Progressive Spatio-temporal Alignment for Efficient Event-based Motion Estimation Supplementary Material

Xueyan Huang Yueyi Zhang * Zhiwei Xiong
University of Science and Technology of China
hxy2020@mail.ustc.edu.cn, {zhyuey, zwxiong}@ustc.edu.cn

1. Distinctions of Our Loss

In general, the losses in the event-based motion estimation field can be divided into two categories: the statistical loss [2] and the registration loss. The classification of related methods is presented in Tab. 6. For the statistical losses, *e.g.,* the image variance in CMax [3], the probabilistic likelihood in ST-PPP [4], and the entropy in EMin [8], they measure the quality of the event alignment based on the state of the whole events in the batch. For the registration losses, *e.g.,* the spatio-temporal consistency in STR [5] and the timestamp consistency in the surface matching (SM) loss [7], they slice the events as a reference part and a target part. They measure the quality of the event alignment by the degree of the registration between these two parts. Since our TS loss measures the alignment between the later events and the former events in the TS map, it can be classified as a registration loss.

Loss	Method
statistical	CMax, ST-PPP, EMin
registration	STR, SM, Ours

Table 6. Classification of related methods based on loss functions.

In STR [5], they conduct the registration in an event-to-event scheme. They register events individually in the spatio-temporal domain based on their geometric distance, which can be expressed as

$$L_{STR}(\boldsymbol{\theta}) = \sum_{k=1}^{N_e/2} \left\| \mathbf{x}_{F(k)} - \mathcal{W}\left(\mathbf{x}_k, t_k; \boldsymbol{\theta}\right) \right\|_2 , \tag{14}$$

where $F(k)$ returns the index of the closest temporal neighborhood of the event e_k. In SM [7], they conduct the registration in a map-to-map scheme. They crop two patches from two time-surface maps and register them based on their timestamp consistency, which can be expressed as

$$L_{SM}(\boldsymbol{\theta}; \mathbf{p}_0) = \sum_{\mathbf{p} \in H(\mathbf{p}_0)} \left\| \mathcal{S}(\mathbf{p}) - \mathcal{S}'(\mathbf{p} + \boldsymbol{\theta} \cdot \Delta t) \right\|_1 , \tag{15}$$

where Δt is the time shift between the time-surface map $\mathcal{S}$ and $\mathcal{S}'$, and $\mathbf{p}$ is the pixel in the patch $H(\mathbf{p}_0)$ centered at $\mathbf{p}_0$. Distinct from these two registration-based methods, our method performs in an event-to-map scheme that registers later events in the spatio-temporal domain to the former events in the TS map based on their temporal information, which can be expressed as

$$L(\boldsymbol{\theta}) = \sum_{e_k \in \xi_s} \mathcal{I}\left(\mathcal{W}\left(\mathbf{x}_k, t_k; \boldsymbol{\theta}\right)\right) . \tag{16}$$

Unlike the one-to-one registration in [5, 7], our scheme allows many-to-one registration, *i.e.,* many later events can register to a single former event. Moreover, the many-to-one registration can cope with the sampling strategy to reduce the computational burden significantly.

2. Derivation of Forward Warping Formula

Rotational model. According to [1], we can use the Baker-Campbell-Hausdorff (BCH) formula to compound two matrix exponentials. In the particular case of $SO(3)$, the BCH formula can be expressed as

$$\begin{aligned} \ln\left(\mathcal{R}_1\mathcal{R}_2\right)^\vee &= \ln\left(\exp(\widehat{\boldsymbol{\phi}}_1)\exp(\widehat{\boldsymbol{\phi}}_2)\right)^\vee \\ &= \boldsymbol{\phi}_1 + \boldsymbol{\phi}_2 + \widehat{\boldsymbol{\phi}}_1\boldsymbol{\phi}_2 + \cdots, \end{aligned} \tag{17}$$

where $\mathcal{R} \in SO(3)$, $\widehat{\boldsymbol{\phi}} \in so(3)$, $\ln : SO(3) \to so(3)$ is the matrix logarithm and the operator $(\cdot)^\vee$ maps the Lie algebra to the real vector space. In our rotational model, $\boldsymbol{\phi}$ is equal to the rotation angle $\boldsymbol{\theta} \cdot \Delta t$. Since the interval Δt is much smaller than the value of angular velocities $\boldsymbol{\theta}$, we retain the first-order term as

$$\ln\left(\mathcal{R}_1\mathcal{R}_2\right)^\vee \approx \boldsymbol{\phi}_1 + \boldsymbol{\phi}_2. \tag{18}$$

*Corresponding author

Then the forward warping function of Eq. (9) can be approximated as

$$\begin{aligned}&\mathcal{R}\left(t_{end}-t_{start};\boldsymbol{\theta}\right)\cdot R^{-1}\left(t_k-t_{start};\boldsymbol{\theta}\right) \\ &\approx \exp\left(\widehat{\boldsymbol{\theta}}\cdot\left(t_{end}-t_k\right)\right) \\ &=\mathcal{R}\left(t_{end}-t_k;\boldsymbol{\theta}\right)\ .\end{aligned} \tag{19}$$

6-DOF model. Given a transformation matrix $\mathcal{T}$, we represent its inverse matrix $\mathcal{T}^{-1}$ as

$$\mathcal{T}^{-1}=\begin{pmatrix}\mathcal{R} & \boldsymbol{t}\\ 0^{\mathsf{T}} & 1\end{pmatrix}^{-1}=\begