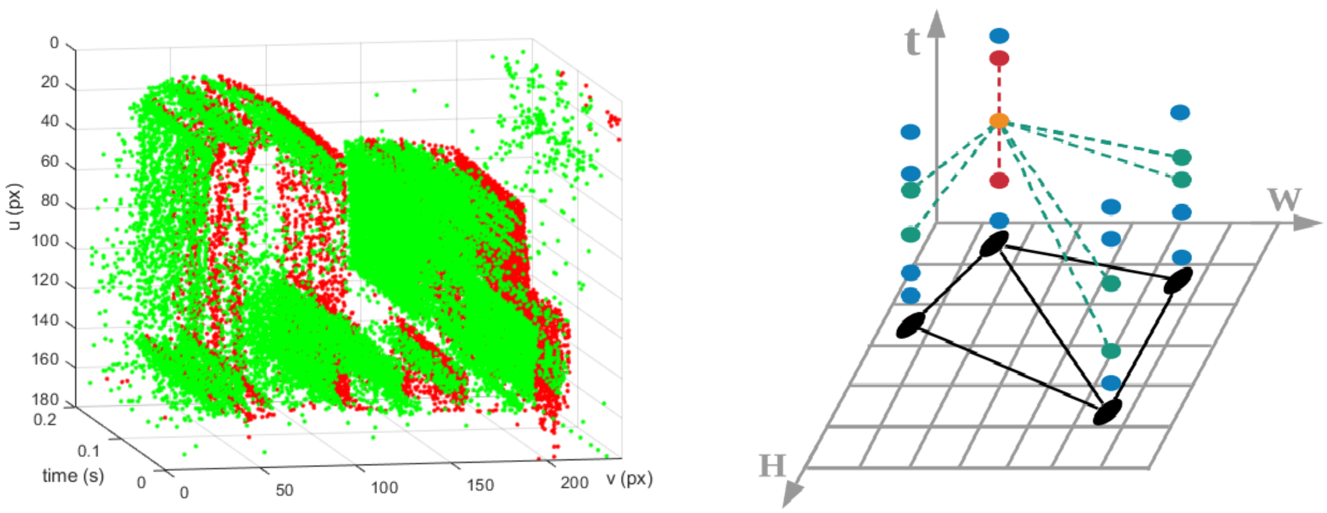
Figure 4. DAVIS240C event camera used in presentation



Video 5. The event of DVS and the image of APS.

Event Representations (Cont'd)

- Event stream: single event is fetched into the algorithm.
- Event packet: multiple events are fetched into the algorithm. The size of event packet can be decided by the number or time window of them.
- 3D spatio-temporal points: 3D points set in (x, y, t) space



(a) Event stream (b) Spatio-temporal graph

Figure 5. Various event representations.

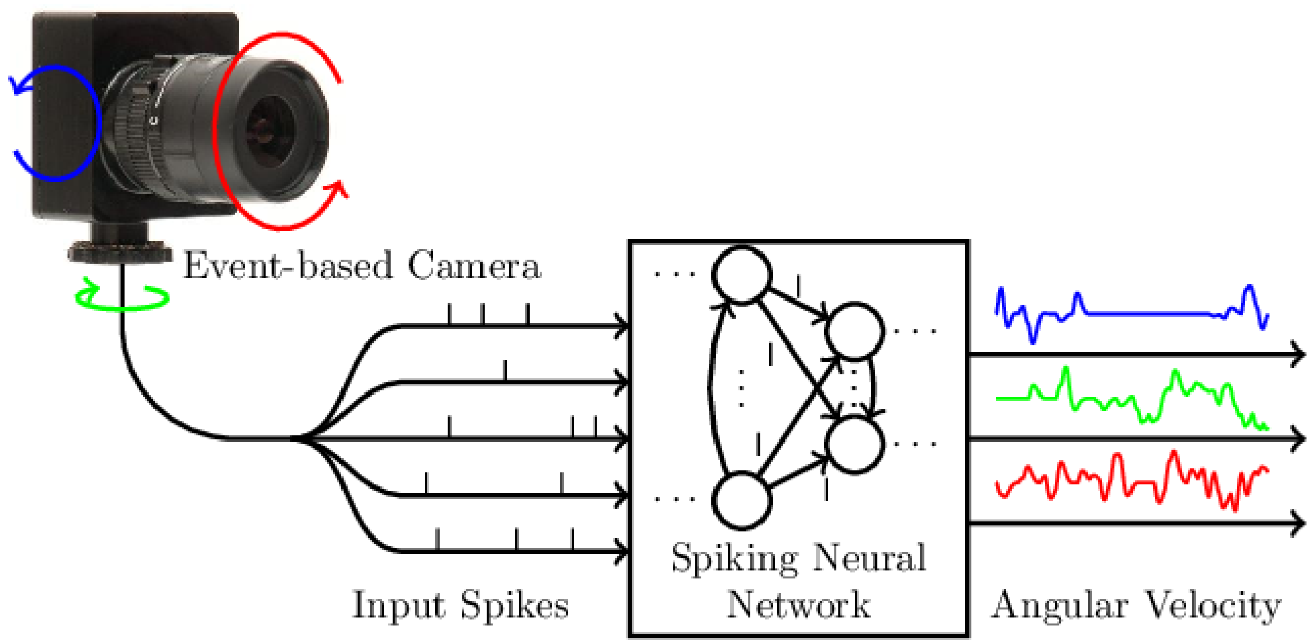
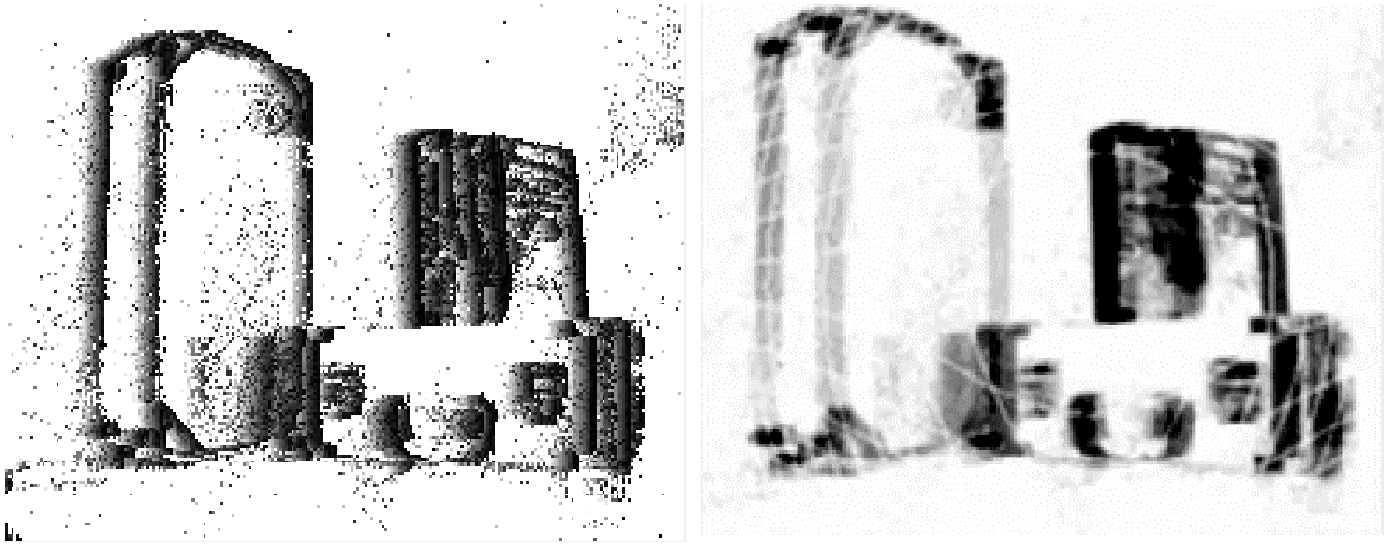


Figure 6. Spiking neural network that predicts angular velocity

Event Representations

- **Image of stacked events (a.k.a. SAE, time slice)**: image stacked by the timestamp of event packet
- **Motion-compensated image of events (a.k.a. IWE)**: the count map of events warped by motion



(a) Image of stacked events (b) Motion-compensated image

Figure 7. Various event representations.

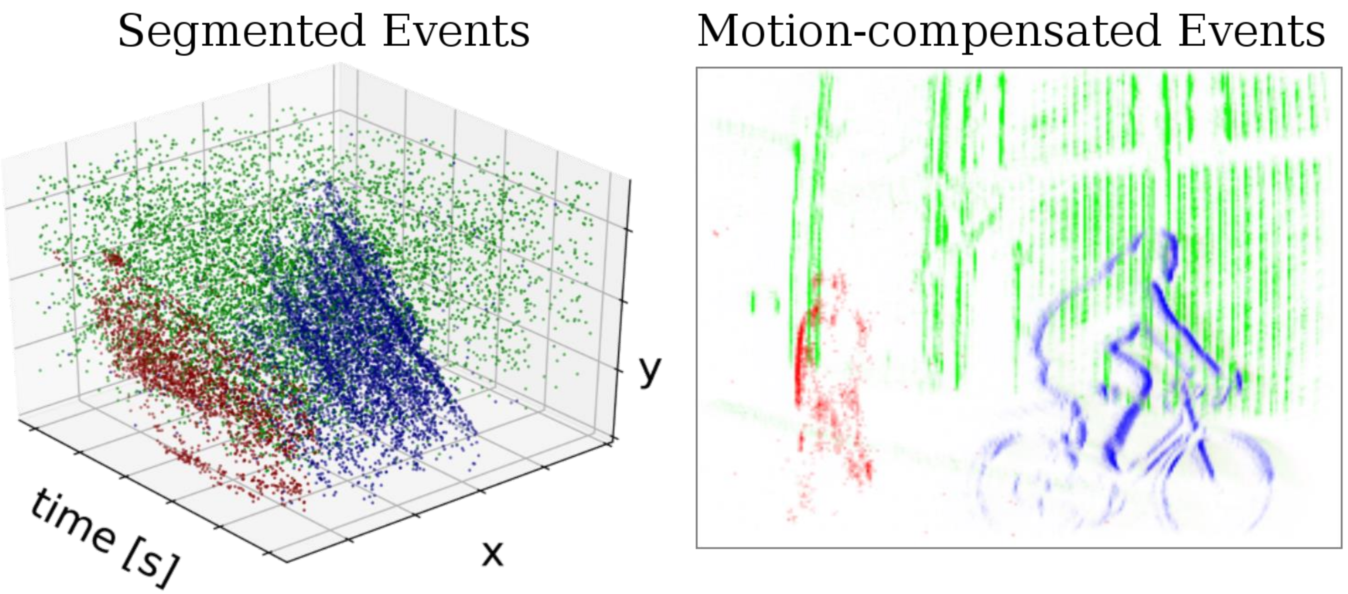


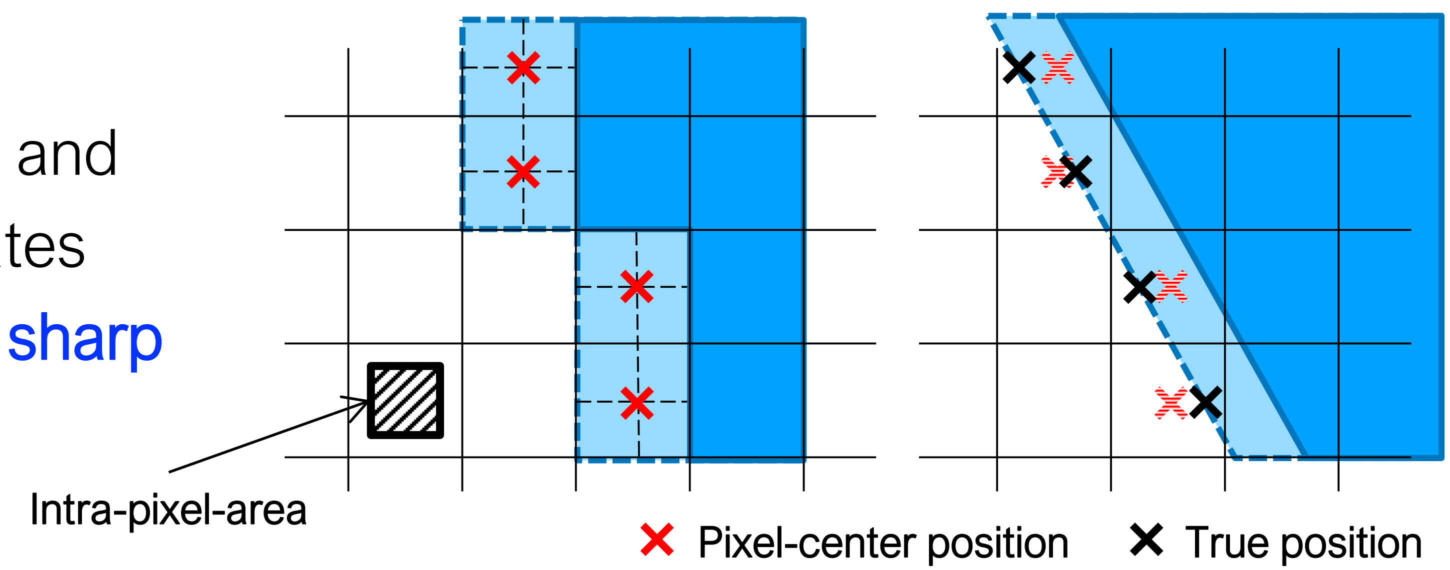
Figure 8. Motion segmentation by motion compensation.

Chapter 3

Objective and Contributions

Objective

- Introduce intra-pixel-area events and develop an algorithm that estimates accurate visual flow and detects sharp edge map



Contributions

- Enhance the visual flow and lifetime estimation using intra-pixel-area event
  : estimates visual flow and lifetime accurately.

- Detect semi-dense edge map from sparse event stream
  : shows high similarity to the Canny edge (GT)

Visual Flow and Lifetime

- SAE, $\Sigma_e(\mathbf{x})$, is defined as

$$\Sigma_e : \mathbb{N}^2 \to \mathbb{R}$$
$$\mathbf{x}_k \mapsto \Sigma_e(\mathbf{x}_k) = t_k$$

- Visual flow can be computed by the gradient vector of the tangent plane fitted on a SAE surface, $\mathbf{n} = (n_1, n_2, n_3)$.

$$v_x = -n_3 / n_1, \;\; v_y = -n_3 / n_2$$

- Then, lifetime of event is the time until the adjacent event is triggered.

$$\tau(\mathbf{x}) = 1[\mathrm{px}]/v[\mathrm{px/s}]$$
$$= \frac{1}{\sqrt{v_x^2 + v_y^2}} = \frac{1}{n_3}\sqrt{n_1^2 + n_2^2}$$

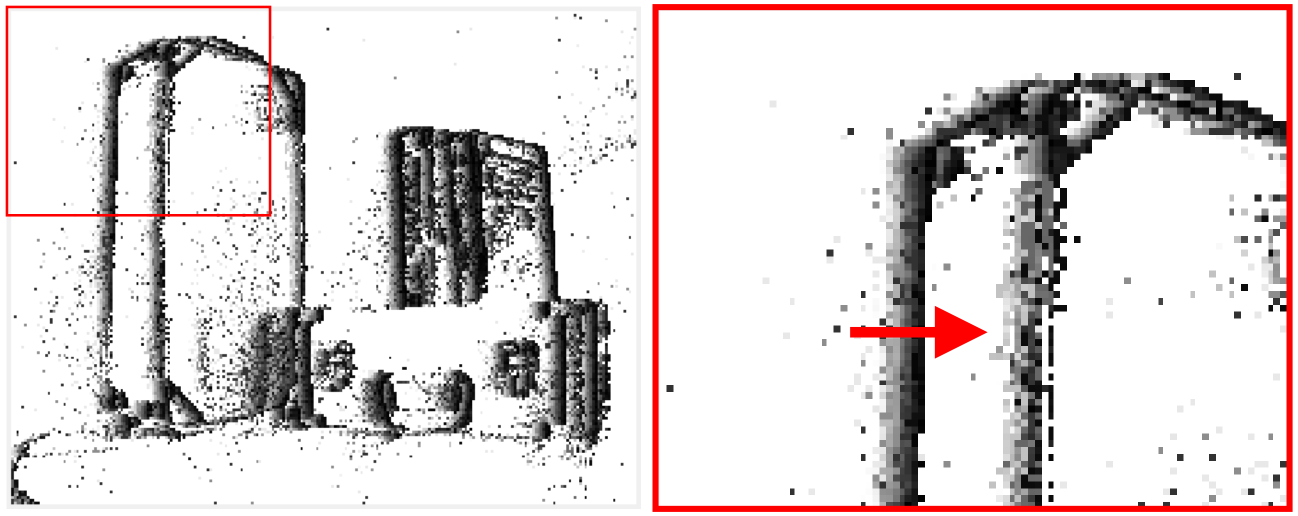


Figure 9. The example of SAE.

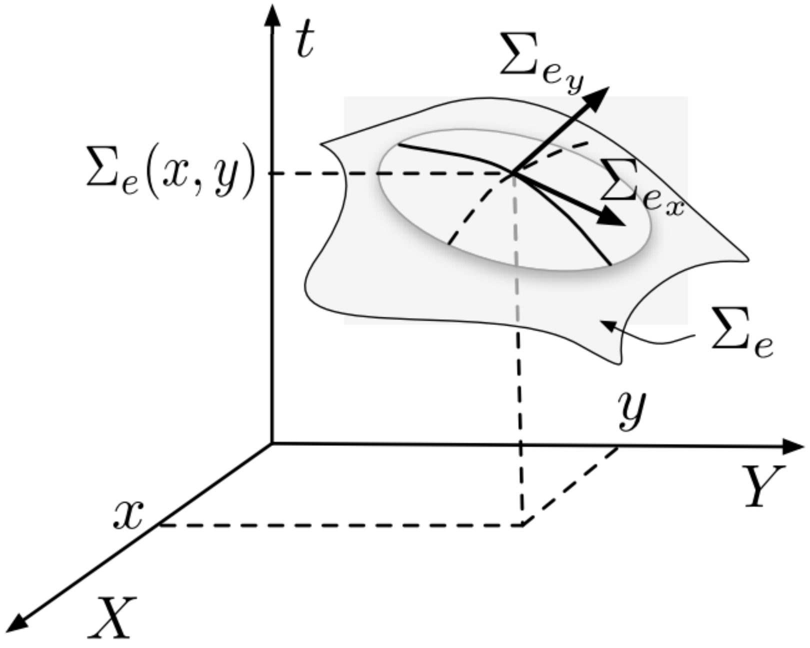


Figure 10. The basic principle of visual flow computation [5].

[5] R. Benosman, et al., “Event-based visual flow,” IEEE Trans. Neural Netw. Learning Syst., vol. 25, no. 2, pp. 407–417, 2014.

Intra-pixel-area Events

- Intra-pixel-area event of an event $e_k = (\mathbf{x}_k, t_k, p_k) = (x_k, y_k, t_k, p_k)$ exists inside

$$S_{\mathbf{x_k}}(\delta) = \{(x, y, t_k) | x_k - \delta < x < x_k + \delta \text{ and } y_k - \delta < y < y_k + \delta\}$$

- Then, the distance function is revised as follows

$$d = dist(\mathbf{n}, \mathbf{z}) \quad \rightarrow \quad d = \min dist(\mathbf{n}, \mathbf{z}), \forall \mathbf{z} \in S_{\mathbf{x}}(\delta),$$

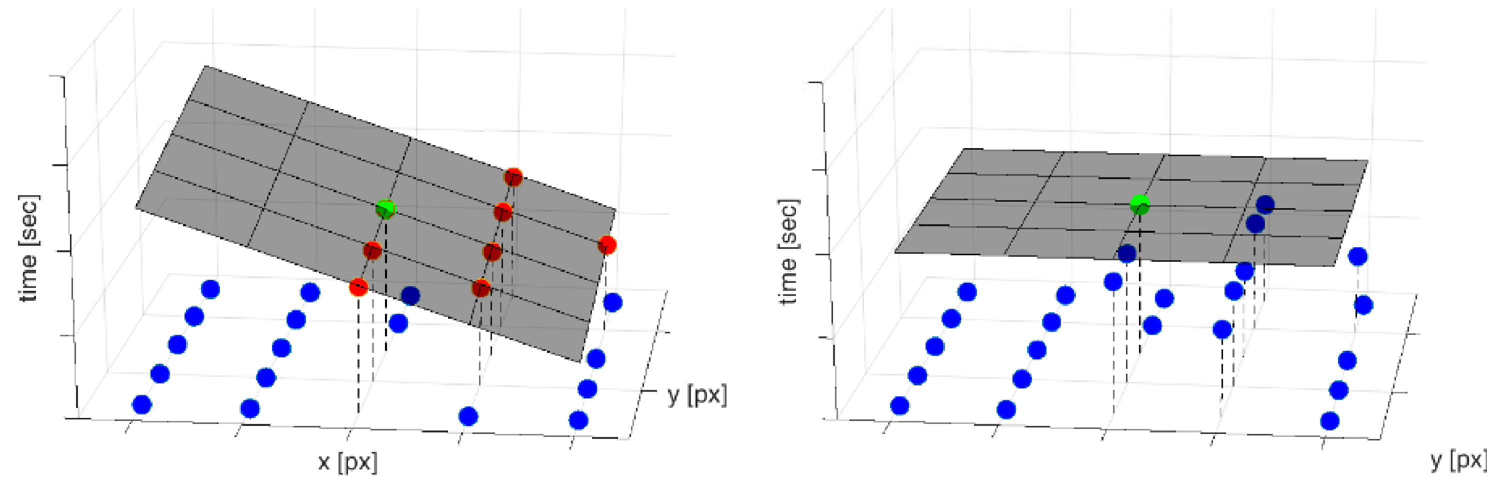
where

$$dist(\mathbf{n}, \mathbf{z}) = \frac{|\mathbf{n}^T \mathbf{z} - 1|}{\|n\|_2^2}$$

- Inlier events are filtered according to $|d| < \epsilon_{th}$ as the same with the original.

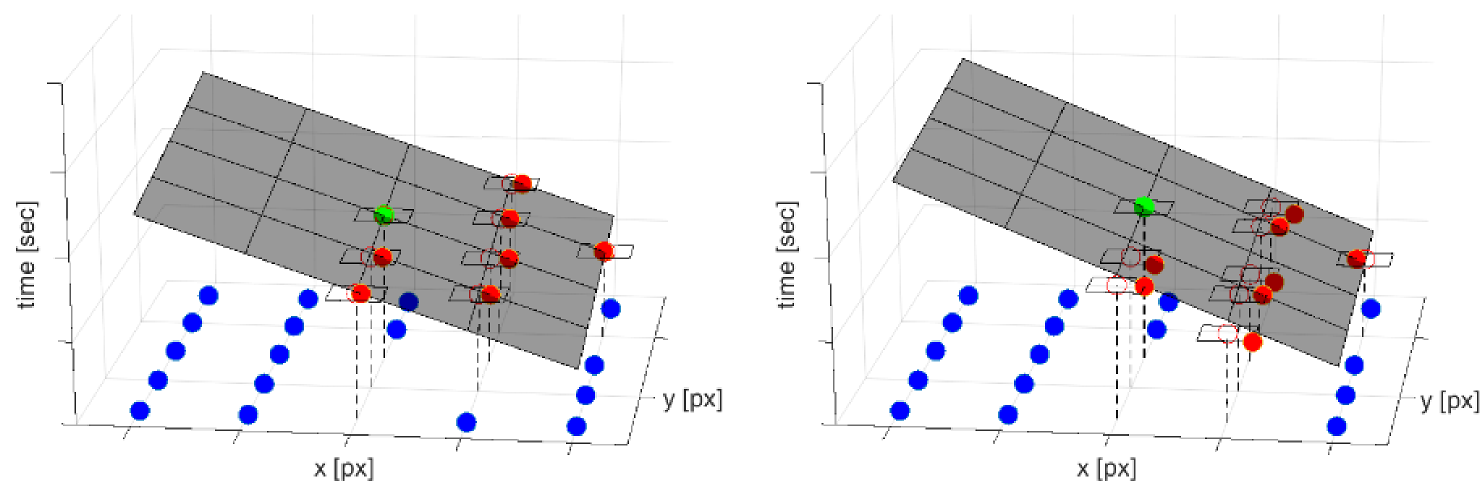
Intra-pixel-area Events

- Intra-pixel-area event makes the fitting plane algorithm more robust against the noise.



(a) w/o intra-pixel-area under mild noise

(b) w/o intra-pixel-area under severe noise



(c) w/ intra-pixel-area under mild noise

(d) w/ intra-pixel-area under severe noise

Figure 11. Description and effectiveness of the intra-pixel-area event. After computing a local plane (gray) w/ or w/o intra-pixel-area event, the outliers (blue), inliers (red), and the current event (green) are drawn.

Intra-pixel-area Events

- In both plots, "$\delta = 0$" means "w/o intra-pixel-area", that is, naïve RANSAC.
- A fitting plane with intra-pixel-area event shows higher F-measure and lower lifetime error within reasonable data noise levels.

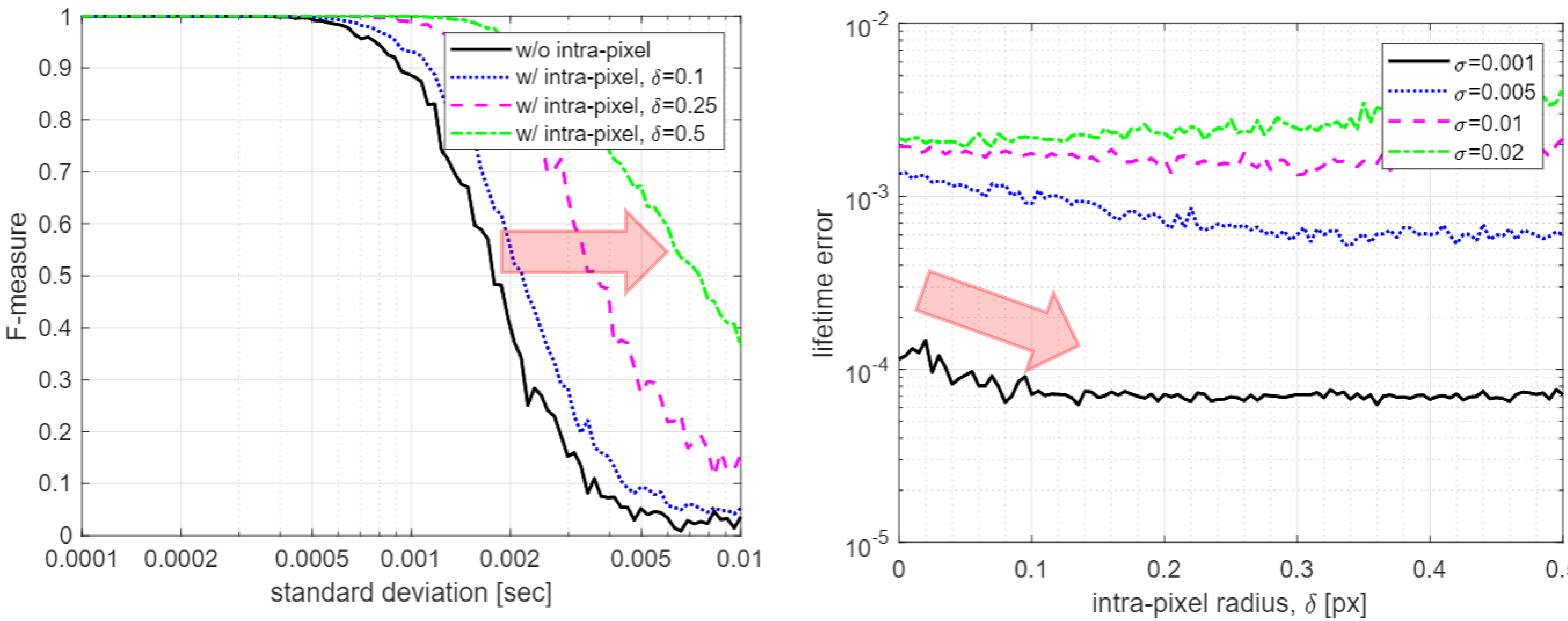
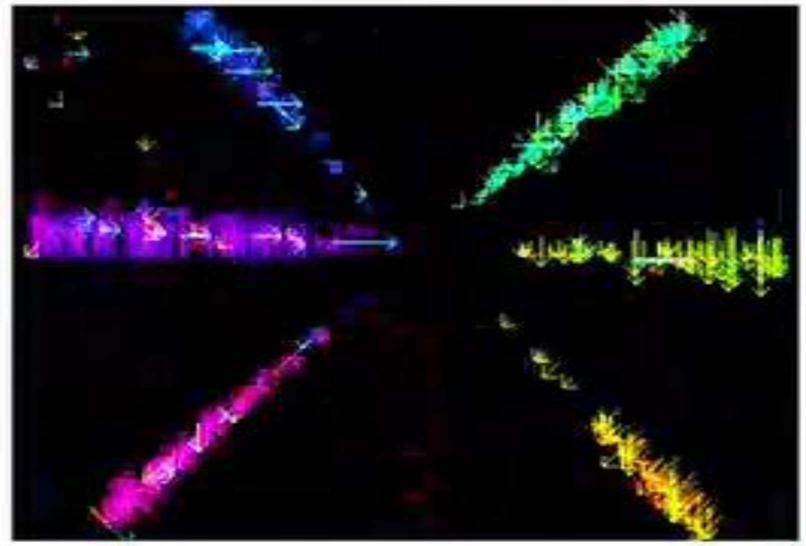


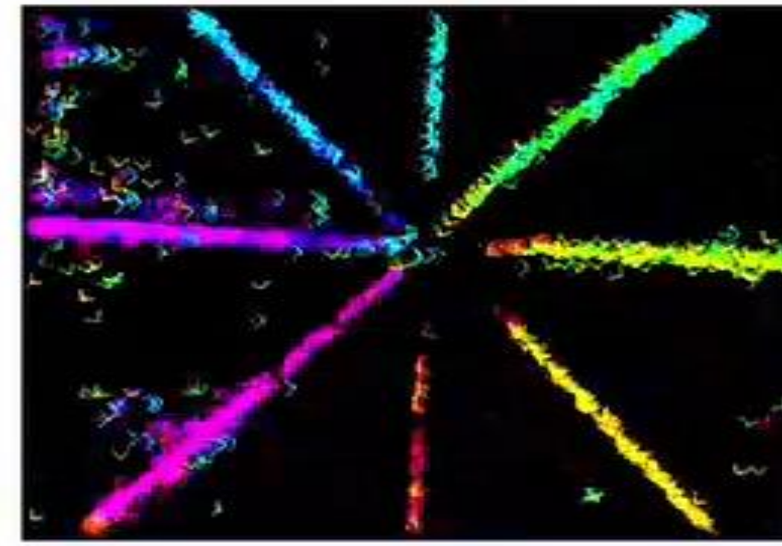
Figure 12. F-measurement evaluation graph with a standard deviation of data noise. (a) F-measure versus standard deviation of data noise depending on intra-pixel radius, (b) lifetime accuracy versus intra-pixel radius depending on data noise.

Evaluation Results

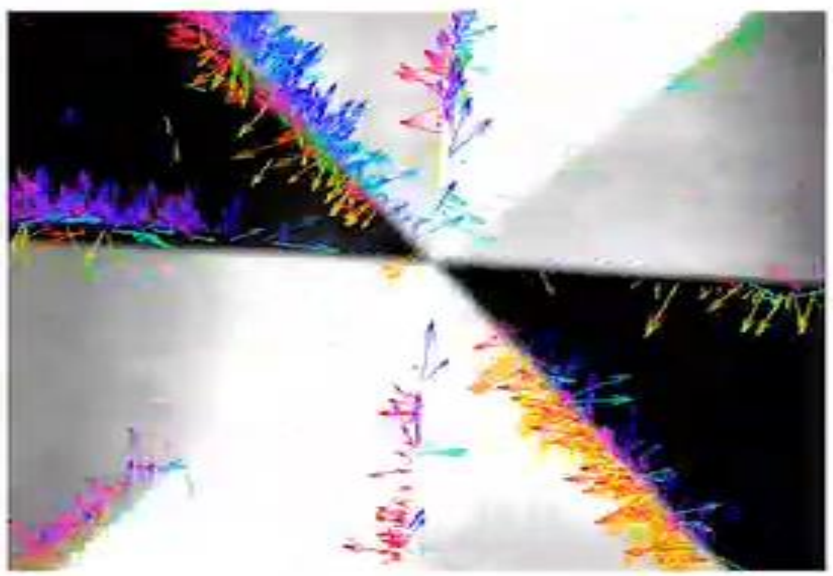
(a) Proposed



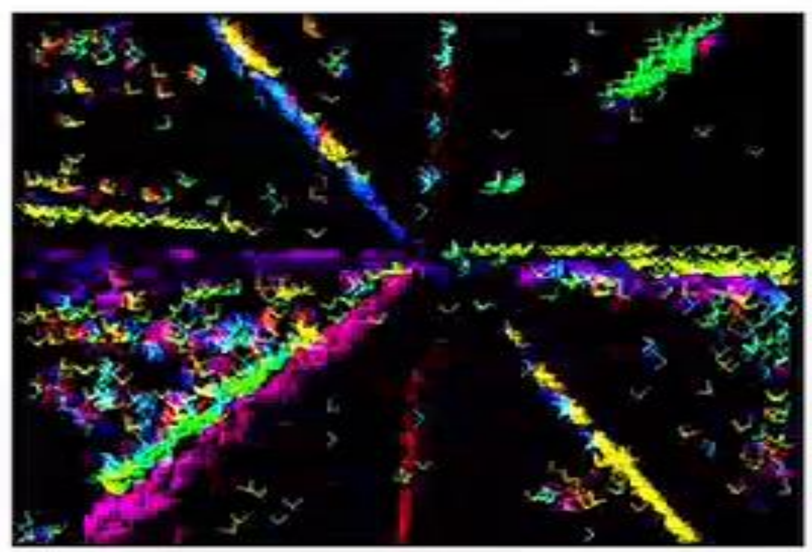
(b) Direction-Selective [7]



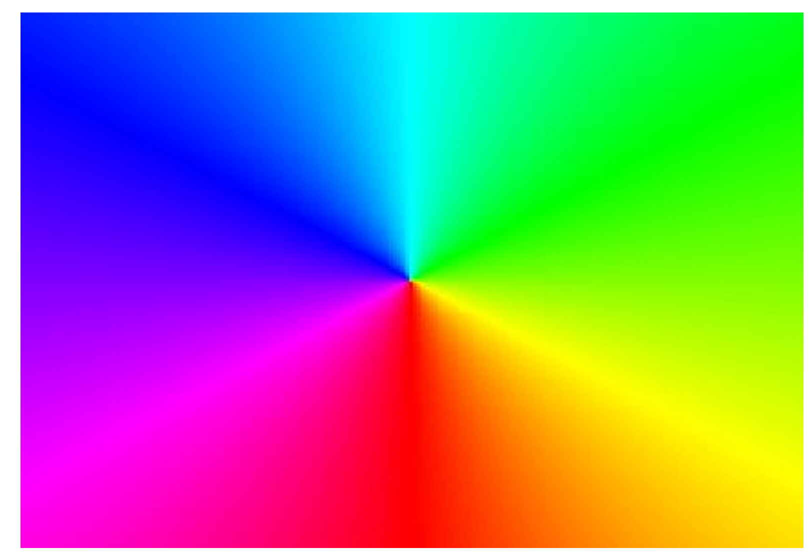
(c) Lucas-Kanade [8]



(d) E. Mueggler [6]



(e) Local Plane [5]



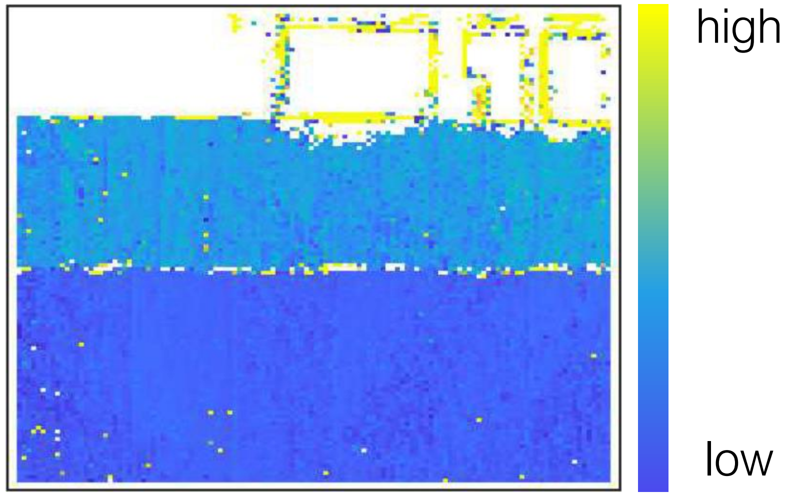
(f) Ground-truth

Video 6. Result of visual flow estimation. The direction of flow vector are represented based on the color wheel.

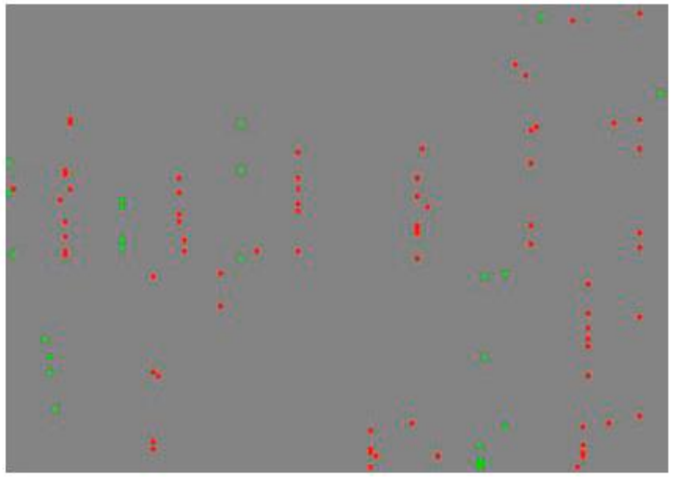
[5] R. Benosman, et al., “Event-based visual flow,” IEEE TNNLS, 2014.
[6] E. Mueggler, et al., “Lifetime estimation of events from dynamic vision sensors,” ICRA, 2015.
[7] T. Delbruck, “Frame-free dynamic digital vision,” in Proceedings of Intl. Symp. on Secure-Life Electronics, 2008.
[8] R. Benosman, et al., “Asynchronous frameless event-based optical flow,” Neural Networks, 2012.

Evaluation Results

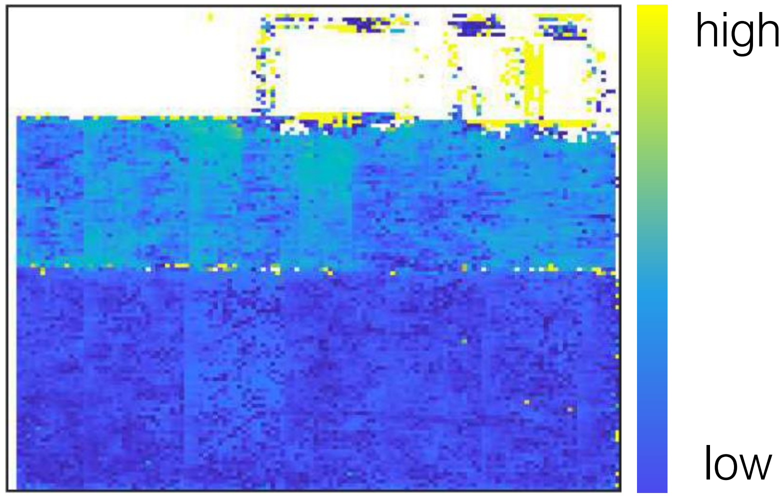
- In bottom figures, the estimated lifetimes are drawn overlaid on a one image when vertical bar moves horizontally.
- Our visual flow shows more consistent and precise results than the existing algorithm.



(a) Proposed

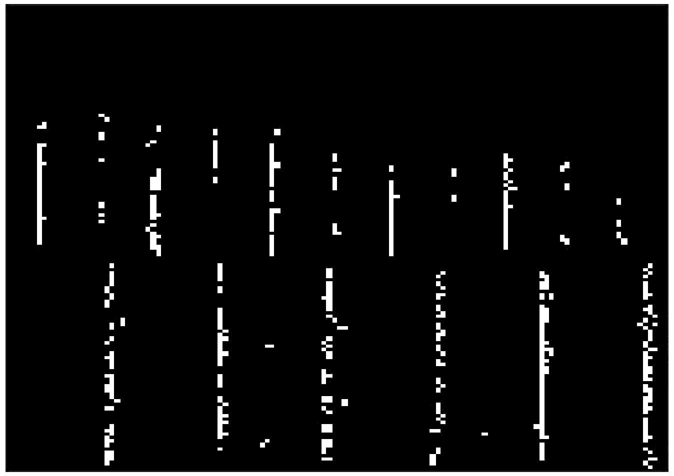


(a) Raw events



(b) E. Mueggler, et al. [6]

Figure 13. The accumulation of lifetime estimates. Yellow color means a large lifetime.



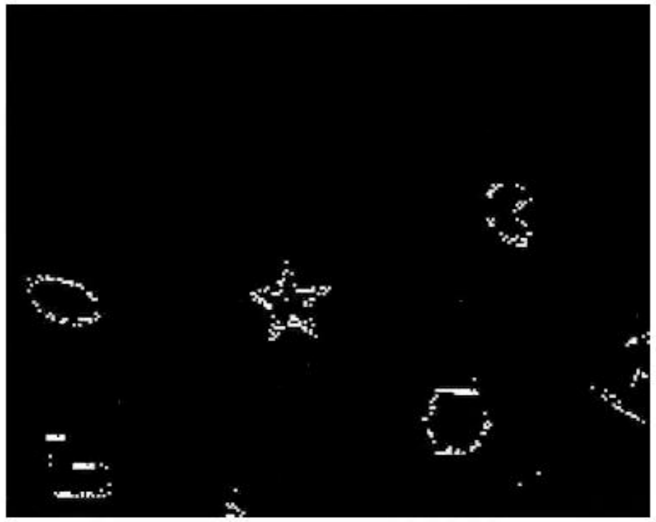
(b) Detected edge

Video 7. Result of edge detection. (top) raw events, (bottom) alive events (detected edge).

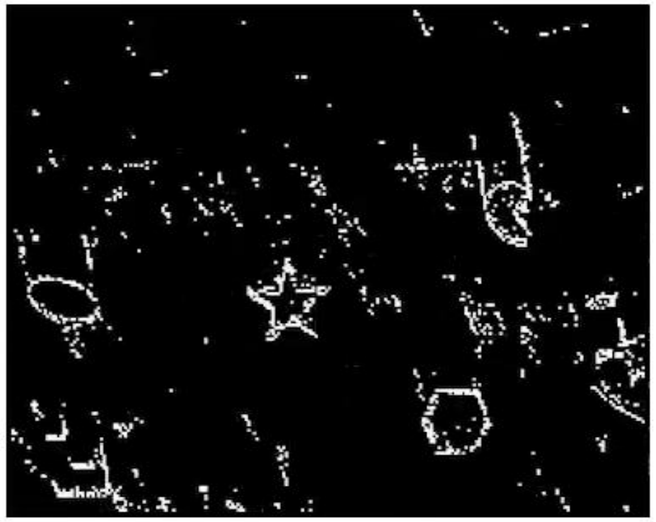
Evaluation Results

- In a complicated scene, the proposed algorithm detects a sharper edge map due to accurate visual flow and lifetime estimation.

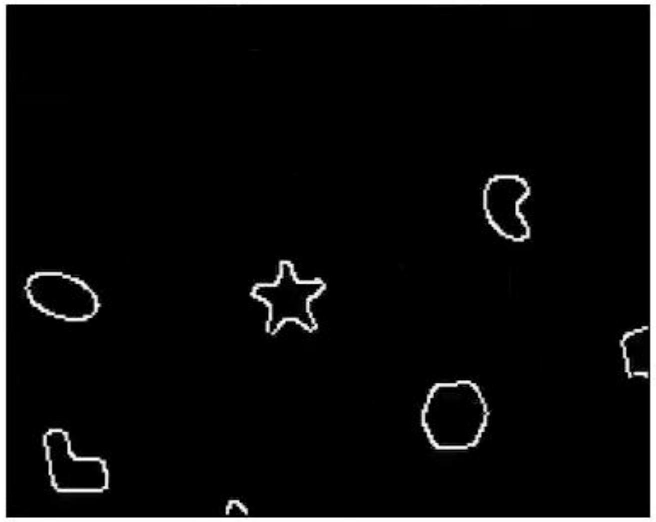
Gray image



1ms accumulation

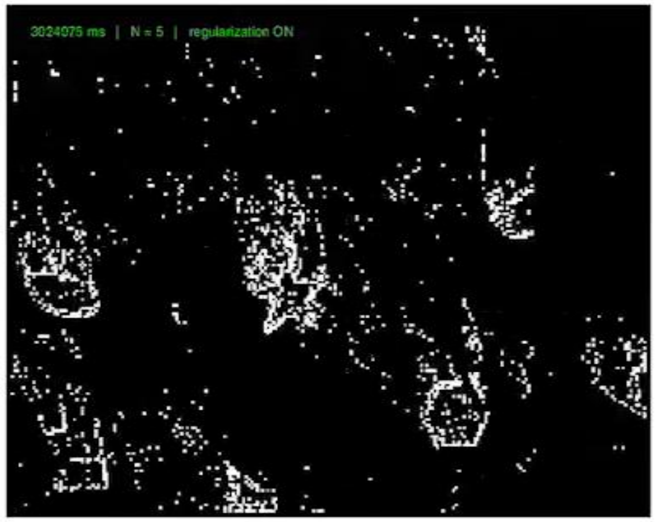


Proposed



Canny edge (GT)

30ms accumulation



E. Mueggler [6]

Video 8. The result of gray image, ground-truth edge map (Canny), Proposed, E. Mueggler, and 30ms, 1ms accumulation in clockwise from top-left.

Evaluation Results

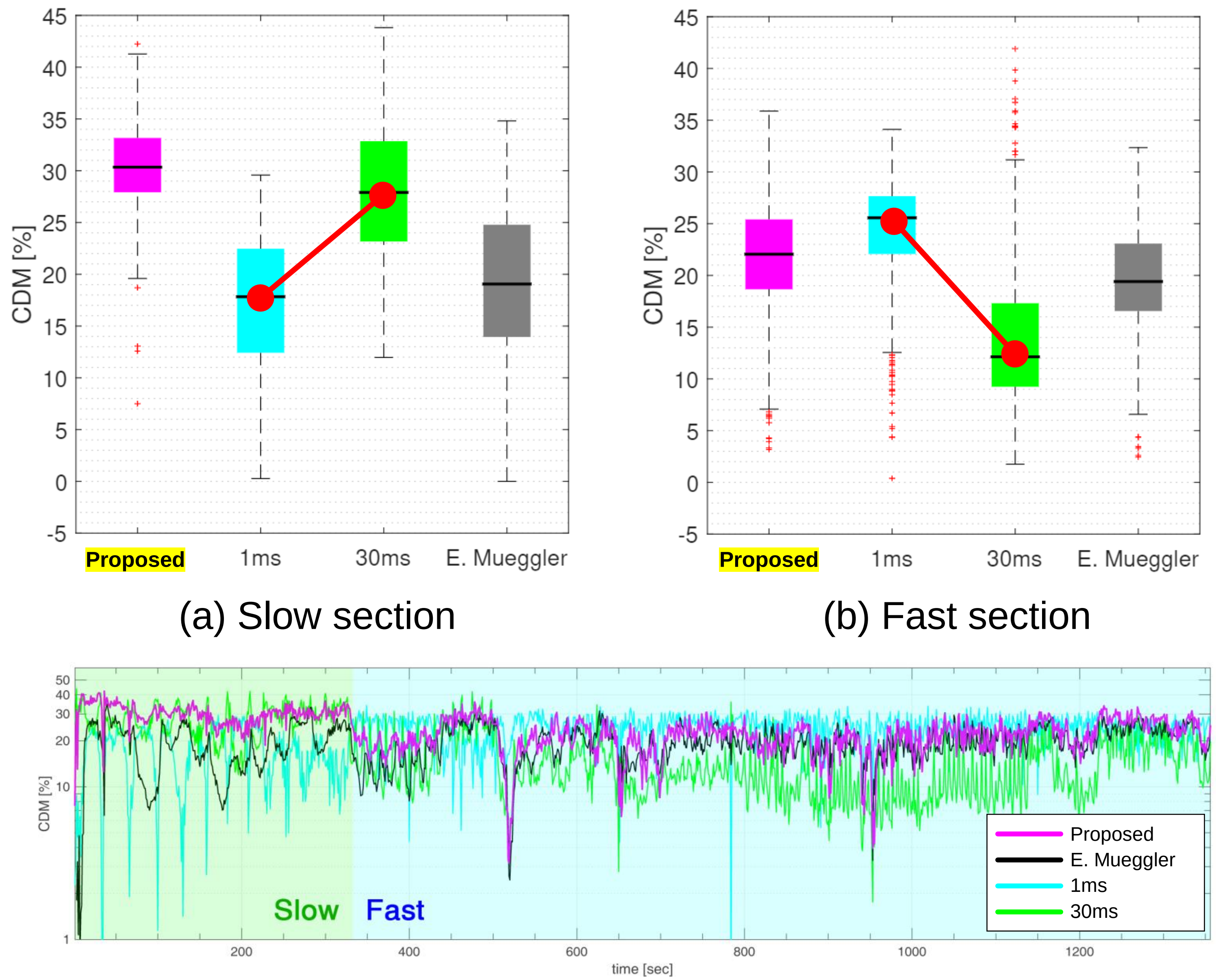
- Closest distance metric (CDM) [9] is defined as

$$CDM_\eta(f,g) = \left(1 - \frac{\mathcal{C}(\mathcal{M}_{cd}(f,g))}{|f \cup g|}\right) \times 100[\%],$$

where η is the pixel radius to find matching edge pixels between two binary images, f and g.

- $\mathcal{C}(\mathcal{M}_{cd}(f,g))$ is the distance cost of a pair matched by the closest-distance.

- $|f \cup g|$ is the number of edge pixels belonging to f or g.



(a) Slow section

(b) Fast section

(c) Whole sequence

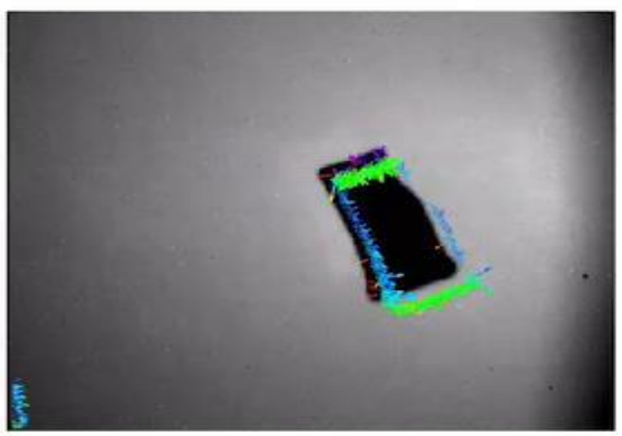
Figure 14. Performance analysis for edge detection. The camera moves slowly in the beginning and quickly in the latter part of the sequence.

[9] M. S. Prieto and A. R. Allen, “A similarity metric for edge images,” TPAMI, 2003.

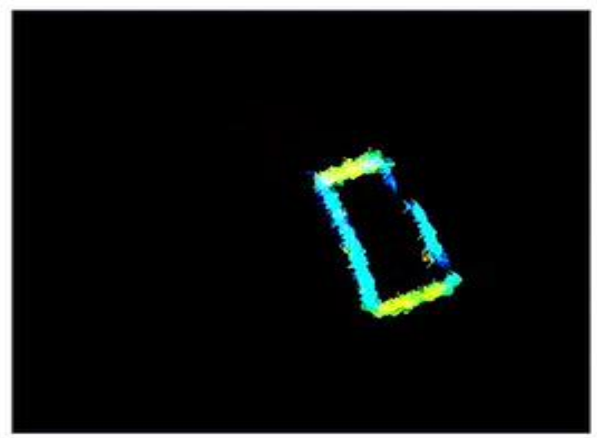
Summary

- We proposed an intra-pixel-area event to improve the performance of RANSAC so that visual flow, lifetime, and edge map can be precisely estimated.

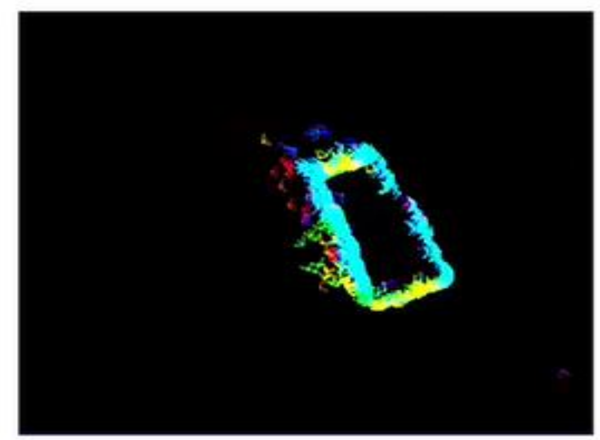
- However, in the natural scene, visual flow suffers from the aperture problem and derives the gradient vector of a straight line.



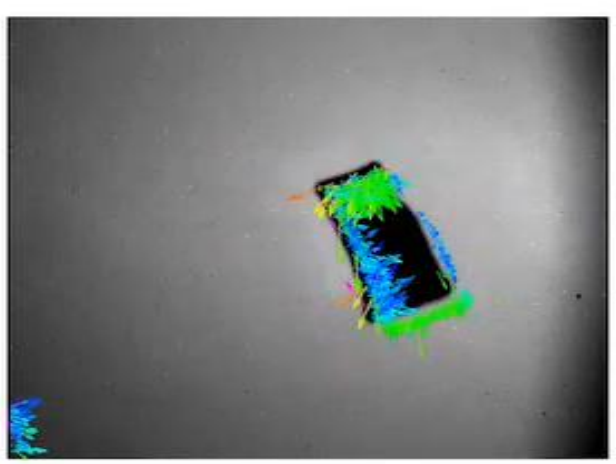
(a) Proposed



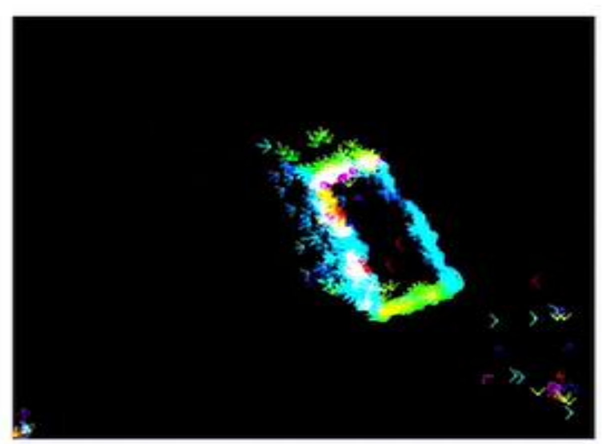
(b) Direction-Selective [7]



(c) Lucas-Kanade [8]



(d) E. Mueggler [6]



(e) Local Plane [5]

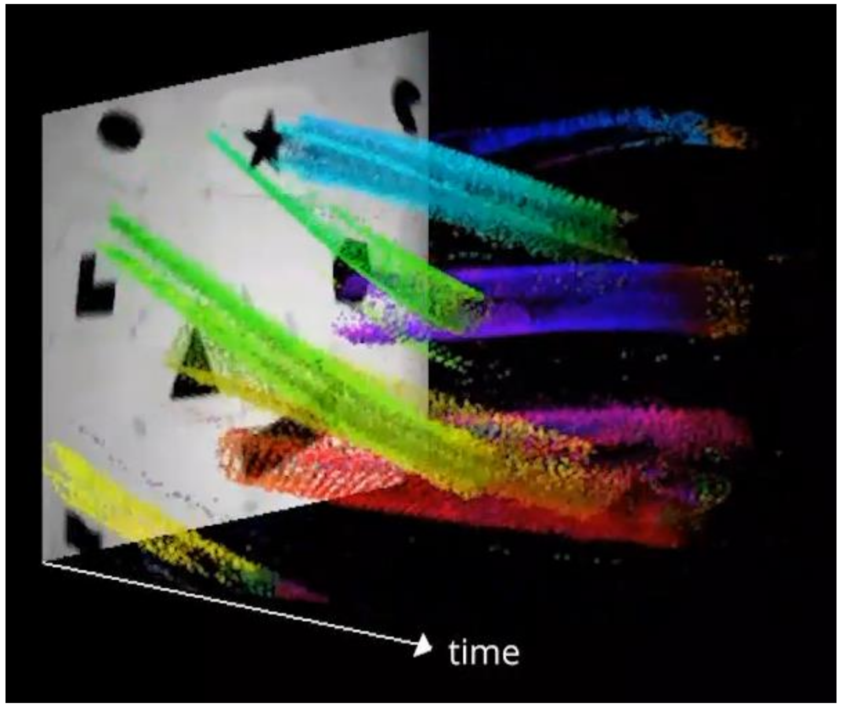
Video 9. Result of visual flow estimation on natural scene. The direction of flow vector are represented based on the color wheel.

Chapter 4

Objective and Contributions

Objective

- Develop an algorithm that estimates asynchronous optical flow stream with low latency and robustness to various scenes



Video 10. Asynchronous event (former) and optical flow (latter) stream.

Contributions

- Enhancement of the patch-matching-based optical flow algorithm
  : our local time slice produces asynchronous optical flow with very low latency
- Angular velocity estimation using above asynchronous optical flow stream
  : updates with high accuracy, low latency and robustness to a various scene

Asynchronous Optical Flow

Adaptive time-slice block-matching optical flow (ABMOF) [10]

- Fetch each incoming event
- Construct time slice (SAE) at certain condition
- Estimate optical flow vector by finding the best matching block in the both of the previous time slices

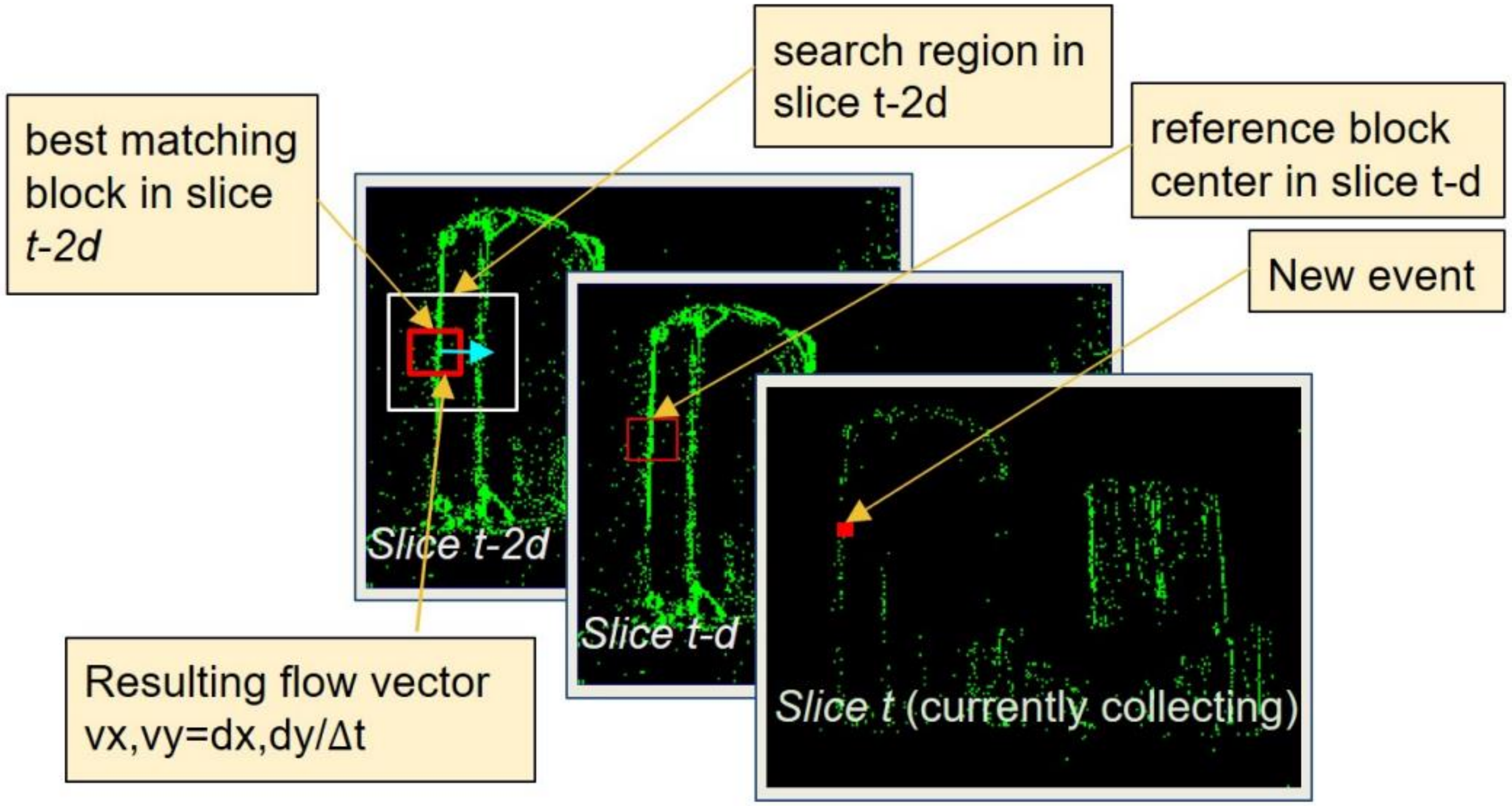


Figure 15. Time-slice block-matching method [14]

[10] M. Liu and T. Delbruck, “Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors,” BMVC, 2018.

Asynchronous Optical Flow

- Adaptive time-slice block-matching optical flow (ABMOF) vs. Proposed:
  1. global time slice over whole image frame vs. local time slice over each patch
  2. share dt vs. compute individual $\boldsymbol{dt_i}$

- In Figure 16, for visual simplicity, we illustrate the case for patch size $w = 3$ in the 3D image and its vertical direction representing the queue capacity and bright values are recent events.

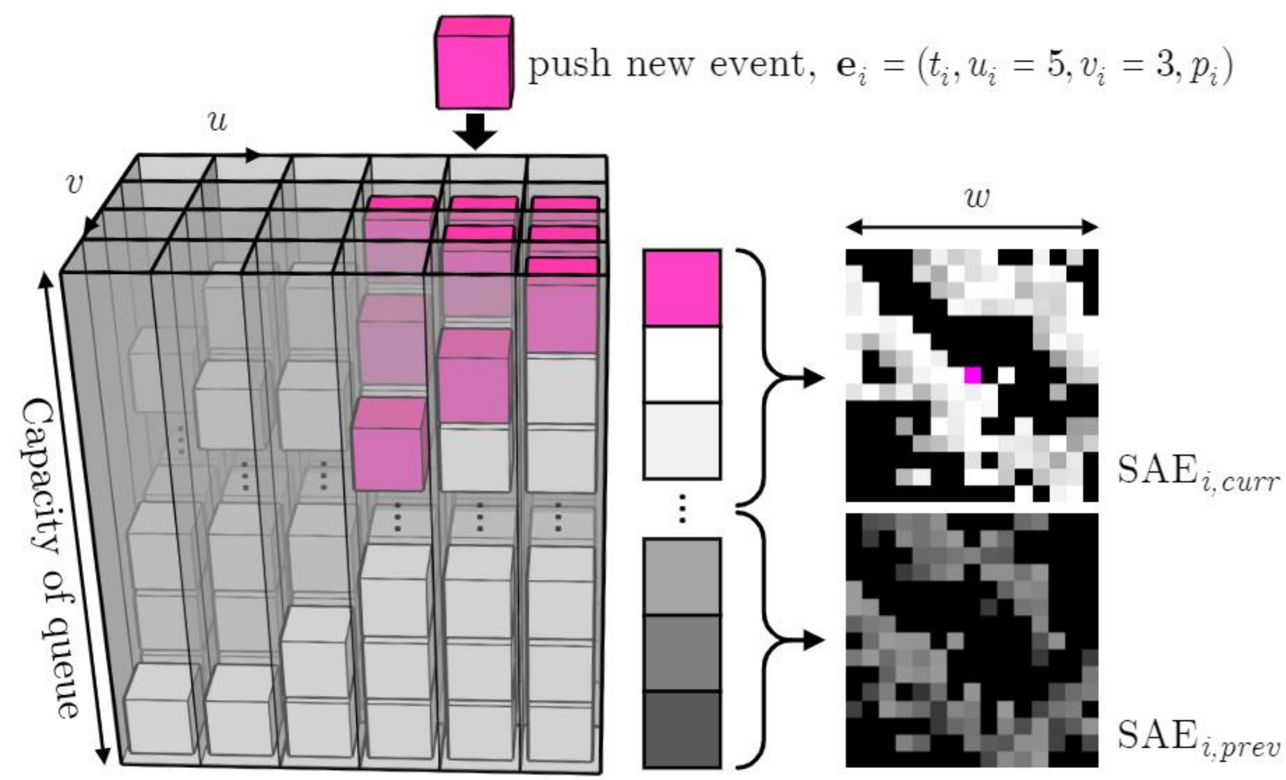
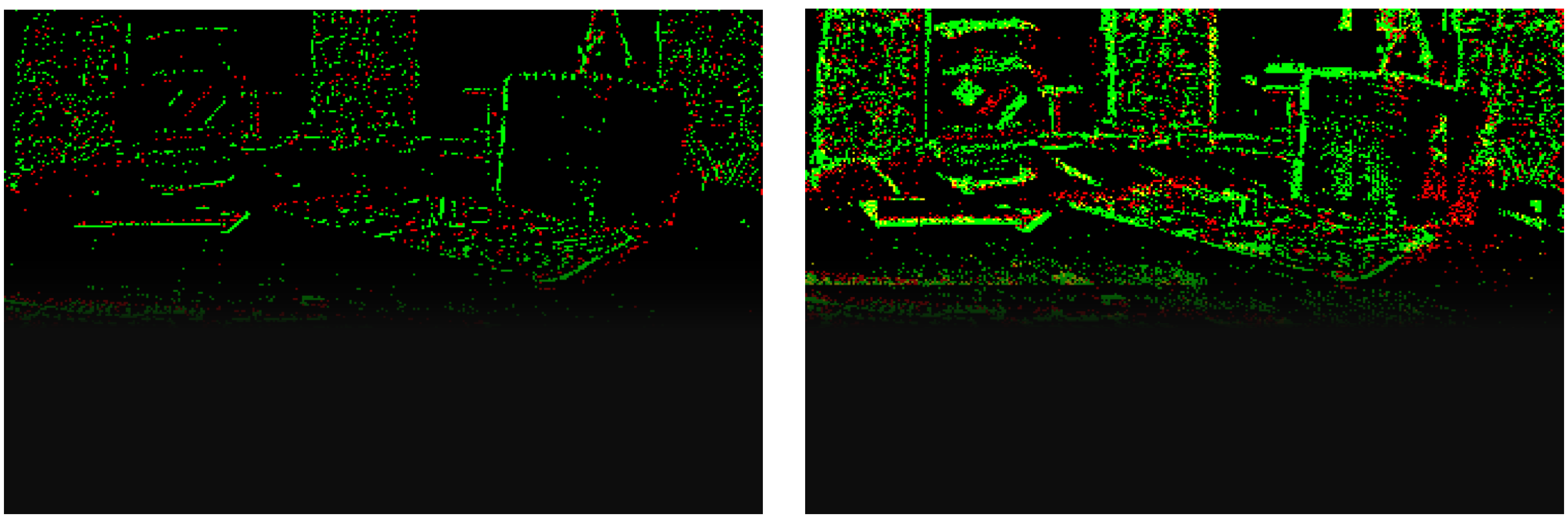


Figure 16. Description of the local time slice. Each bin denotes the queue of a pixel, and a new event (magenta cube) is pushed into adjacent bins.

Asynchronous Optical Flow

- Since ABMOF constructs a global time slice, it does not consider the texture distribution.
- Thus, erroneous optical flow may occur in areas with low texture.
- Our local time slice can extract the detail of a scene regardless of the local texture level so that the block-matching method successfully estimates an offset, i.e., optical flow.



(a) ABMOF (b) Proposed

Figure 17. The snapshot of time slice of (a) ABMOF and (b) ours. Since ours does not construct a full-sized time slice, local time slices of pixels is drawn overlaid on a frame for visualization purposes only.

Angular Velocity Estimation

- We compute angular velocity analytically from a bunch of optical flows.
- For an optical flow $(\mathrm{p}_k, \mathrm{v}_k, dt_k) = (x_k, y_k, u_k, v_k, dt_k)$, the below equation is satisfied:

$$\mathbf{v}_k = \begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} f_x & 0 \\ 0 & f_y \end{pmatrix} \begin{pmatrix} x_k y_k & -1 - x_k^2 & y_k \\ 1 + y_k^2 & -x_k y_k & -x_k \end{pmatrix} dt_k \omega = A_k dt_k \omega$$

- Then, for a total of n optical flows, angular velocity can be solved by least-square method

$$\omega = (A_{1:n}^T A_{1:n})^{-1} A_{1:n}^T T_{1:n}^{-1} \mathbf{v}_{1:n},$$

where

$$A_{1:n} = [A_1^T, \cdots, A_n^T]^T \in \mathbb{R}^{2n\times 3}, \ \mathbf{v}_{1:n} = [\mathbf{v}_1^T, \cdots, \mathbf{v}_n^T]^T \in \mathbb{R}^{2n\times 1}$$

$$T_{1:n} \in \mathbb{R}^{2n\times 2n}, \quad T_{2i-1,2i-1} = T_{2i,2i} = dt_i, \forall i = 1, \ldots, n$$

- When the variance becomes less than the threshold, angular velocity is computed.

$$\begin{aligned} \mathrm{Var}(\omega) &= (A^T A)^{-1} A^T T^{-1} \mathrm{Var}(v) T^{-1} A (A^T A)^{-1} \\ &= \sigma^2 (A^T A)^{-1} A^T T^{-2} A (A^T A)^{-1} \end{aligned}$$

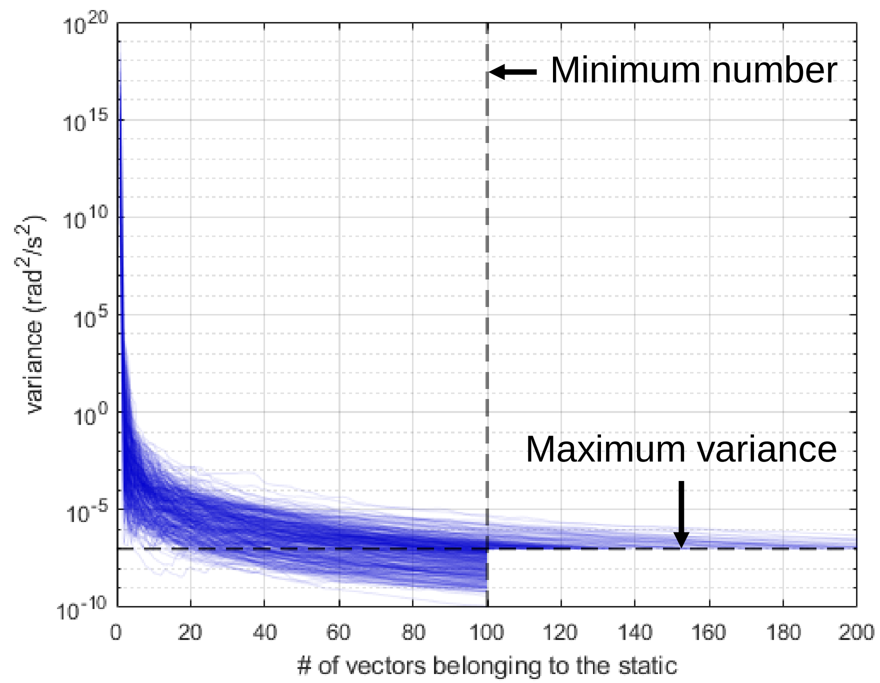


Figure 18. Variance versus iterations

Evaluation Results

- Evaluation of asynchronous optical flow on GT frame-based flowmap [11]

  (i) Since events are generated at the edge of a scene at which the true flowmap has discontinuities,

  (ii) Since our optical flows are generated asynchronously whereas the true flowmap is synchronized with frame rate,

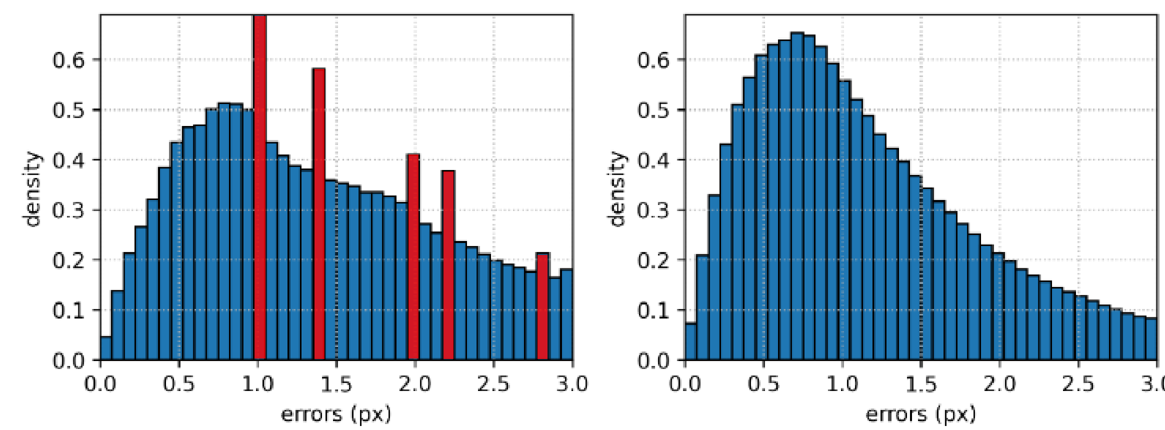
- Thus, we compensate optical flow vectors

$$\hat{\mathbf{v}}_i = \mathbf{v}_i \times \frac{t_{gt,k} - t_{gt,k-1}}{dt_i},$$

$$\hat{\mathbf{p}}_i = \mathbf{p}_i + \mathbf{v}_i \times \frac{t_{gt,k} - t_i}{dt_i},$$

$$\text{s.t. } t_{gt,k-1} < t_i < t_{gt,k}, \ \forall i,$$

where $t_{gt,k}$ is the timestamp of the k-th flowmap.



(a) w/o compensation (b) w/ compensation

Figure 19. Histogram of end-point errors of optical flow vectors with or without vector compensation.

Table 1. Evaluations of optical flow on GT flowmap

Sequence	Average end-point error (px)				
	Ours	ABMOF	Zhu [12]	EV-FN [13]	Spike-FN [14]
indoor1	1.09	1.28	0.58	1.03	0.84
indoor2	1.76	1.98	1.02	1.72	1.28
indoor3	1.54	1.74	0.87	1.53	1.11
outdoor1	2.43	2.88	0.32	0.49	0.49

[11] Zhu, A. Z., et al., “The Multi Vehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception.” RA-L, 2018.
[12] A. Z. Zhu, et al., “Unsupervised event-based learning of optical flow, depth, and egomotion,” CVPR, 2019.
[13] A. Z. Zhu, et al., “Ev-flownet: Self-supervised optical flow estimation for event-based cameras,” arXiv, 2018.
[14] C. Lee, et al., “Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks,” ECCV, 2020.

Evaluation Results

- Although EV-Flow produces a dense flowmap, we mask its result with the existence of events.
- ABMOF and EV-Flow yield erroneous optical flows at a distance where events are not triggered enough.

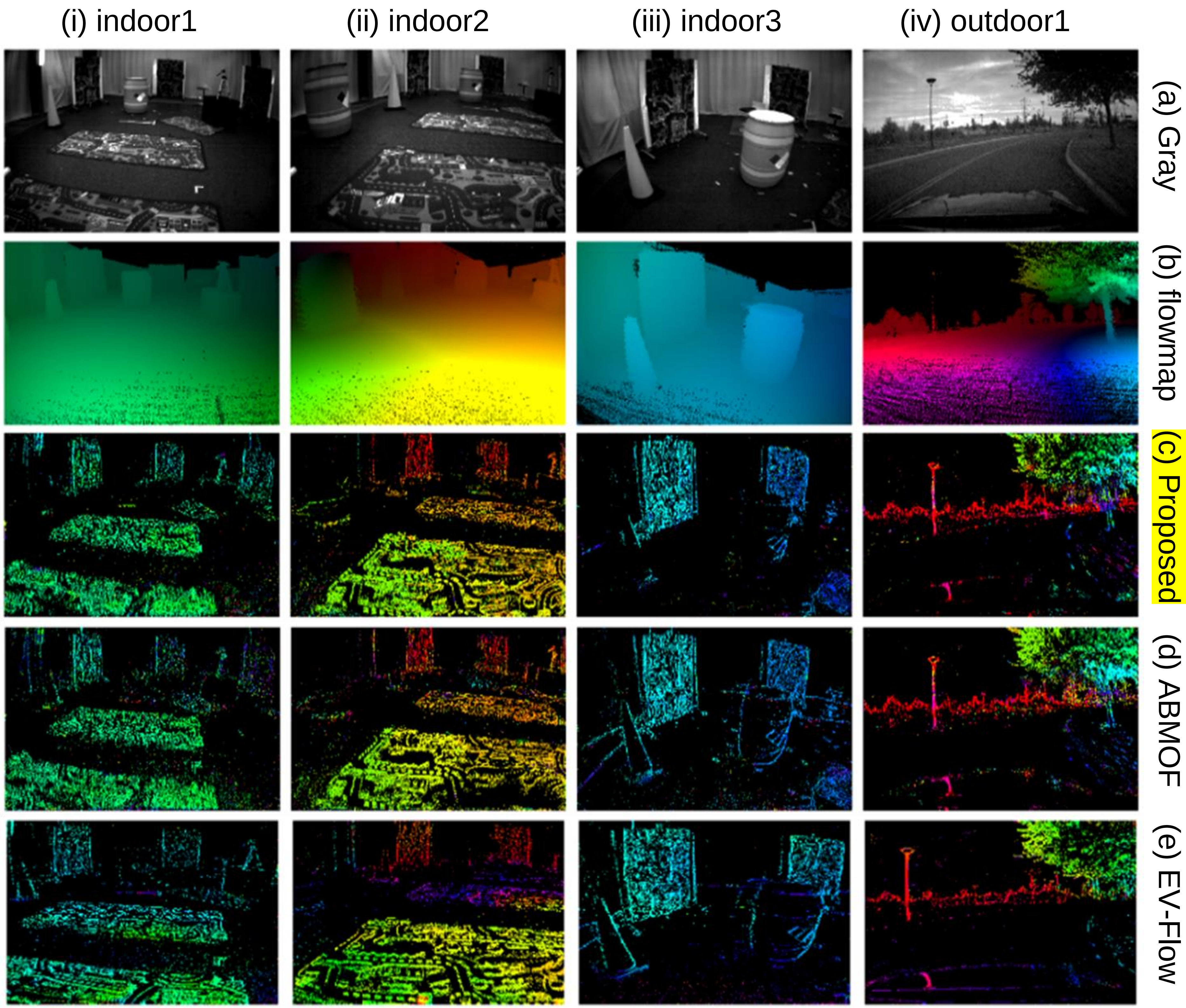


Figure 20. Qualitative results of optical flow on MVSEC sequences.

Evaluation Results

- Although EV-Flow produces a dense flowmap, we mask its result with the existence of events.
- ABMOF and EV-Flow yield erroneous optical flows at a distance where events are not triggered enough.

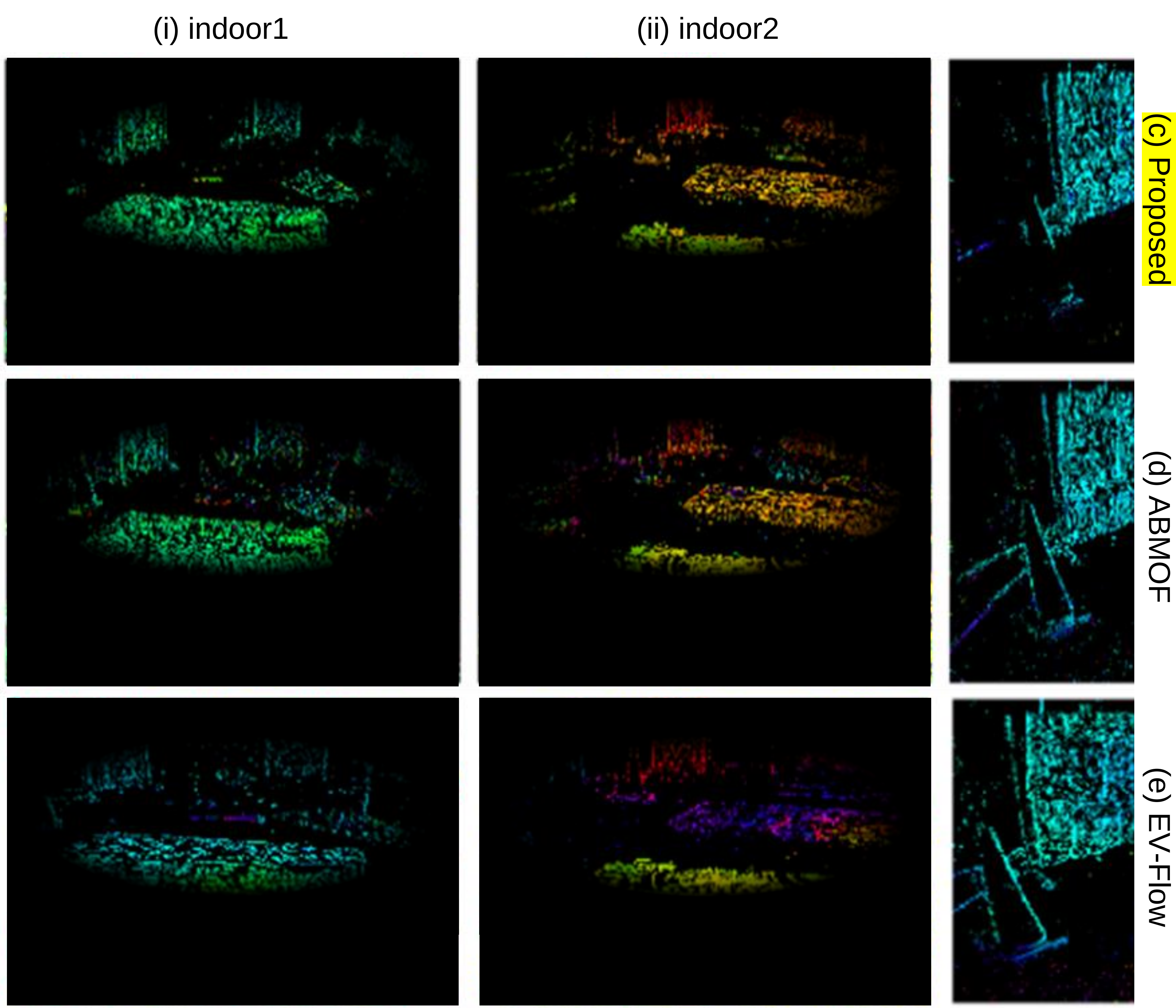
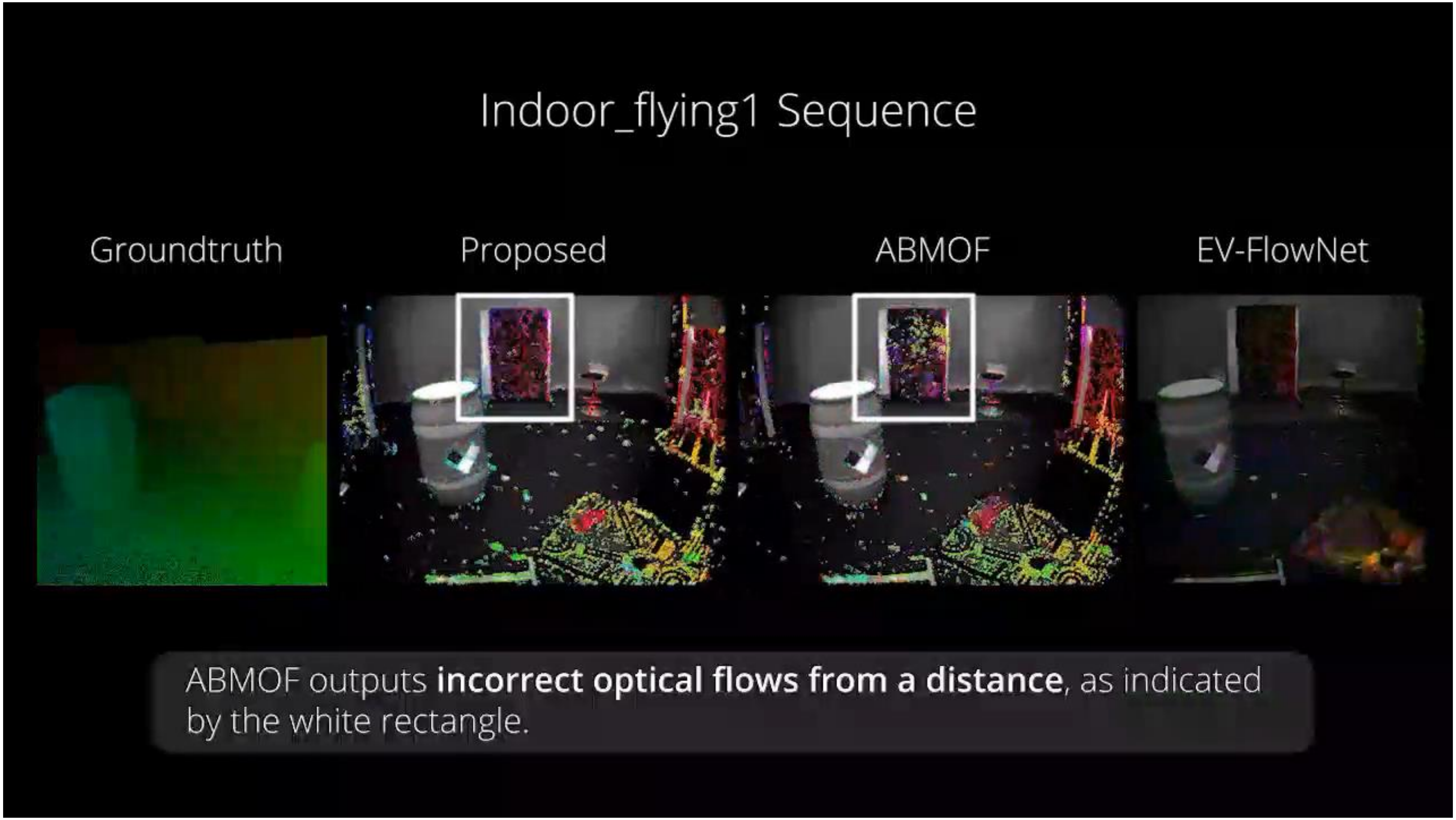


Figure 20. Qualitative results of optical flow on MVSEC sequences.

Evaluation Results



Video 11. Qualitative results of optical flow on MVSEC sequences [11].

Evaluation Results

- Latency is computed by minimizing the below:

$$\sum_i w_i L_\delta \left(A\hat{\mathbf{x}}_{est}(\tau_i + \tau_d) + \mathbf{b} - \hat{\mathbf{x}}_{gt}(\tau_i) \right),$$

  where τ_d is a latency, A is a diagonal coefficient, $\mathbf{b}$ is a bias coefficient, and $L_\delta(\cdot)$ is the Huber loss function.

- The number in parentheses indicates the accuracy of zero-latency angular velocity that is corrected by the above equation.

Table 2. Evaluations for latency and accuracy of optical flow. Best results are in green.

Sequence		Average latency (ms)			Average accuracy[1] (rad/s)		
		Ours	ABMOF	EV-FN	Ours	ABMOF	EV-FN
shapes	low	**14.40**	31.86	26.95	**0.110** (0.108)	0.147 (0.137)	0.269 (0.264)
	mid	**5.15**	9.76	20.56	**0.228** (0.195)	0.350 (0.256)	1.545 (1.524)
	high	**2.93**	7.09	20.48	**0.350** (0.310)	0.691 (0.563)	3.082 (3.039)
	whole	**3.04**	7.31	20.45	**0.296** (0.261)	0.548 (0.442)	2.350 (2.311)
boxes	low	**7.51**	13.12	19.55	**0.138** (0.134)	0.142 (0.136)	0.156 (0.146)
	mid	**1.86**	3.97	20.91	**0.297** (0.293)	0.361 (0.352)	1.199 (1.162)
	high	**0.59**	2.35	22.57	**0.355** (0.354)	0.380 (0.361)	2.666 (2.596)
	whole	**0.58**	2.27	21.74	**0.328** (0.327)	0.362 (0.346)	1.955 (1.903)
poster	low	**8.21**	10.39	16.01	0.238 (0.238)	**0.186** (0.185)	0.202 (0.190)
	mid	**2.14**	3.24	20.74	**0.303** (0.302)	0.347 (0.341)	1.291 (1.267)
	high	**0.52**	1.93	19.81	**0.359** (0.357)	0.409 (0.389)	2.906 (2.835)
	whole	**0.49**	1.87	19.91	**0.328** (0.327)	0.371 (0.356)	2.009 (1.960)
dynamic	low	**8.15**	8.41	26.02	**0.231** (0.229)	0.232 (0.227)	0.405 (0.391)
	mid	**4.35**	6.46	22.40	**0.242** (0.239)	0.272 (0.267)	0.539 (0.521)
	high	**1.80**	3.23	21.83	**0.289** (0.286)	0.341 (0.331)	1.525 (1.493)
	whole	**1.71**	3.19	21.97	**0.264** (0.261)	0.299 (0.291)	1.004 (0.980)

Evaluation Results

- In evaluation of angular velocity, eSNN [15] and CM [16] are compared with the proposed algorithm.
- Ours estimates accurate angular velocity with low latency and robustness to various scenes, whereas the performance of CM highly depends on the texture of the scene.

Table 3. Evaluations of angular velocity. Best results are in green.

Sequence		Average latency (ms)			Average accuracy (rad/s)		
		Ours	eSNN	CM	Ours	eSNN	CM
shapes	low	**14.40**	31.54	63.42	**0.110**	0.463	0.203
	mid	**5.15**	35.59	20.05	**0.228**	1.291	0.678
	high	**2.93**	35.93	14.57	**0.350**	2.578	1.154
	whole	**3.04**	35.99	14.42	**0.296**	1.972	0.987
boxes	low	7.51	45.49	**5.51**	**0.138**	0.447	0.145
	mid	1.86	39.90	**0.39**	**0.297**	1.333	0.380
	high	0.59	38.00	**0.26**	**0.355**	2.451	0.396
	whole	0.58	38.31	**0.18**	**0.328**	1.856	0.378
poster	low	8.21	39.74	**7.37**	**0.238**	0.458	0.250
	mid	2.14	40.31	**1.60**	**0.303**	1.163	0.463
	high	0.52	39.29	**0**	**0.359**	2.901	0.461
	whole	0.49	39.40	**0**	**0.328**	1.998	0.443
dynamic	low	**8.15**	46.36	9.68	0.231	0.594	**0.149**
	mid	**4.35**	43.67	5.63	0.242	0.628	**0.160**
	high	**1.80**	41.69	2.44	0.289	1.314	**0.183**
	whole	**1.71**	41.67	2.32	0.264	0.939	**0.169**

[15] C. Lee, “Spike-FlowNet: Event-based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks,” arXiv, 2020.
[16] G. Gallego, et al., “A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation,” CVPR, 2018.

Summary

- We developed an algorithm that estimates accurate and low-latency optical flow.
- This approach takes advantage of event cameras such as low latency and high temporal resolution.
- The proposed optical flow and angular velocity estimation are exploited to develop robust angular velocity estimation in dynamic environments.

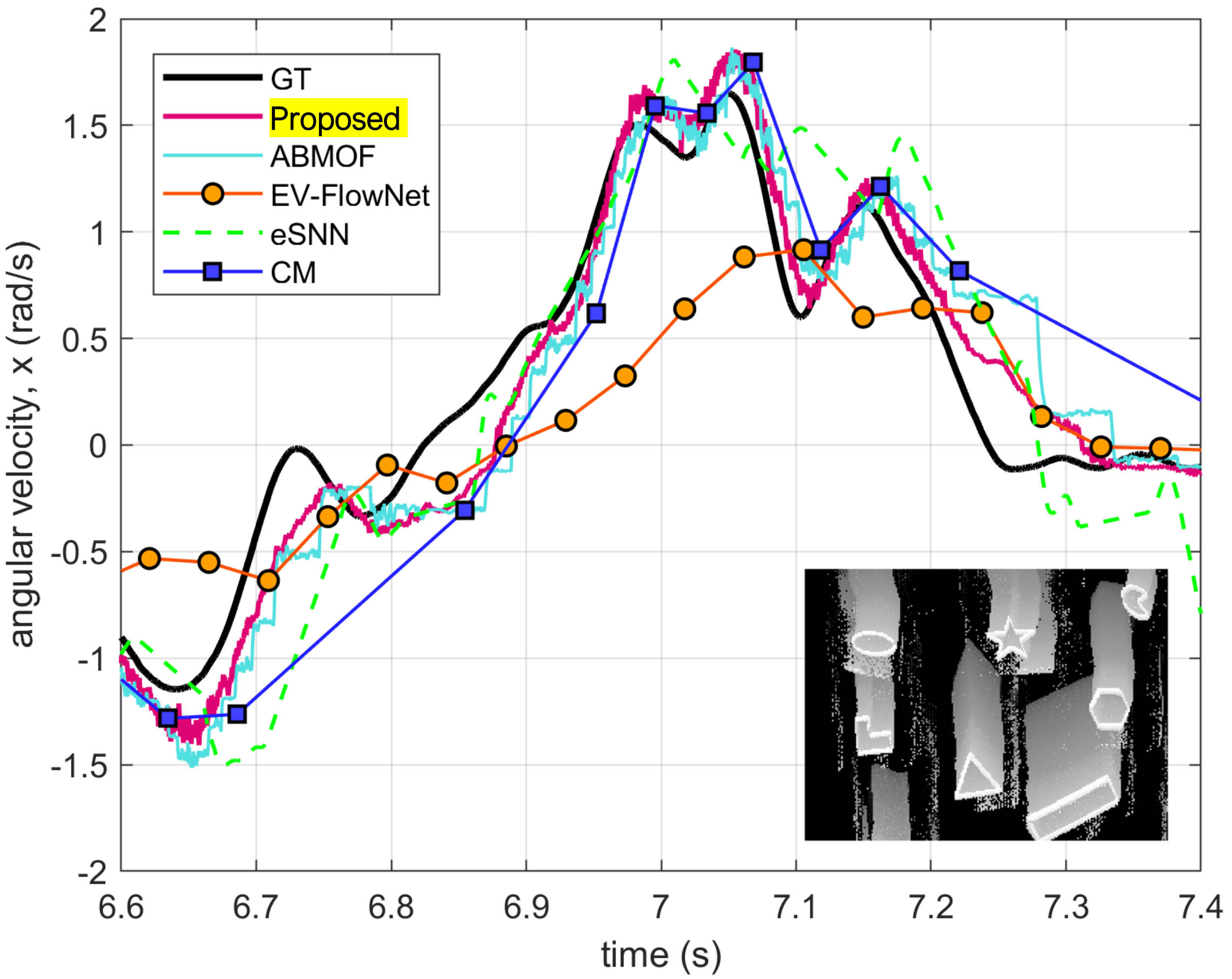


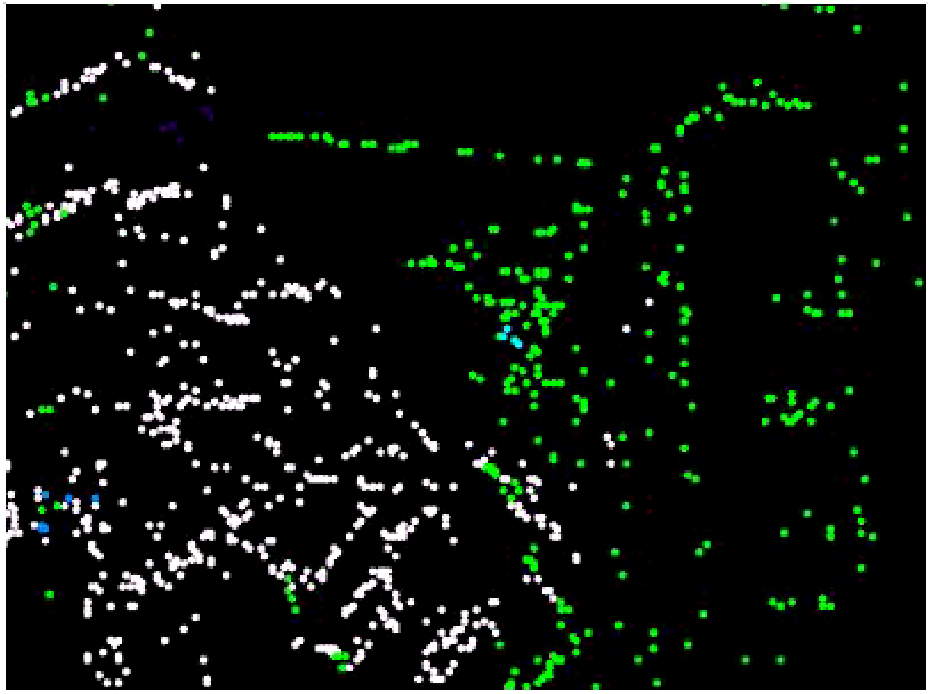
Figure 21. The result of angular velocity estimation. The bottom-right grayscale image denotes an image of stacked events showing the camera movement at a certain moment. Bright values are recent events.

Chapter 5

Objective and Contributions

Objective

- Develop an algorithm that robustly estimates angular velocity of the camera in dynamic environments where moving objects exist



Video 12. The class-separated optical flow

Contributions

- Segmentation of static background and dynamic foreground
  : our algorithm identifies a static background even if the camera stops

- Dual-mode motion model for event camera estimates the motion of static background
  : estimates the angular velocity of ego-motion robustly and accurately

Motion Segmentation

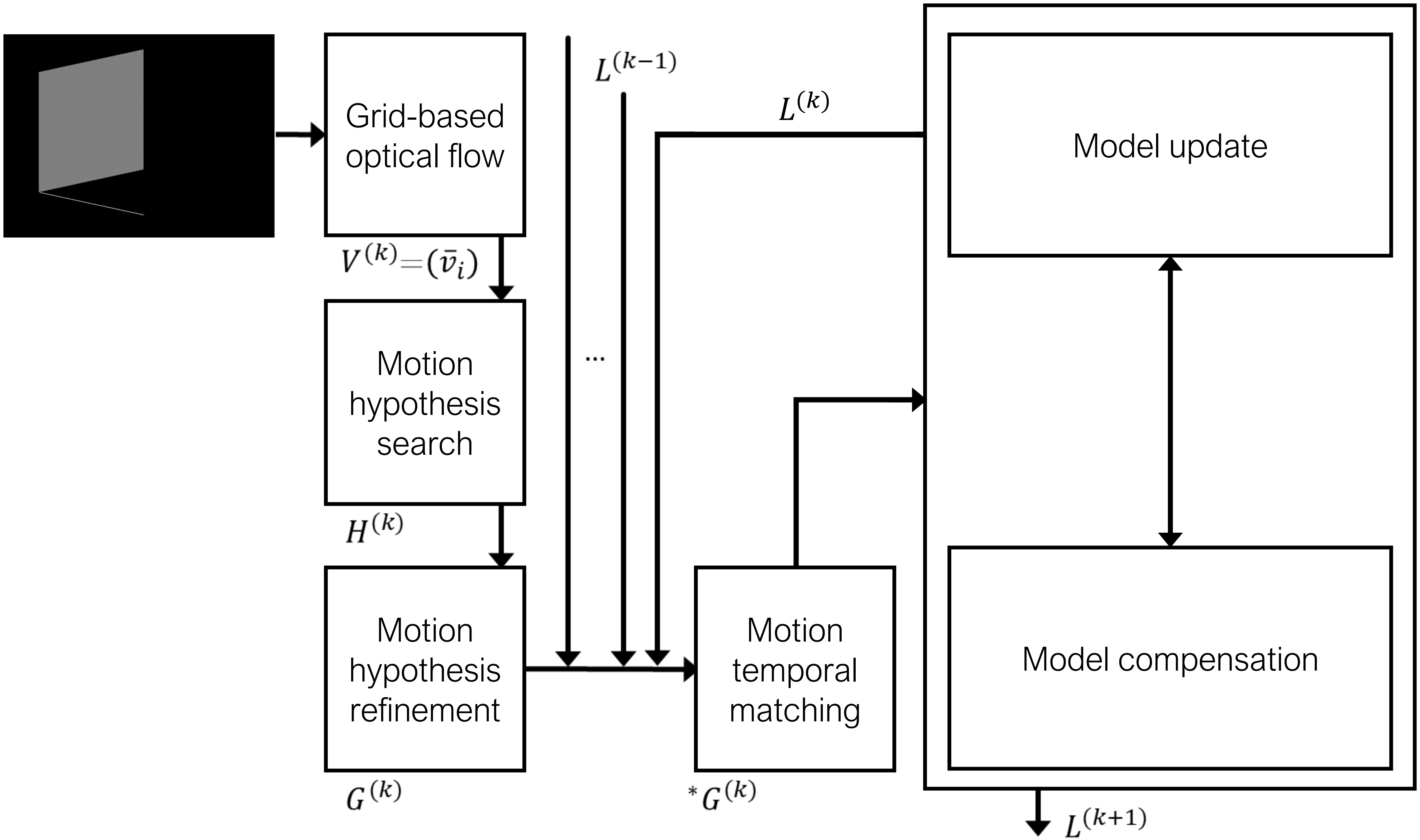


Figure 22. The pipeline description of the algorithm. A stream of optical flow is depicted based on a color wheel. $H^{(k)}$ and $G^{(k)}$ are the result of motion hypothesis search and refinement. Motion temporal matching yields $^*G^{(k)}$ from the label $L^{(k)}$ and the past.

Motion Segmentation

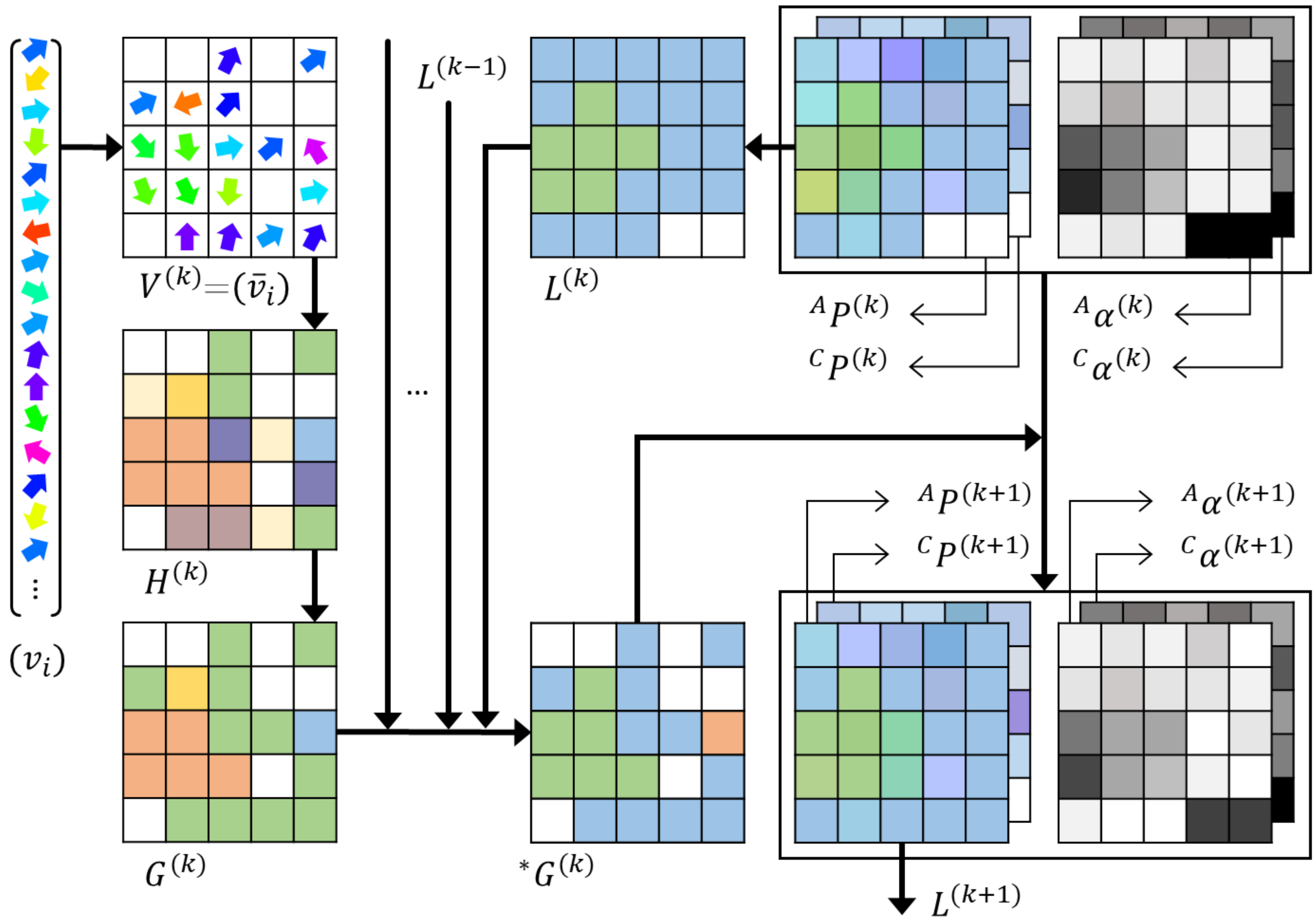


Figure 23. The pipeline description of the algorithm. Flowmap of grid-based optical flow $V^{(k)}$ is depicted based on a color wheel. $H^{(k)}$ and $G^{(k)}$ are the result of motion hypothesis search and refinement. Motion temporal matching yields ${}^*G^{(k)}$ from the label $L^{(k)}$ and the past. An IDs in $H^{(k)}$, $G^{(k)}$, ${}^*G^{(k)}$, and $L^{(k)}$ are indicated as eigen colors, whereas the probability P are represented by the mixture of eigen colors. In age map α, the brighter denotes the older.

Motion Segmentation

Grid-based optical flow

- To segment incoming spatial information instantly, a grid-based flow map is built.
- The representative optical flow of a cell is obtained from the medoid.

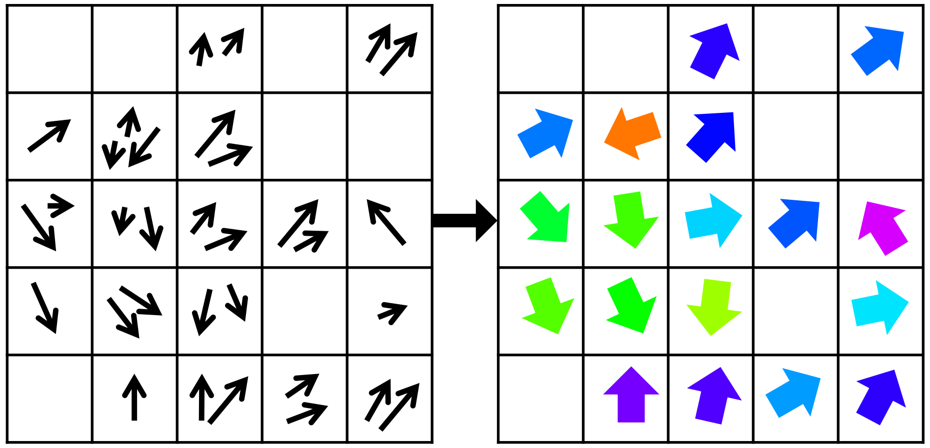


Figure 24. The construction of a grid-based optical flow map

Motion hypothesis search

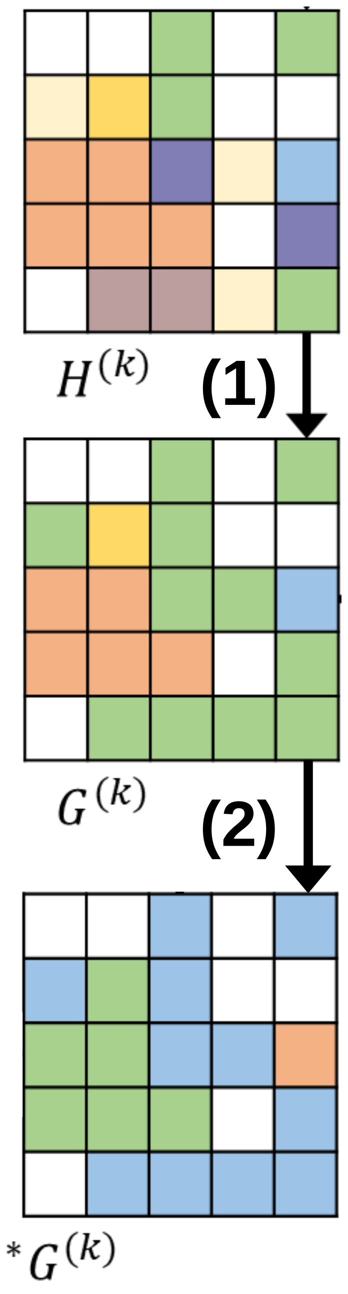
- We randomly select nearby cells with a probability proportional to the min-error, s_i:

$$
\begin{aligned}
s_i &= \min\left(\min_h e_{h,i}, 1\right),\ i = 1 \dots n, \\
E &= (e_{h,i}) \\
&= \begin{pmatrix} \mathrm{dist}(\boldsymbol{\omega}_1, \bar{v}_1) & \cdots & \mathrm{dist}(\boldsymbol{\omega}_1, \bar{v}_n) \\ \vdots & \ddots & \vdots \\ \mathrm{dist}(\boldsymbol{\omega}_{n_{hyp}}, \bar{v}_1) & \cdots & \mathrm{dist}(\boldsymbol{\omega}_{n_{hyp}}, \bar{v}_n) \end{pmatrix}
\end{aligned}
$$

Motion Segmentation

Motion hypothesis refinement and clustering (1)

- After searching n_{hyp} motion hypothesis, we repeat (E-step) finding the cell inliers and (M-step) estimating the their motion, until the supported number of each motion hypothesis remains unchanged.
- DBSCAN groups a number of motion hypotheses into fewer motions.



Motion temporal matching (2)

- Under the assumption that the distributions of the cell of a specific object are not changed unexpectedly in a short time.
- A matching coefficient between the i-th and the j-th motion at different timestamp *(k)* and *(l)*:

$$C_{ij}^{(k,l)} = \frac{\sum (K_\sigma * \delta(G^{(k)}, i)) \odot \delta(G^{(l)}, j)}{\sqrt{\sum \delta(G^{(k)}, i) \sum \delta(G^{(l)}, j)}}, \quad C_{ij}^{(k,l)} = \frac{1}{1 + ||\omega_i^{(k)} - \omega_j^{(l)}||_2^2},$$

where δ operator constructs a binary matrix with one if the element value of the first term is equal to the second term.

Dual-mode Motion Model Management

Motion model update

- $P_i(h)$ means a probability of cell i is associated to motion h, and its label denotes dominant motion:

$$L_i = \underset{1 \leq h \leq n_{hyp}}{\operatorname{argmax}} P_i(h)$$

- When the result of spatial segmentation and temporal matching $G_i^{(k)}$ is fetched,

Apparent model:

$${}^{A}P_i^{(k+1)} = \begin{cases} \frac{{}^{A}\tilde{\alpha}_i^{(k)}}{{}^{A}\tilde{\alpha}_i^{(k)}+1} {}^{A}\tilde{P}_i^{(k)} + \frac{vec(G_i^{(k)})}{{}^{A}\tilde{\alpha}_i^{(k)}+1}, & \text{if } G_i^{(k)} = {}^{A}L_i^{(k)}, \\ {}^{A}\tilde{P}_i^{(k)}, & \text{otherwise,} \end{cases}$$

Lead to the final label

$${}^{A}\alpha_i^{(k+1)} = \begin{cases} \min\left({}^{A}\tilde{\alpha}_i^{(k)} + 1, \alpha_{\max}\right), & \text{if } G_i^{(k)} = {}^{A}L_i^{(k)}, \\ {}^{A}\tilde{\alpha}_i^{(k)}, & \text{otherwise.} \end{cases}$$

Candidate model:

$${}^{C}P_i^{(k+1)} = \begin{cases} \frac{{}^{C}\tilde{\alpha}_i^{(k)}}{{}^{C}\tilde{\alpha}_i^{(k)}+1} {}^{C}\tilde{P}_i^{(k)} + \frac{vec(G_i^{(k)})}{{}^{C}\tilde{\alpha}_i^{(k)}+1}, & \text{if } G_i^{(k)} \neq {}^{A}L_i^{(k)}, \\ \mathbf{0}, & \text{else if } {}^{C}\alpha_i^{(k+1)} = 0 \text{ and } {}^{C}\alpha_i^{(k)} > 0, \\ {}^{C}\tilde{P}_i^{(k)}, & \text{otherwise,} \end{cases}$$

$${}^{C}\alpha_i^{(k+1)} = \begin{cases} \min\left({}^{C}\tilde{\alpha}_i^{(k)} + 1, \alpha_{\max}\right), & \text{if } G_i^{(k)} = {}^{C}L_i^{(k)}, \\ \max\left({}^{C}\tilde{\alpha}_i^{(k)} - \tau \cdot (t^{(k+1)} - t^{(k)}), 0\right), & \text{otherwise,} \end{cases}$$

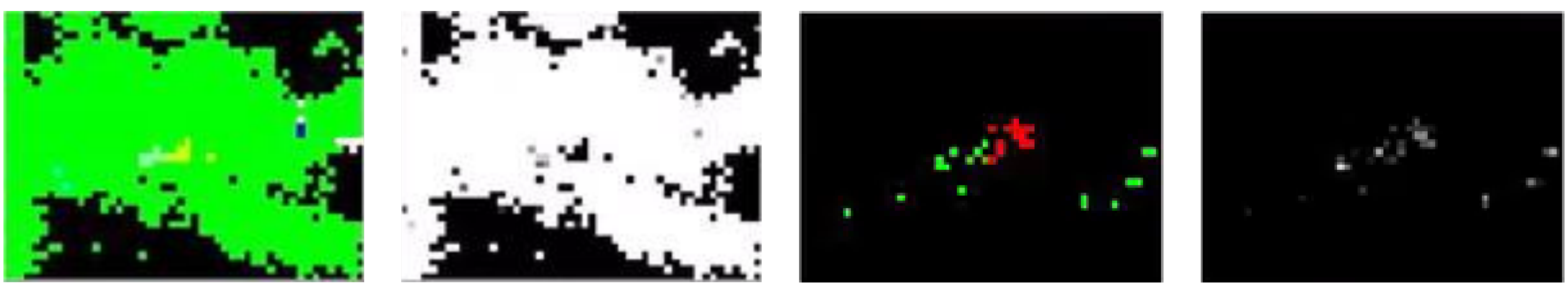
Dual-mode Motion Model Management

Motion model swap

- Model swap is designed for preventing the apparent model from updating erroneous motion or from missing new motion.
- When the age of candidate model is saturated or larger than that of apparent model:

$$
{}^{A}P_i^{(k+1)} = {}^{C}\tilde{P}_i^{(k+1)}, \qquad {}^{C}P_i^{(k+1)} = \mathbf{0} \in \mathbb{R}^{n_{hyp}},
$$

$$
{}^{A}\alpha_i^{(k+1)} = 0, \qquad {}^{C}\alpha_i^{(k+1)} = 0.
$$



(a) ${}^{A}P$ (b) ${}^{A}\alpha$ (c) ${}^{C}P$ (d) ${}^{C}\alpha$

Video 13. The internal parameter of apparent and candidate model when motion model swap happens.

Dual-mode Motion Model Management

Motion model compensation

- Since the apparent and candidate models operate on the camera coordinate, their parameters are need to be compensated through optical flow measurement.

- For each single optical flow vector, the probability vector and age of each model are compensated by 2D area-weighted interpolation:

$$\tilde{P}_i^{(k)} = \sum_{j \in \mathcal{S}_i^{(k)}} \omega_{ij} P_j^{(k-1)},$$

$$\tilde{\alpha}_i^{(k)} = \sum_{j \in \mathcal{S}_i^{(k)}} \omega_{ij} \alpha_j^{(k-1)}.$$

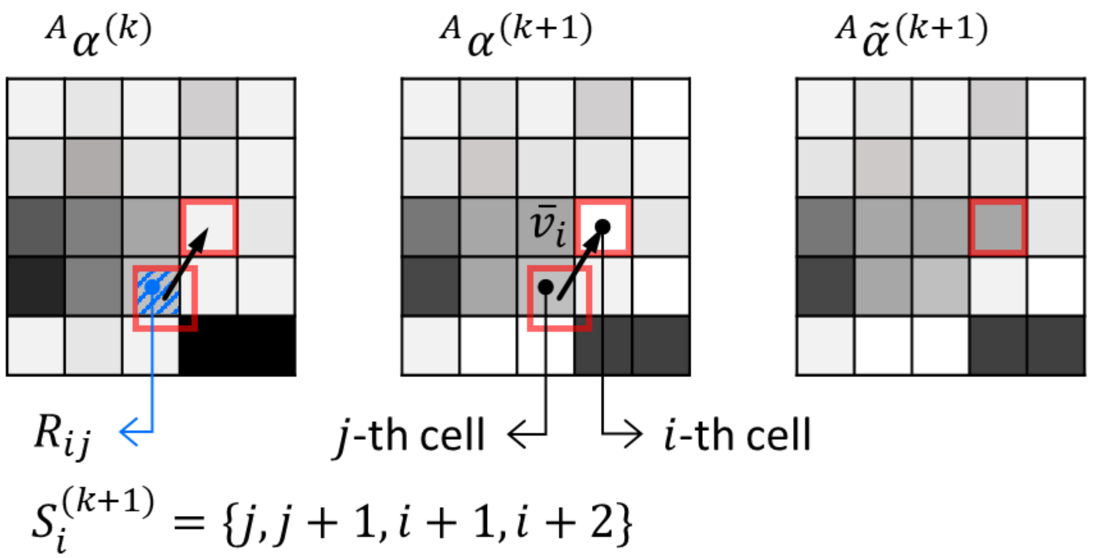
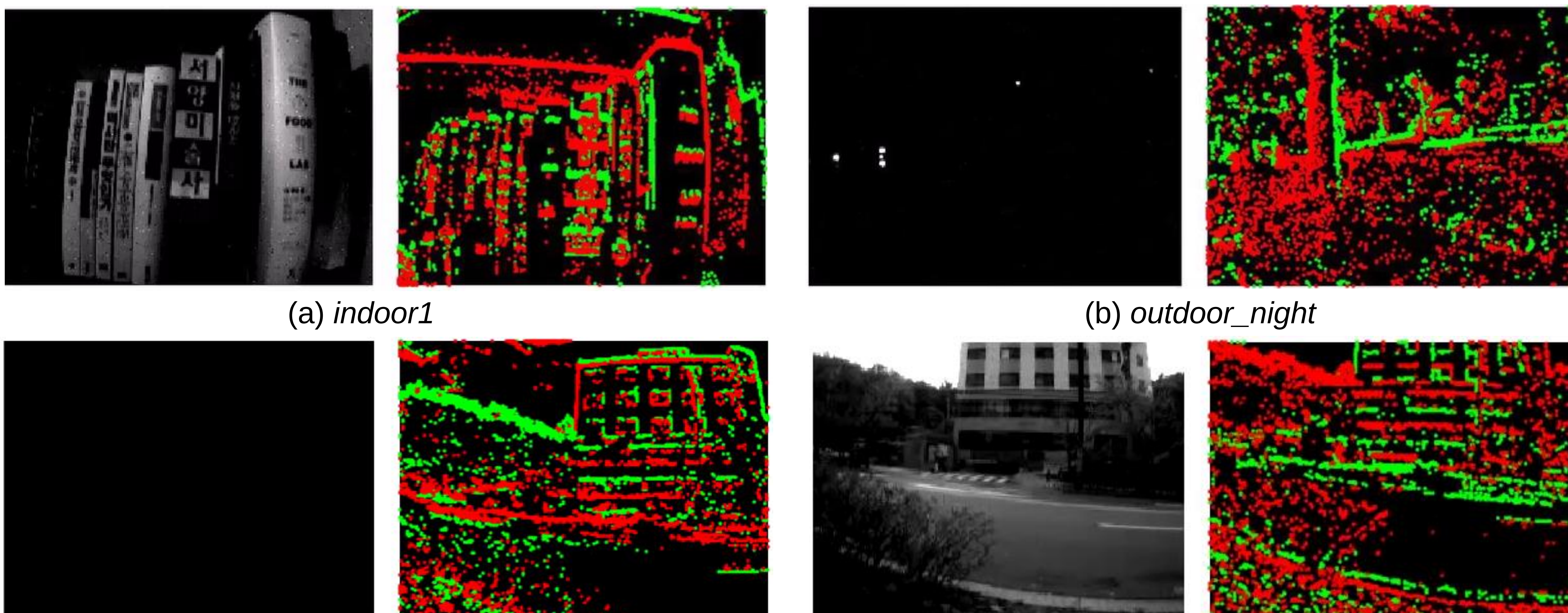


Figure 25. The description of how the parameters of cell are compensated. The average optical flow of i-th cell is $\bar{v}_i$, and then the parameters of i-th cell are compensated based on the parameters of cells that are elements of $S_i^{(k+1)}$.

Evaluation Results

Collected dataset

- Test sequences include indoor/outdoor, day/night environments.
- Test sequences include moving object such as human's interaction, vehicles.
- Published online at https://sangillee.com/_pages/larr-dvs-de-dataset/

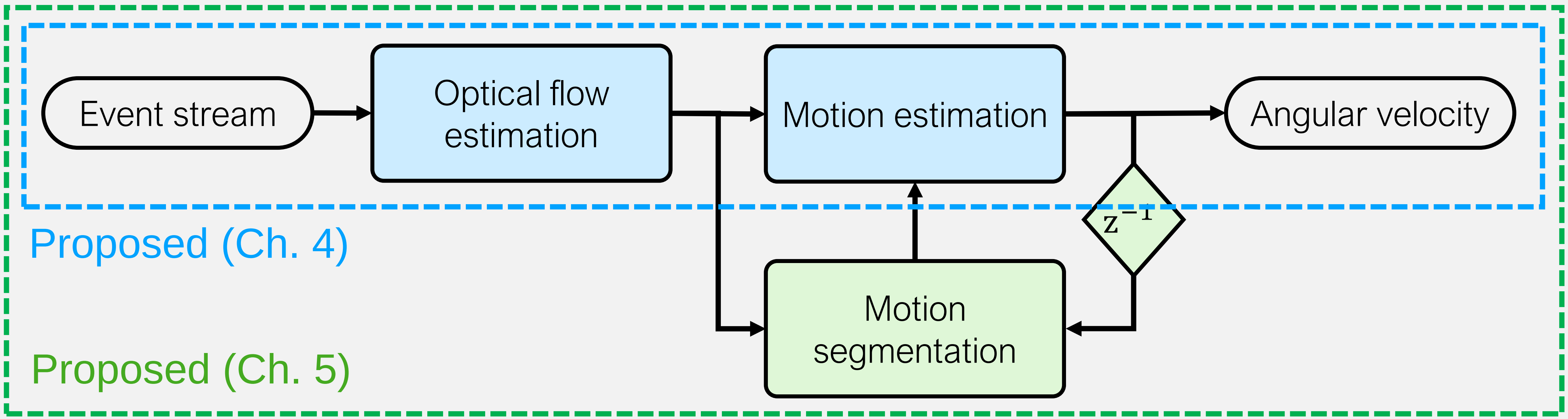


(a) *indoor1* (b) *outdoor_night* (c) *outdoor_day1* (d) *outdoor_day2*

Video 14. Frames and events of tested sequences. Positive and negative events are represented as green and red dots, respectively.

Evaluation Results

Compared algorithm



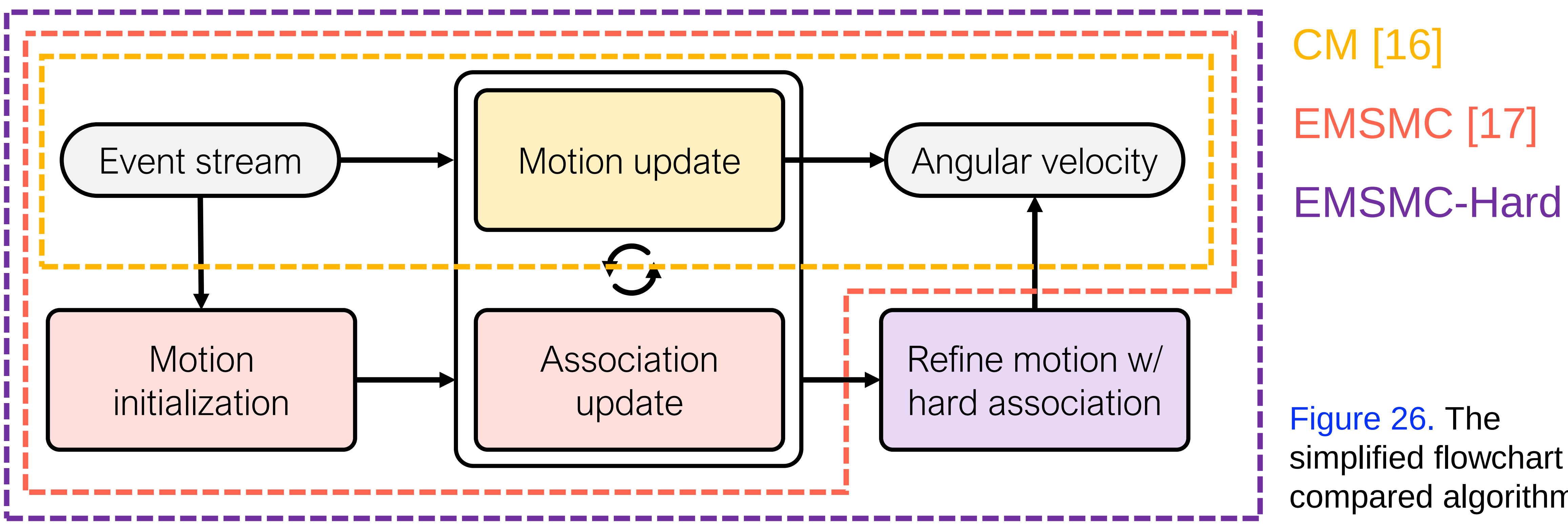


Figure 26. The simplified flowchart of compared algorithms.

Evaluation Results

- The model of ours, and IWE of both EMSMC and CM are depicted.
- EMSMC segments different motions, whereas IWE of CM is corrupted due to moving object.
- Ours also can detect moving objects.

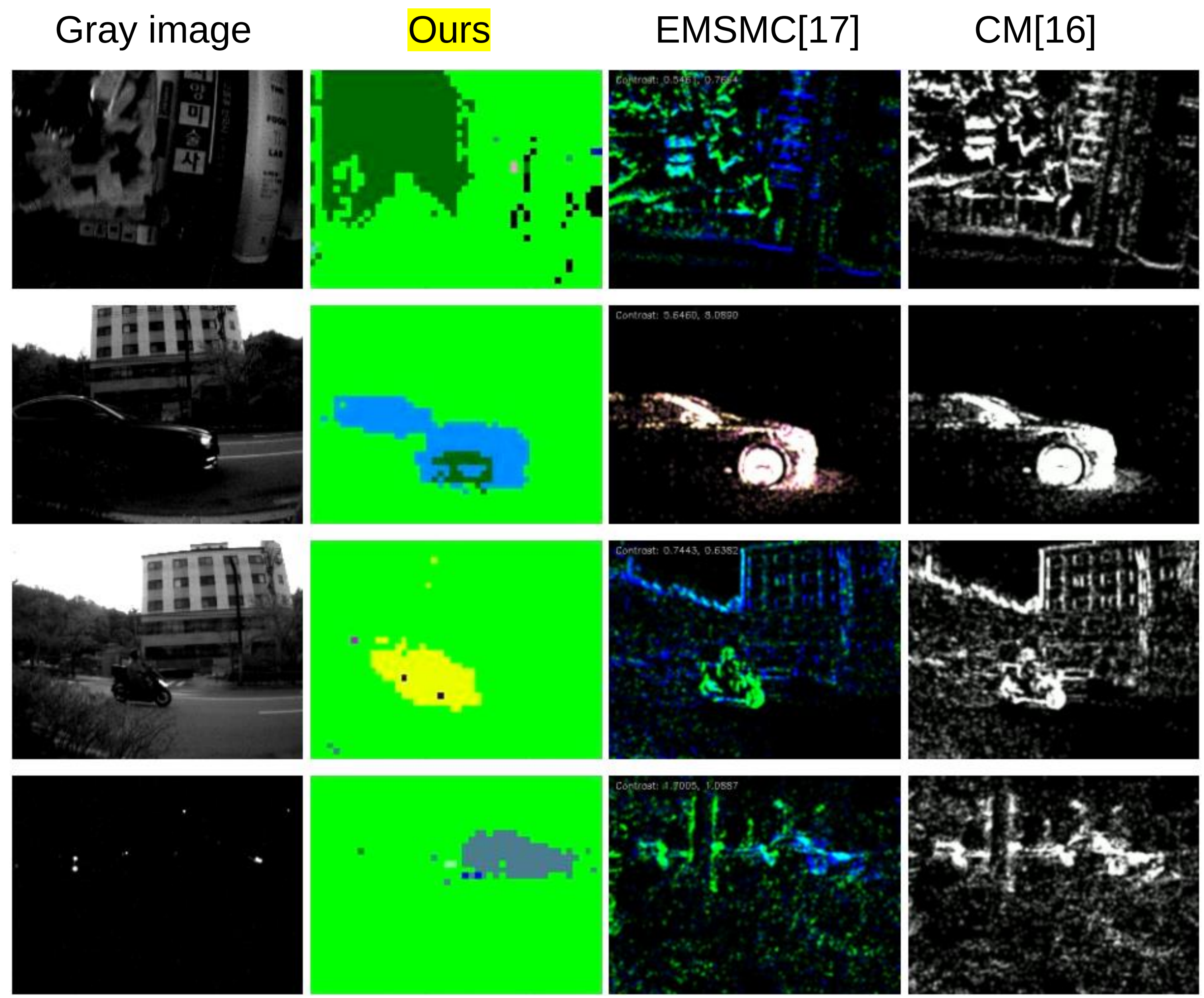
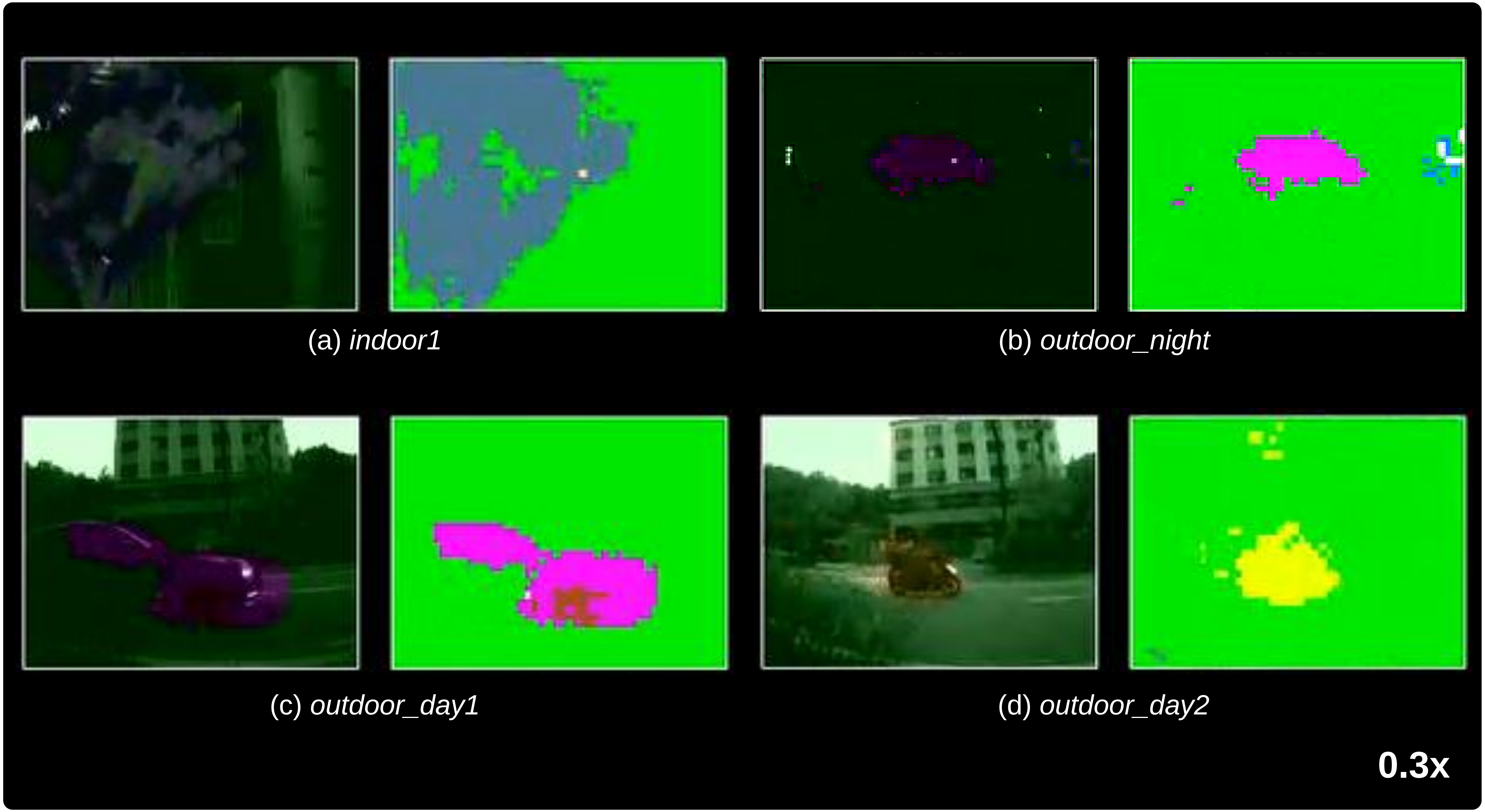


Figure 27. The internal parameter of compared algorithms.

[16] G. Gallego, et al., “A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation,” CVPR, 2018.
[17] T. Stoffregen, et al., "Event-based motion segmentation by motion compensation," ICCV, 2019.

Evaluation Results



Video 15. The internal parameter of our algorithm on the collected dataset.

Evaluation Results

- Table 4 shows the quantitative result of evaluations on the collected dataset.
- The latency of our algorithm is increased due to motion model update.
- Nevertheless, our algorithm is superior to other algorithms in terms of accuracy.
- The performance of EMSMC and CM deteriorate due to the existence of moving object.
- Although ours (Ch.4) does not consider moving objects, it estimates angular velocity accurately than other algorithms due to its robust estimator, RANSAC.

Table 4. Evaluations of angular velocity. Best results are in bold.

	Sequence	indoor	outdoor_day1	outdoor_day2	outdoor_night
Avg. latency (ms)	Ours	16.42	14.36	15.16	29.27
	Ours (Ch.4)	8.72	12.59	11.26	24.22
	EMSMC	**5.66**	10.22	11.23	25.86
	EMSMC-Hard	16.91	**4.19**	11.29	26.15
	CM	5.88	8.91	**9.35**	**18.78**
Avg. accuracy (rad/s)	Ours	**0.264**	**0.141**	**0.145**	0.160
	Ours (Ch.4)	0.537	0.409	0.153	**0.155**
	EMSMC	0.937	0.779	0.391	1.242
	EMSMC-Hard	1.045	0.537	0.570	0.753
	CM	0.874	0.539	0.542	0.294

Evaluation Results

- For visual clearness, we draw plots of angular velocity for x-axis only.
- The dark shaded region denotes the existence of moving object.
- Our algorithm robustly and accurately estimates the angular velocity.

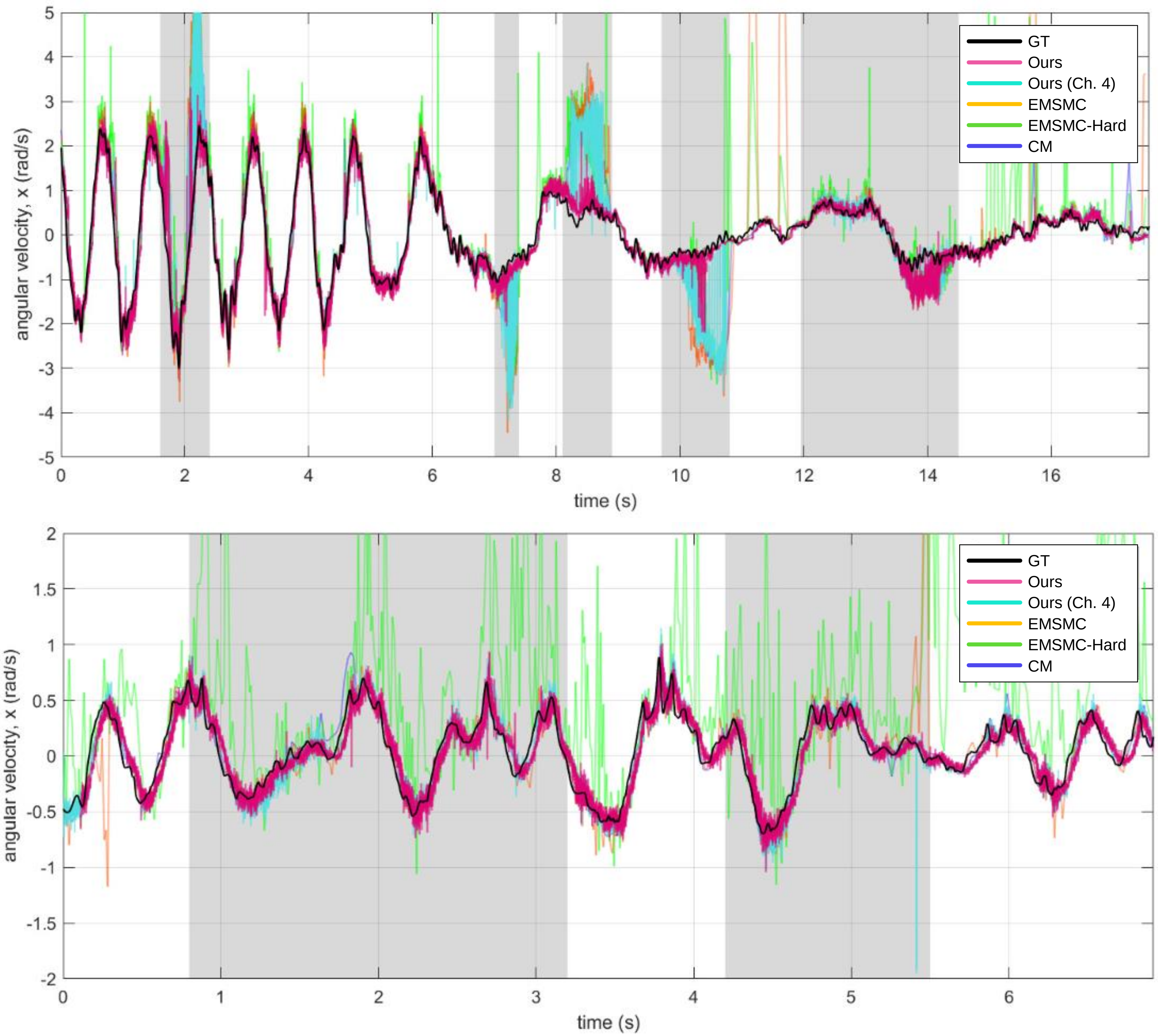
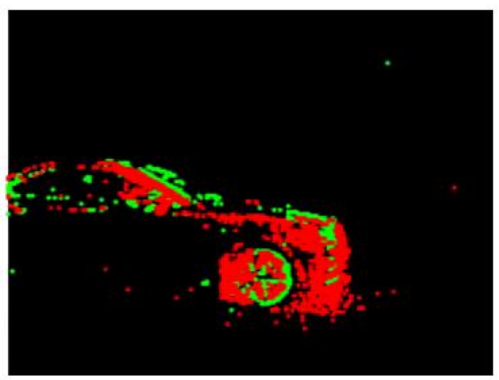


Figure 28. The angular velocity estimates of compared algorithms on (upper) *indoor* and (lower) *outdoor_night* sequences

Evaluation Results

- Even in the situation in which the camera stops, ours estimates the zero angular velocity of the camera.



Video 16. Event stream in outdoor_day1 sequence.

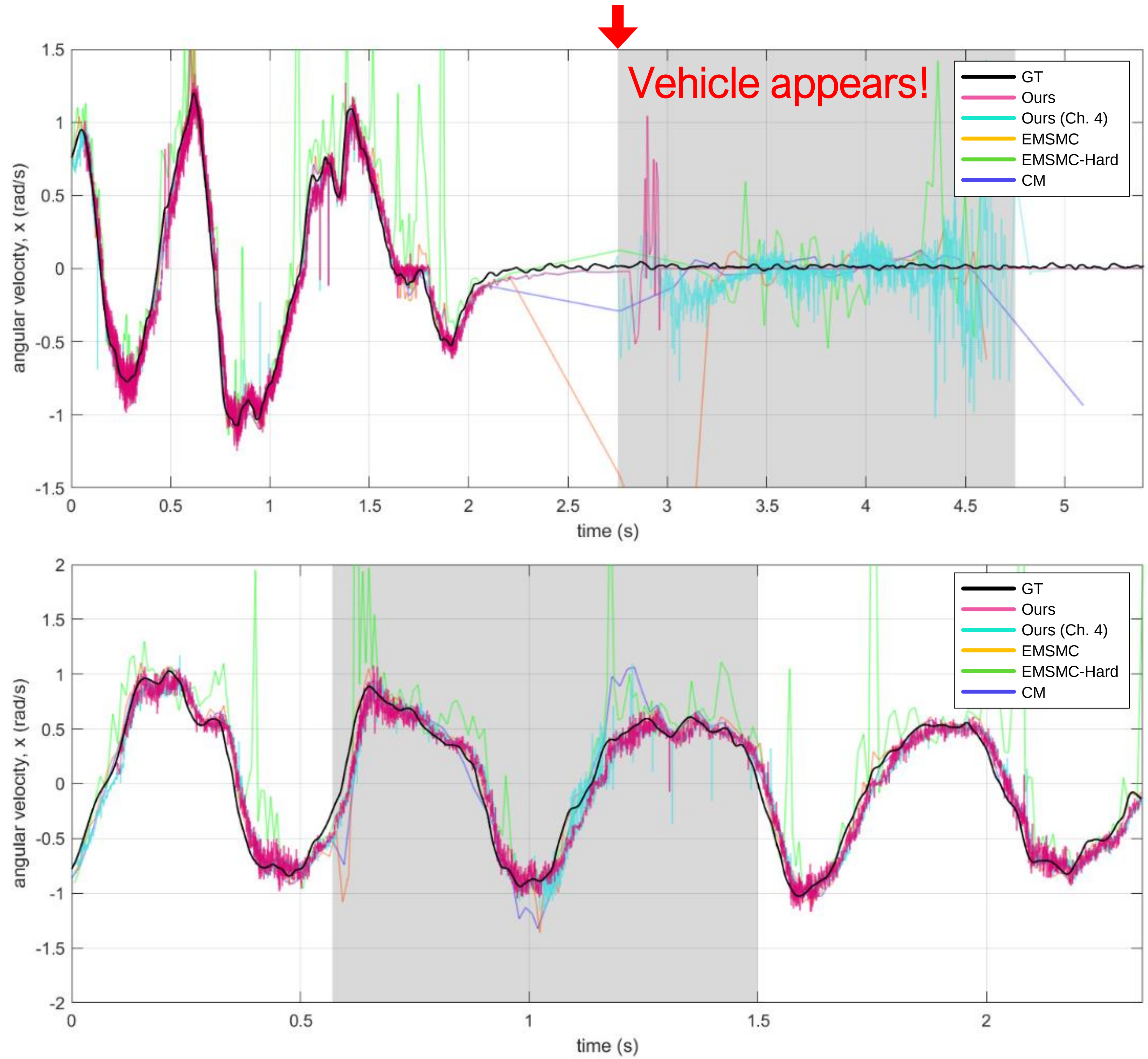


Figure 29. The angular velocity estimates of compared algorithms on (upper) *outdoor_day1* and (lower) *outdoor_day2* sequences

Summary

- We developed an algorithm that segments dynamic foreground from a static background and identifies their regions.
- Therefore, our algorithm robustly estimates angular velocity in dynamic environments where moving object exists.
- Since the motion segmentation using dual-mode motion models and the motion estimation modules are loosely coupled to each other, our motion segmentation can be exploited to the other applications.

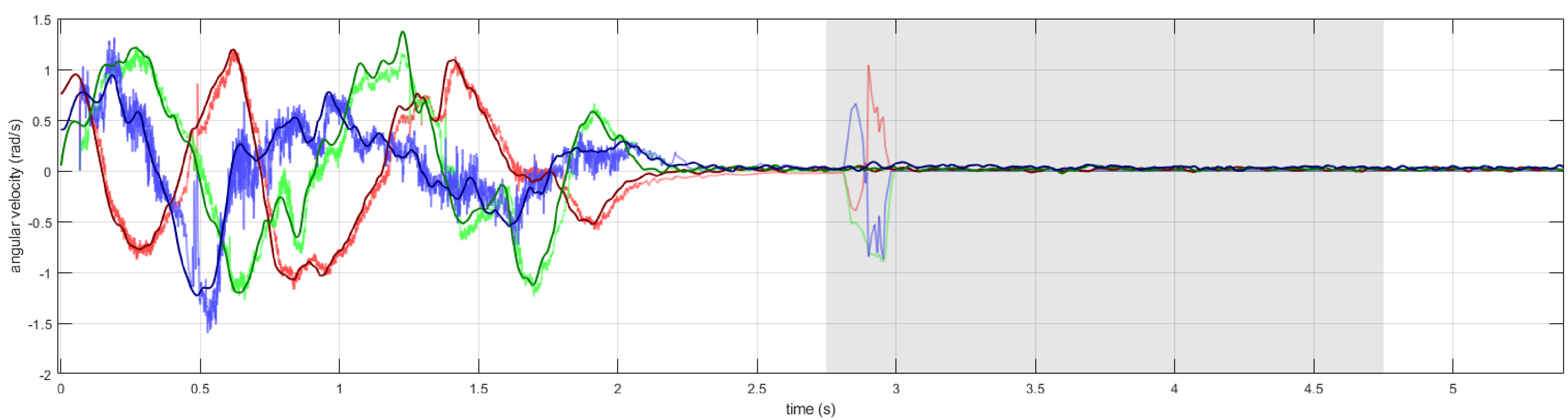


Figure 30. The angular velocity of our algorithm on the *indoor_day1* sequence.

Chapter 6

Summary and Future Work

Robust event-based angular velocity estimation in dynamic environments

- Chapter 3: Visual Flow with Intra-pixel-area Events
  - Intra-pixel-area events for enhancing the performance of plane fitting
  - Visual flow estimation and edge map detection that is accurate and robust

    ⇒ Robustness to timestamp noise

- Chapter 4: Low-latency and Scene-robust Optical Flow
  - Optical flow with low latency under 15 ms consistently
  - Accurate estimation for various scene of tested sequences with the same parameters

    ⇒ Robustness to various scenes

    ⇒ Future research remains to increase the computational efficiency of the algorithm

- Chapter 5: Robust Angular Velocity Estimation in Dynamic Environments
  - Accurate angular velocity estimation using dual-mode motion models

    ⇒ Robustness to moving objects

    ⇒ Future research remains to expand the dimension of motion from 3-DoF to full DoF.

References

[1] Lee, Sangil, Son, Clark Youngdong and Kim, H Jin, "Robust Real-time RGB-D Visual Odometry in Dynamic Environments via Rigid Motion Model", IROS, 2019.

[2] Lee, Sangil, Lee, Chungkeun, Kim, Haram, and Kim, H Jin, "Real-time Object-aware Monocular Depth Estimation in Onboard Systems," International Journal of Control, Automation and Systems, 2021.

[3] E. Mueggler, B. Huber, and D. Scaramuzza, "Event-based, 6-dof pose tracking for high-speed maneuvers," IROS, 2014, pp. 2761–2768.

[4] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," ECCV, 2016, pp.349–364.

[5] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," IEEE Trans. Neural Netw. Learning Syst., vol. 25, no. 2, pp. 407–417, 2014.

[6] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza, "Lifetime estimation of events from dynamic vision sensors," ICRA, IEEE, 2015, pp. 4874–4881.

[7] T. Delbruck, "Frame-free dynamic digital vision," in Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society, 2008, pp. 21–26.

[8] R. Benosman, S.-H. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan, "Asynchronous frameless event-based optical flow," Neural Networks, vol. 27, pp. 32–37, 2012.

[9] M. S. Prieto and A. R. Allen, "A similarity metric for edge images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 10, pp. 1265–1273, 2003.

[10] M. Liu and T. Delbruck, "Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors," BMVC, 2018.

References

[11] Zhu, A. Z., Thakur, D., Ozaslan, T., Pfrommer, B., Kumar, V., & Daniilidis, K, “The Multi Vehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception.” IEEE Robotics and Automation Letters, 3(3), 2032-2039, 2018.

[12] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, “Unsupervised event-based learning of optical flow, depth, and egomotion,” CVPR, pp. 989–997, 2019.

[13] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, “Ev-flownet: Self-supervised optical flow estimation for event-based cameras,” arXiv, 2018.

[14] C. Lee, A. K. Kosta, A. Z. Zhu, K. Chaney, K. Daniilidis, and K. Roy, “Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks,” ECCV, 2020.

[15] C. Lee, “Spike-FlowNet: Event-based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks,” arXiv, 2020.

[16] G. Gallego, et al., “A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation,” CVPR, 2018.

[17] T. Stoffregen, et al., "Event-based motion segmentation by motion compensation," ICCV, 2019.

Thank you for your attention!

Sangil Lee

- **Affiliation**: Laboratory for Autonomous Robotics Research (LARR),
  Seoul National University
- **Advisor**: H. Jin Kim
- **Date**: November 23, 2021

Appendix 1-1: Motivation

- Real work environment is apparently different from well-defined environment.
- Humans and vehicles can interfere in front of the camera.

Video A-1. Well-defined dataset. (left) KITTI [1], (right) DAVIS240C [2]

Figure A-1. Examples of dynamic environments [3,4]

[1] Andreas Geiger, et al., “Vision meets Robotics: The KITTI Dataset,” IJRR, 2013.
[2] E. Mueggler, et al., “The Event-Camera Dataset and Simulator: Event-based Data for Pose Estimation, Visual Odometry, and SLAM,” IJRR, 2017.
[3] "[서울포토] ‘인천공항 안내도 하고 사진도 찰칵!’”, “https://www.seoul.co.kr/news/newsView.php?id=20180711500128”, Accessed: 2021.10.29.
[4] “신호등 지키고, 울퉁불퉁 길 피하고... 로봇 일개미 “점심 배달 왔습니다””, “https://www.chosun.com/economy/tech_it/2021/01/19/2CZGVGDZSFC3DOATRTIX5QWKZU/”, Accessed: 2021.10.29.

Appendix 1-2: Retina vs Frame-based Camera

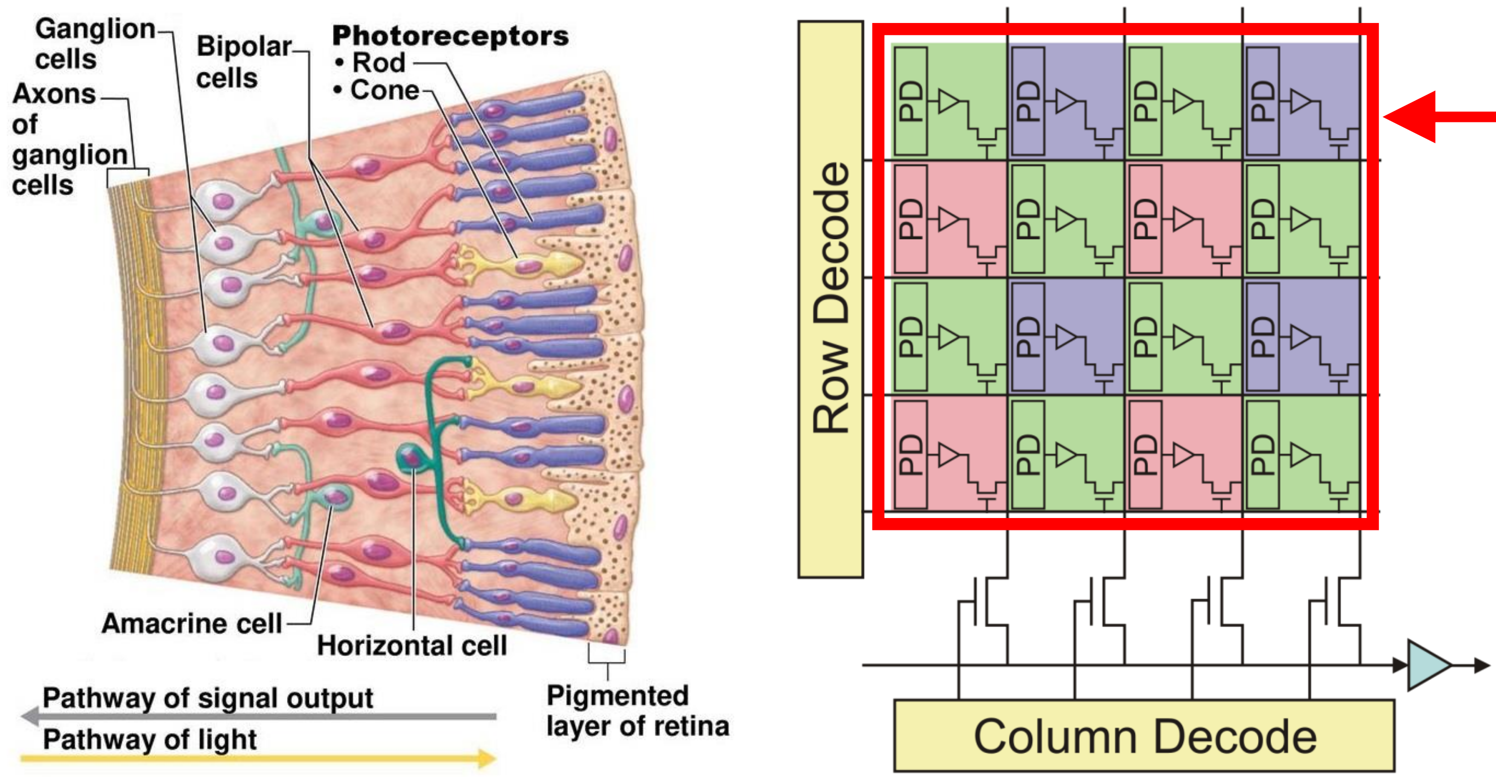


Figure A-2. Biological retina vs. CMOS image sensor.
Copyright ©PearsonEducation, Inc., 2013.

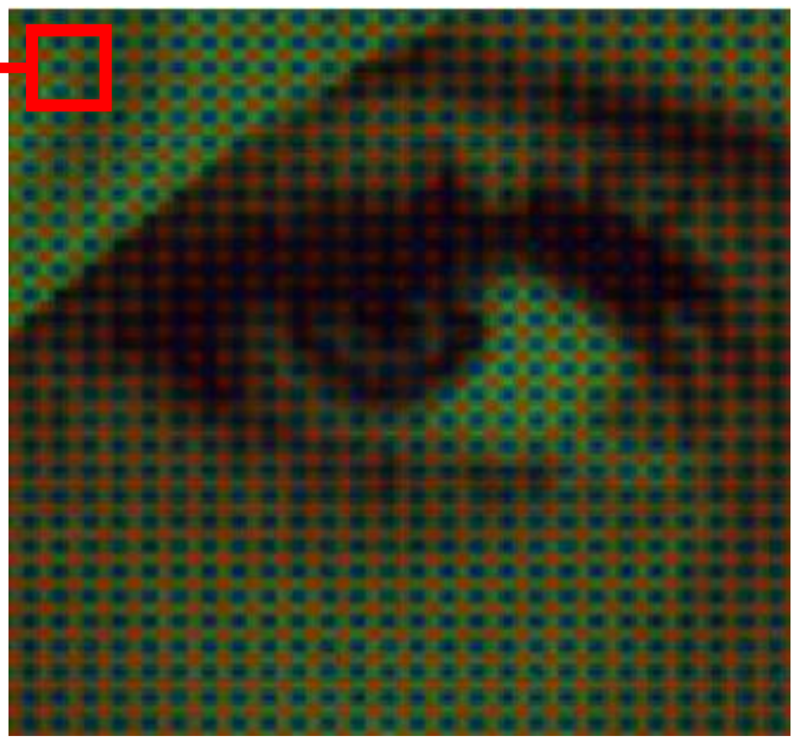


Figure A-3. Bayer pattern of image

- The biological retina and CMOS image sensor have similar mechanisms in terms of a single receptor.
- However, the big difference is how each system delivers light information, i.e., asynchronism vs. synchronism.

Appendix 2-1: Retina

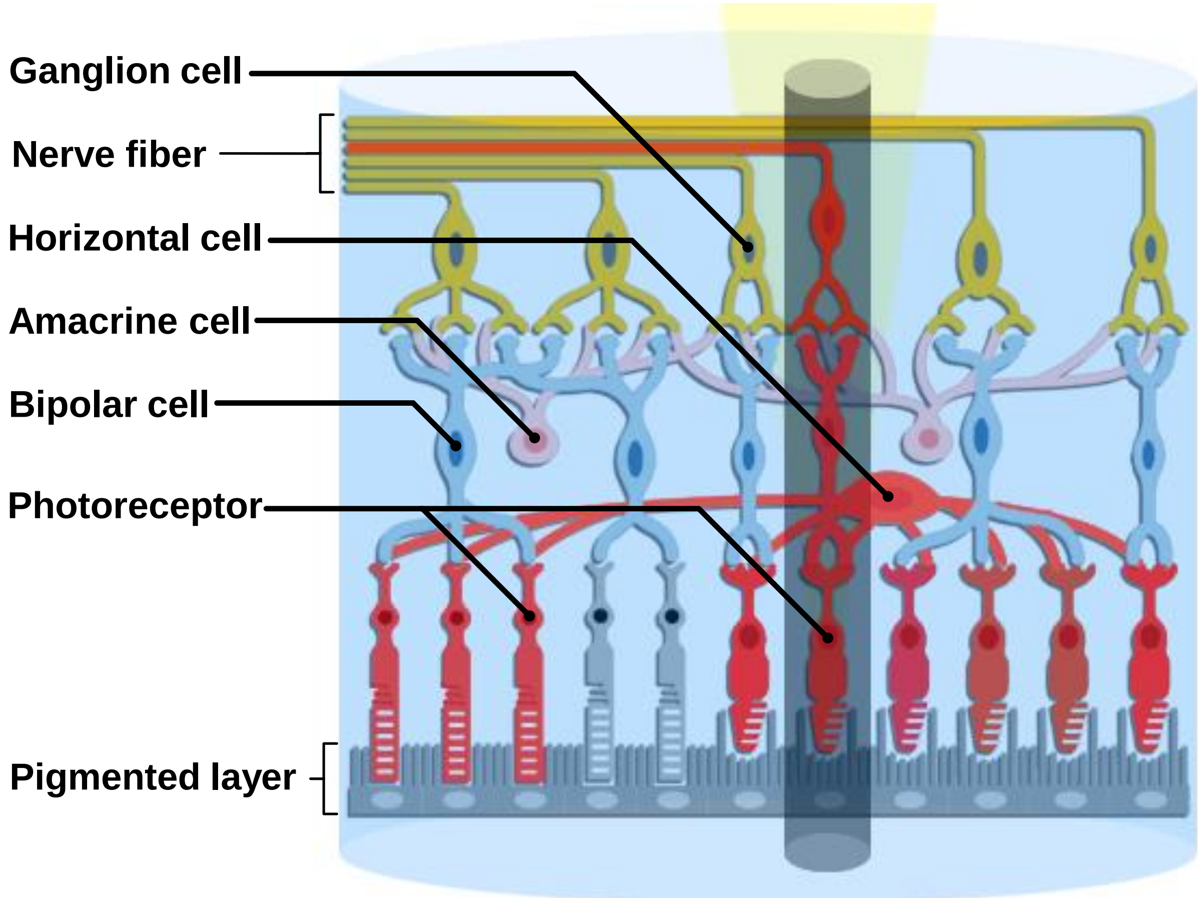


Figure A-4. Biological retina and receptive field of visual system.

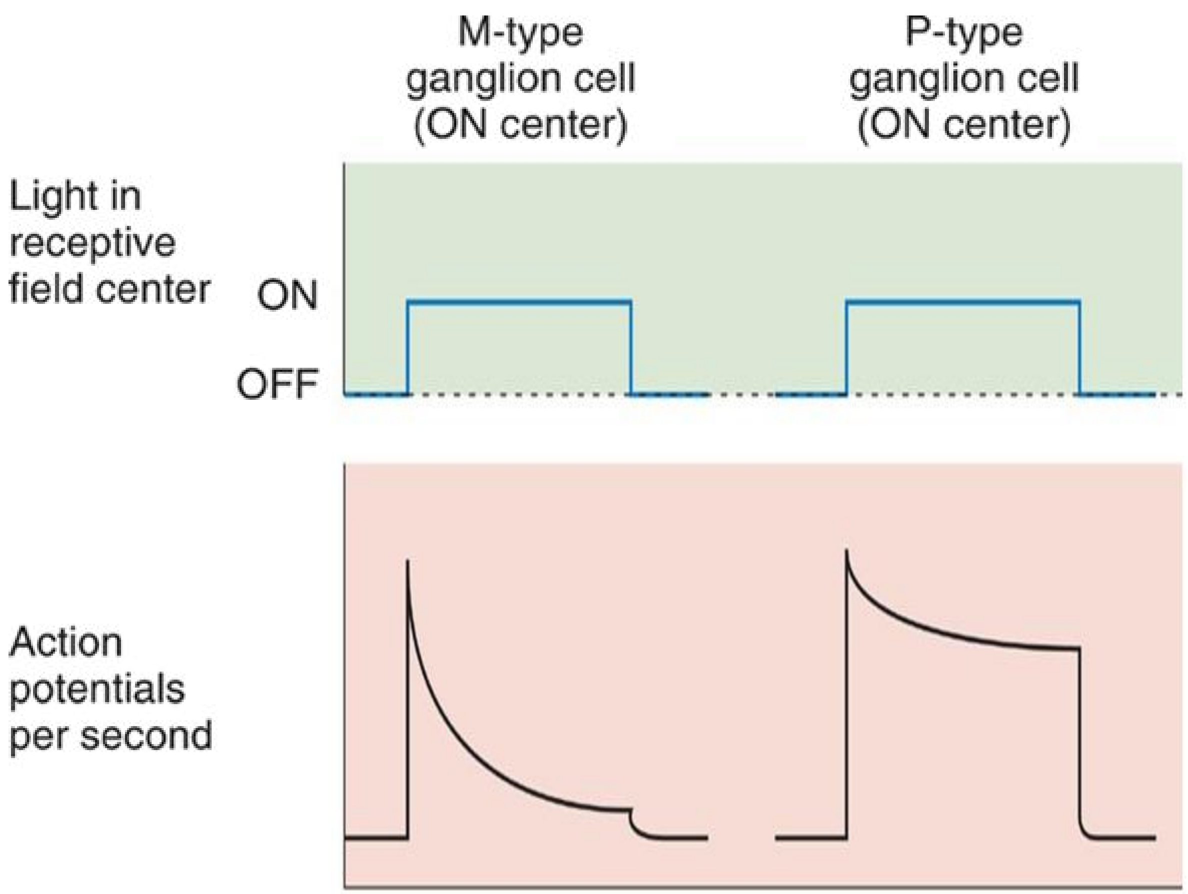


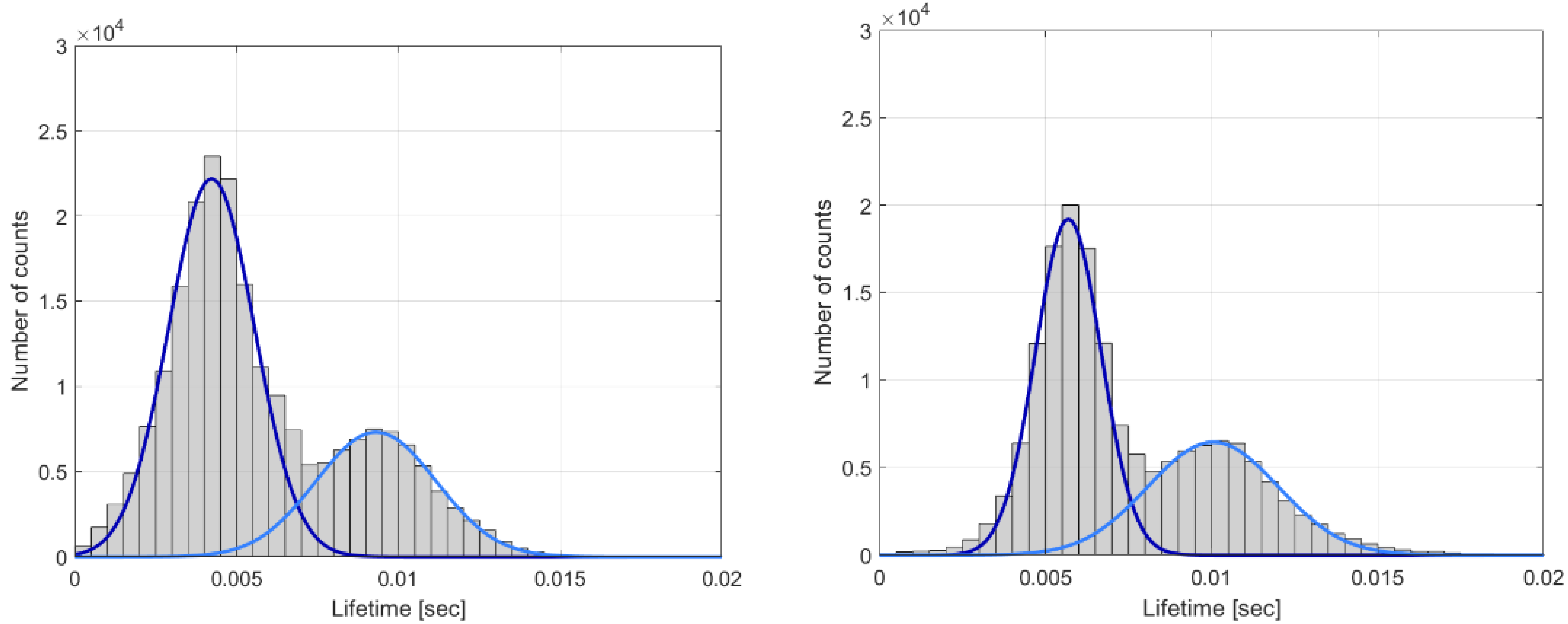
Figure A-5. Action potential of magno- and parvo-ganglion cell

- The ganglion cells in vertebrate retina constitute magnocellular (M-type) or parvocellular (P-type) pathway that respond to motion or color/shapes, respectively.
- Event camera is developed to imitate the magnocellular pathway in the brain.

Appendix 3-1: Histogram of lifetime estimates

- To measure the preciseness of lifetime estimates, Gaussian curve is fitted to the histogram.

- In (a), $\mu_1 = 0.004, \sigma_1 = \mathbf{0.0013}$ and $\mu_2 = 0.009, \sigma_2 = 0.0019$,
  In (b), $\mu_1 = 0.006, \sigma_1 = \mathbf{0.0010}$ and $\mu_2 = 0.010, \sigma_2 = 0.0019$.

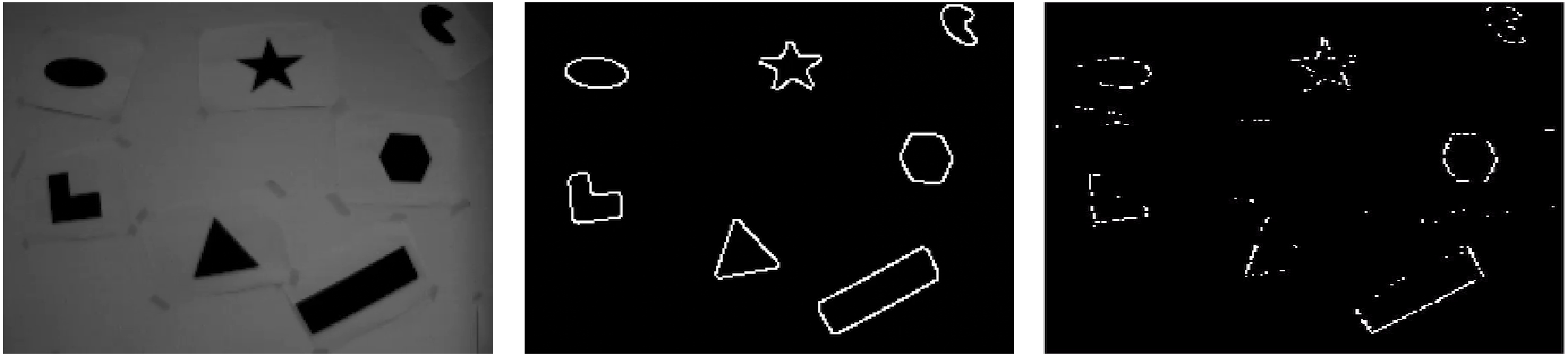


(a) E. Mueggler [6]   (b) Proposed

Figure A-6. The histogram of lifetime estimates. Gaussian curves fitted to the histogram are drawn overlaid.

Appendix 3-2: Frame vs. event-based edge detection

- Canny edge can be regarded as the ground truth of edge detection only at that time the frame is generated.
- Event-based edge detection shows a much higher temporal resolution.

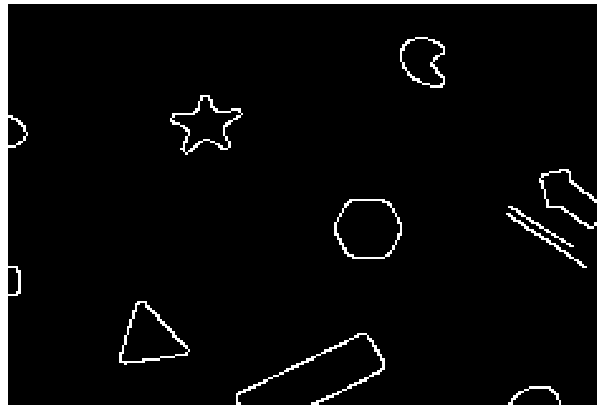


Gray image | Canny edge (GT) | Proposed | **0.2x**

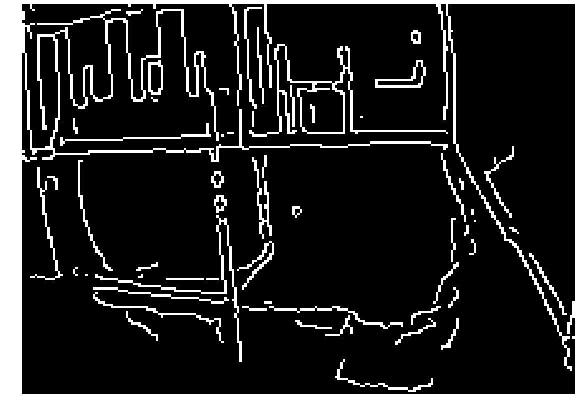
Video A-2. Grayscale image and the detected edge map by Canny and the proposed event-base method.

Appendix 3-3: Self-collected Sequence


(a) Gray image

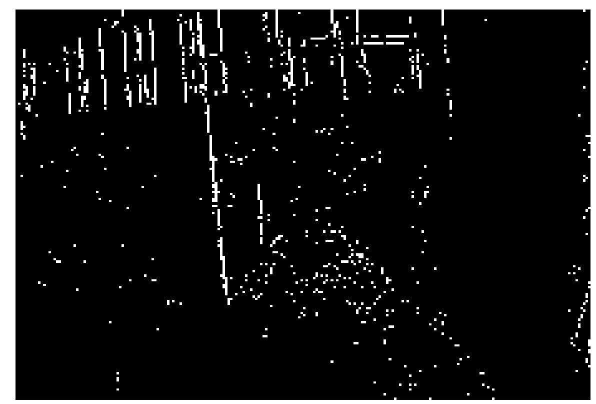

(b) Canny edge

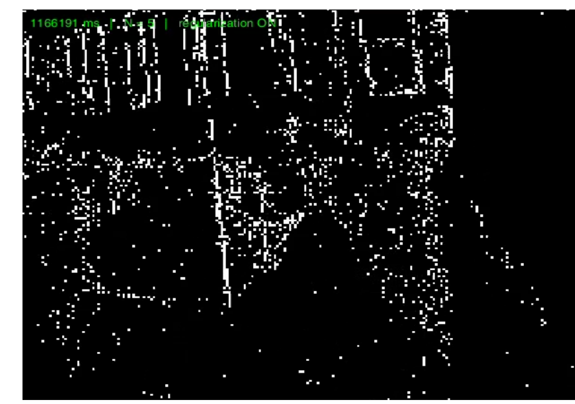

(a) Gray image

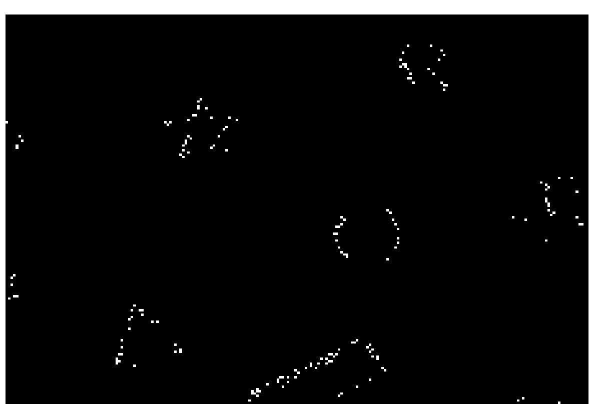

(b) Canny edge

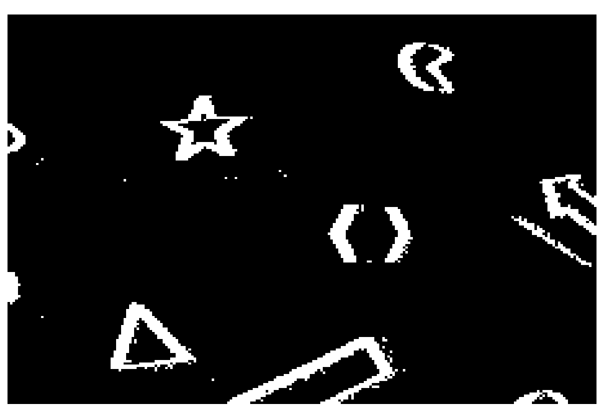

(c) Proposed

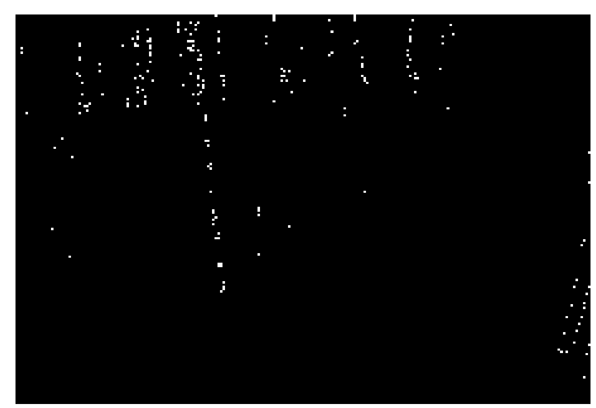
(d) E. Mueggler [6]


(c) Proposed


(d) E. Mueggler [6]


(e) 1ms accumulation


(f) 30ms accumulation


(e) 1ms accumulation


(f) 30ms accumulation

Video A-3. The result of edge detection for each sequence: shapes_rotation and self-collected sequence.

Appendix 4-1: Asynchronous Optical Flow

- Adaptive time-slice block-matching optical flow (ABMOF) vs. Proposed:
  1. global time slice over whole image frame vs. local time slice over each patch
  2. share dt vs. compute individual dt_i

- In Algorithm A-1, pseudocode for motion spatial segmentation is provided.

Algorithm A-1. Asynchronous optical flow estimation

Input: $e_i = (t_i, p_i, p_i)$ ▷ *single event*

Output: $OF_i = (t_i, p_i, v_i, dt_i)$ ▷ *optical flow*

1: **for all** $p_j \in \mathrm{Adj}(p_i)$ **do** ▷ *adjacent pixels in* $(w \times w)$
2: push e_i into queue(p_j)
3: **end for**
4: **if** queue(p_i) is full **then**
5: construct $\mathrm{SAE}_{i,curr}$ and $\mathrm{SAE}_{i,prev}$ from queue(p_i)
6: compute v_i by matching $\mathrm{SAE}_{i,curr}$ and $\mathrm{SAE}_{i,prev}$
7: $dt_i \leftarrow \mathrm{mean}(\mathrm{SAE}_{i,curr}) - \mathrm{mean}(\mathrm{SAE}_{i,prev})$
8: **return** OF_i
9: **end if**

Appendix 4-2: Discussion on Robustness

- Ours estimates accurate angular velocity with low latency and robustness to various scenes, whereas the performance of CM highly depends on the texture of the scene.

Table A-1. Evaluations of angular velocity. Best results are in bold.

Sequence		Average latency (ms) Ours	eSNN	CM	Average accuracy (rad/s) Ours	eSNN	CM
shapes	low	**14.40**	31.54	63.42	**0.110**	0.463	0.203
	mid	**5.15**	35.59	20.05	**0.228**	1.291	0.678
	high	**2.93**	35.93	14.57	**0.350**	2.578	1.154
	whole	**3.04**	35.99	14.42	**0.296**	1.972	0.987
boxes	low	7.51	45.49	**5.51**	**0.138**	0.447	0.145
	mid	1.86	39.90	**0.39**	**0.297**	1.333	0.380
	high	0.59	38.00	**0.26**	**0.355**	2.451	0.396
	whole	0.58	38.31	**0.18**	**0.328**	1.856	0.378
poster	low	8.21	39.74	**7.37**	**0.238**	0.458	0.250
	mid	2.14	40.31	**1.60**	**0.303**	1.163	0.463
	high	0.52	39.29	**0**	**0.359**	2.901	0.461
	whole	0.49	39.40	**0**	**0.328**	1.998	0.443
dynamic	low	**8.15**	46.36	9.68	0.231	0.594	**0.149**
	mid	**4.35**	43.67	5.63	0.242	0.628	**0.160**
	high	**1.80**	41.69	2.44	0.289	1.314	**0.183**
	whole	**1.71**	41.67	2.32	0.264	0.939	**0.169**

Table A-2. Evaluations with different parameters. Results better than ours are in bold.

Average latency (ms) KF	$ABMOF_*$	CM_{5k}	CM_{50ms}	Average accuracy (rad/s) KF	$ABMOF_*$	CM_{5k}	CM_{50ms}
16.47	20.16	16.40	28.54	0.126	0.140	0.138	0.136
5.31	6.13	**5.14**	23.95	0.952	0.329	0.285	0.758
-	3.79	**2.91**	26.94	-	0.454	0.397	1.960
-	3.84	**2.93**	26.56	-	0.401	0.353	1.450
12.16	11.27	-	22.92	**0.095**	0.333	-	**0.123**
3.28	2.60	-	26.66	**0.204**	0.511	-	0.742
1.33	0.70	-	-	**0.263**	0.566	-	-
1.26	0.67	-	-	**0.239**	0.537	-	-
14.25	**4.03**	-	23.52	**0.124**	0.484	-	**0.144**
3.31	**1.63**	-	24.90	**0.218**	0.498	-	0.654
1.54	0.81	-	-	**0.283**	0.619	-	-
1.50	0.76	-	-	**0.250**	0.570	-	-
15.28	**5.36**	-	27.85	**0.167**	0.502	-	**0.213**
10.36	**3.63**	**1.53**	25.04	**0.216**	0.608	1.280	0.263
4.24	**1.71**	**0**	23.34	0.394	0.678	2.637	0.738
4.22	**1.56**	**0**	23.39	0.309	0.616	2.641	0.491

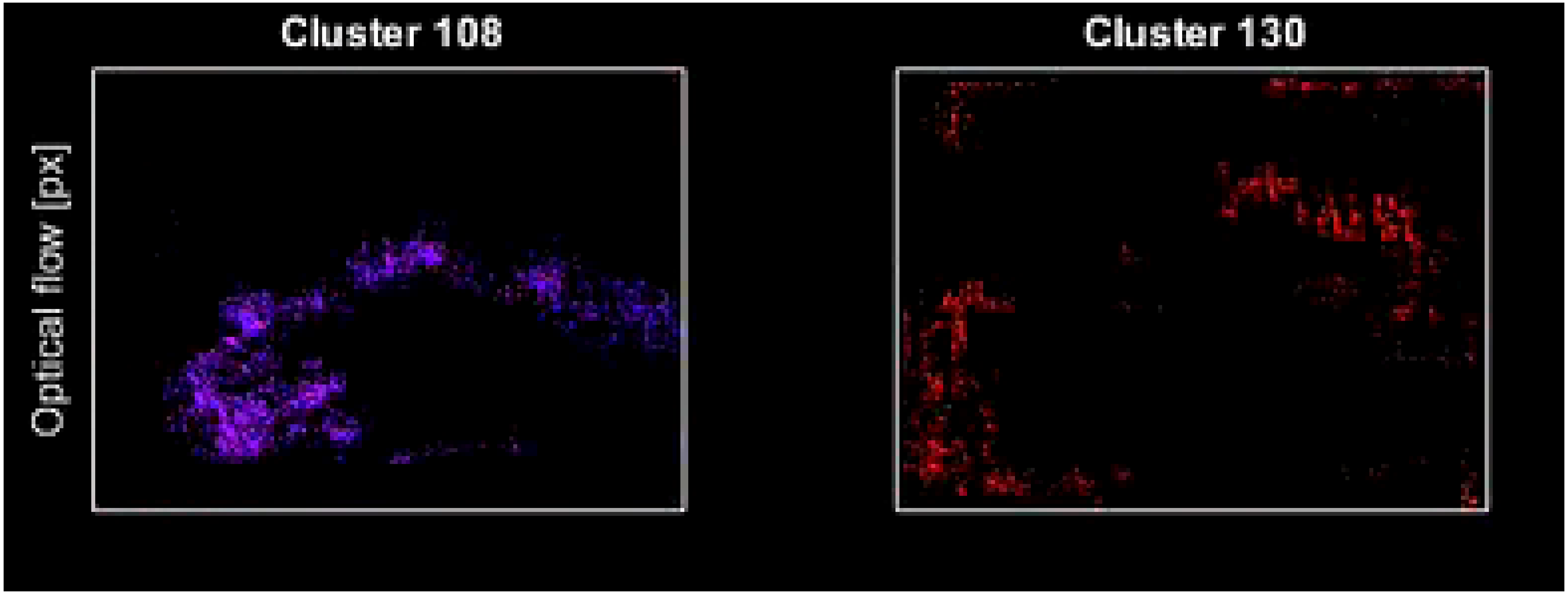
Appendix 4-3: Discussion on Robustness

Table A-3. Evaluations for latency and accuracy of modifications of CM framework. The second row denotes the event grouping method of each CM framework. Results better than ours are in bold. Results larger than 1 are in red.

Sequence		Average latency (ms)								Average accuracy (rad/s)							
		$10ms$	$50ms$	$0.1s$	$0.2s$	$2.5k$	$5k$	$10k$	$15k$	$10ms$	$50ms$	$0.1s$	$0.2s$	$2.5k$	$5k$	$10k$	$15k$
shapes	low	**7.93**	28.54	53.28	104.11	**5.62**	16.40	36.08	63.42	0.816	0.136	0.177	0.316	0.184	0.138	0.154	0.203
	mid	**2.56**	23.95	65.93	126.74	**1.46**	**5.14**	12.62	20.05	0.276	0.758	2.012	3.404	0.372	0.285	0.451	0.678
	high	**2.55**	26.94	73.89	-	**0.18**	**2.91**	8.54	14.57	0.351	1.960	4.909	-	0.491	0.397	0.710	1.154
	overall	**2.55**	26.56	72.52	-	**0.20**	**2.93**	8.56	14.42	0.450	1.450	3.649	-	0.441	0.353	0.615	0.987
boxes	low	8.16	22.92	49.58	-	-	-	**3.74**	**5.51**	0.966	**0.123**	0.197	-	-	-	1.345	0.145
	mid	2.76	26.66	-	-	-	-	**0.15**	**0.39**	0.301	0.742	-	-	-	-	0.994	0.380
	high	2.83	-	-	-	-	-	**0.51**	**0.26**	**0.234**	-	-	-	-	-	0.943	0.396
	overall	2.79	-	-	-	-	-	**0.40**	**0.18**	0.520	-	-	-	-	-	0.993	0.378
poster	low	-	23.52	46.65	92.32	-	-	**7.93**	**7.37**	-	**0.144**	**0.207**	**0.236**	-	-	1.464	0.250
	mid	-	24.90	-	-	-	-	**2.07**	**1.60**	-	0.654	-	-	-	-	1.523	0.463
	high	-	-	-	-	-	-	**0**	**0**	-	-	-	-	-	-	1.100	0.461
	overall	-	-	-	-	-	-	**0**	**0**	-	-	-	-	-	-	1.290	0.443
dynamic	low	**5.22**	27.85	52.73	-	-	-	**4.85**	9.68	0.348	**0.213**	0.353	-	-	-	**0.185**	**0.149**
	mid	**2.58**	25.04	46.84	-	-	**1.53**	**2.76**	5.63	**0.231**	0.263	0.468	-	-	1.280	**0.203**	**0.160**
	high	2.63	23.34	-	-	-	**0**	**0.81**	2.44	**0.166**	0.738	-	-	-	2.637	**0.253**	**0.183**
	overall	2.60	23.39	-	-	-	**0**	**0.83**	2.32	**0.266**	0.491	-	-	-	2.641	**0.225**	**0.169**

Appendix 5-1: Particle-filter-based Object Detection

- A filter-based motion segmentation algorithm fails to identify a static background.



Video A-4. A particle-filter-based motion segmentation algorithm developed before.

Appendix 5-2: Motion Temporal Matching

- A matching coefficient between the i-th and the j-th motion at different timestamp *(k)* and *(l)*:

$$C_{ij}^{(k,l)} = \frac{\sum (K_\sigma * \delta(G^{(k)}, i)) \odot \delta(G^{(l)}, j)}{\sqrt{\sum \delta(G^{(k)}, i) \sum \delta(G^{(l)}, j)}}, \quad C_{ij}^{(k,l)} = \frac{1}{1 + ||\omega_i^{(k)} - \omega_j^{(l)}||_2^2},$$

where δ operator constructs a binary matrix with one if the element value of the first term is equal to the second term:

$$B = \delta(A, x) = (b_{ij}),$$

$$b_{ij} = \begin{cases} 1, & \text{if } a_{ij} = x \\ 0, & \text{otherwise.} \end{cases}$$

and operator $\odot$ and * denote elementwise multiplication and 2D convolution, respectively, and K_σ is a kernel matrix with all elements 1.

Appendix 5-3: Evaluation Results

(a) indoor1

(b) outdoor_night

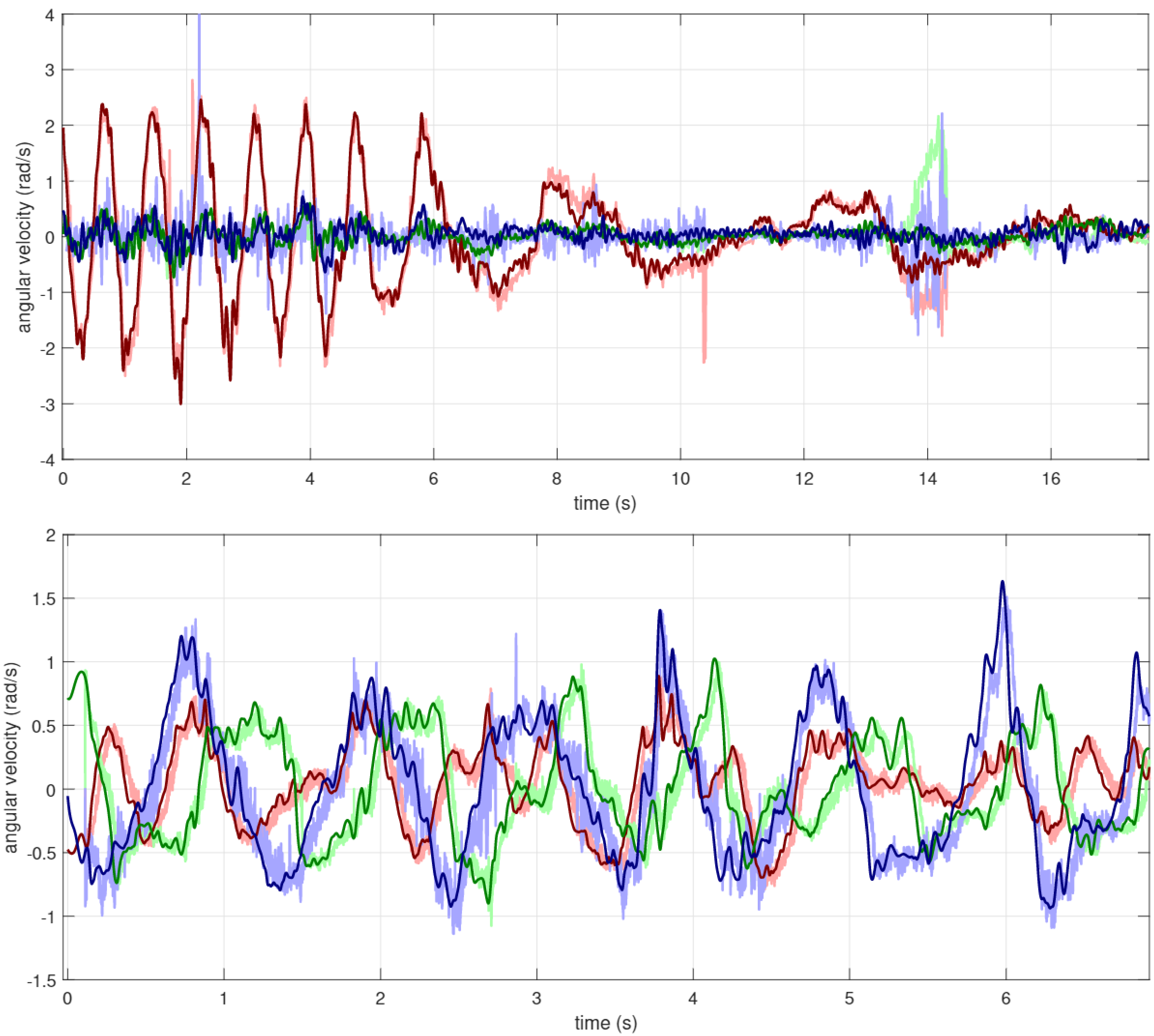
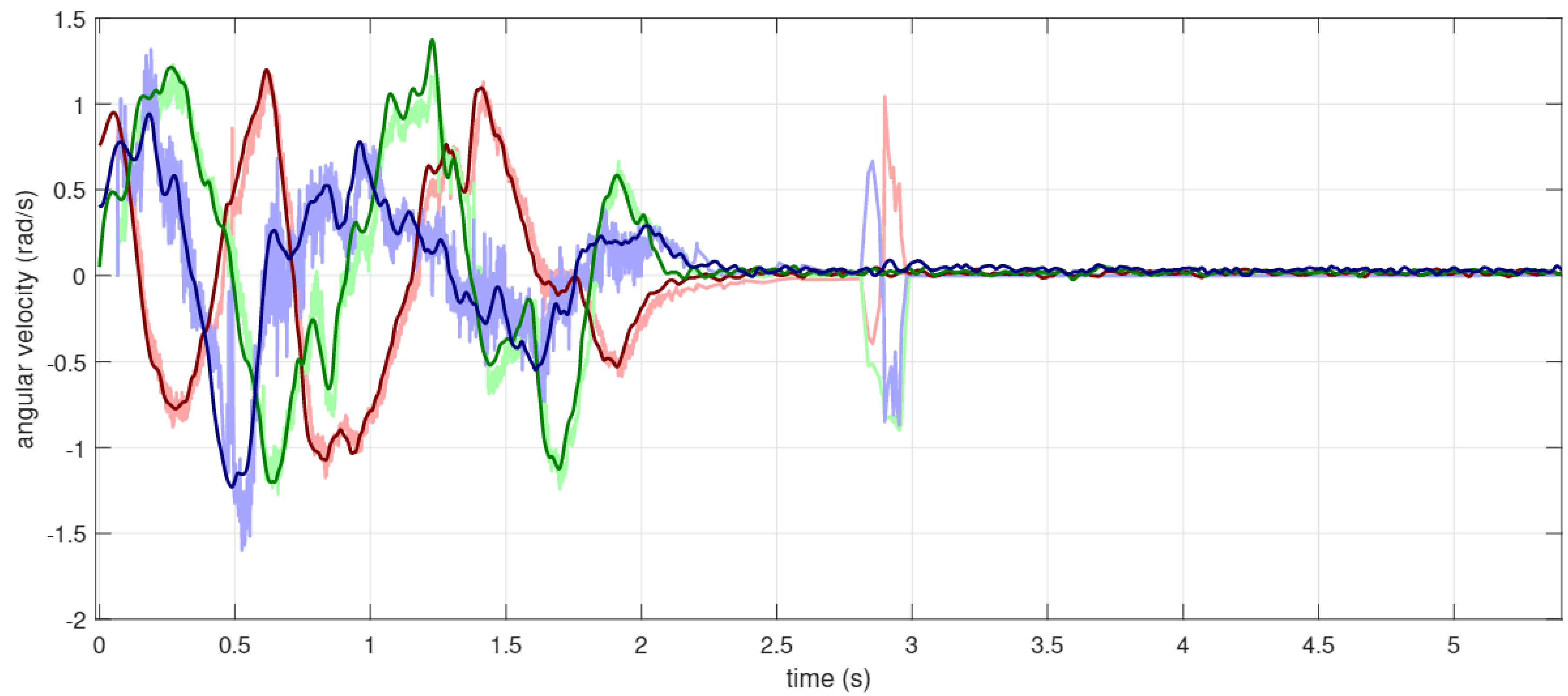


Figure A-7. The angular velocity estimates of our algorithm on (upper) *indoor* and (lower) *outdoor_night* sequences

Appendix 5-4: Evaluation Results

(a) outdoor_day1



(b) outdoor_day2

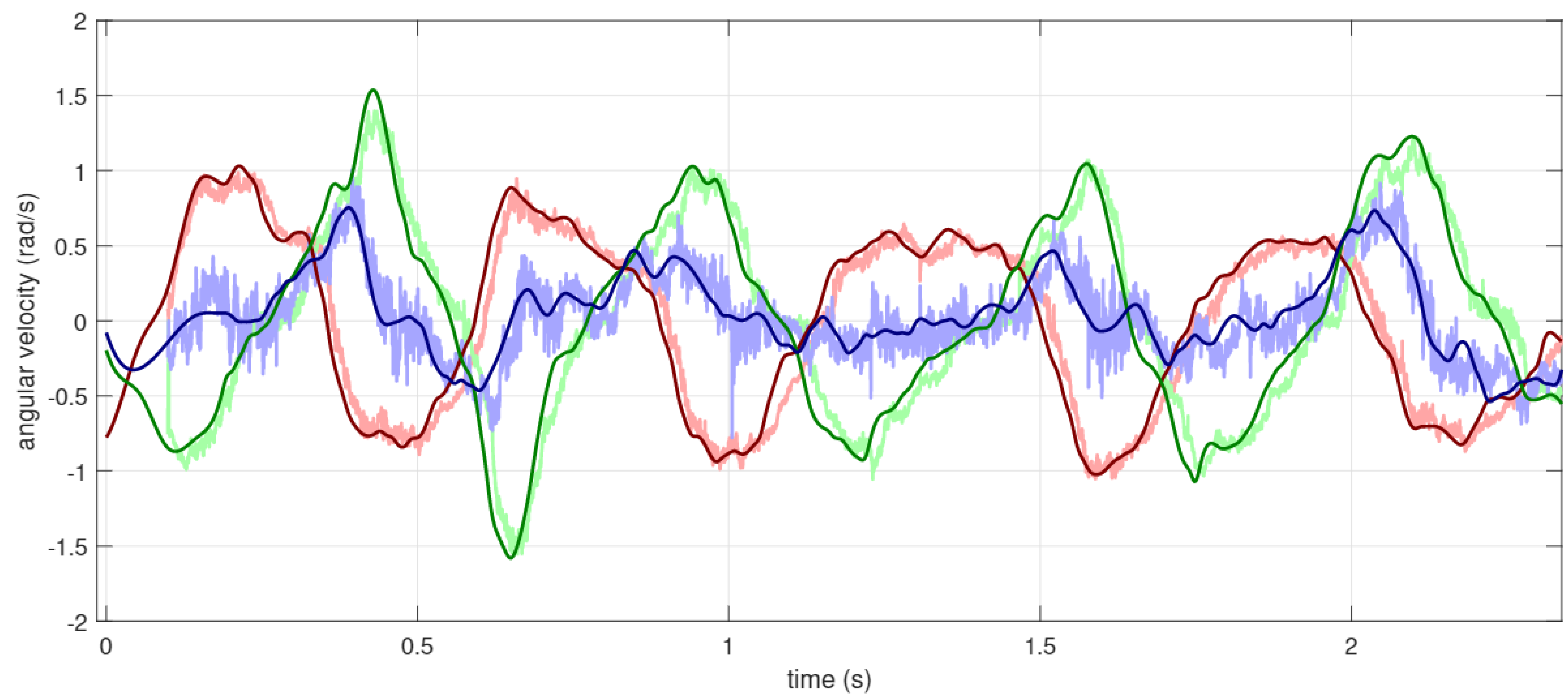


Figure A-8. The angular velocity estimates of our algorithm on (upper) *outdoor_day1* and (lower) *outdoor_day2* sequences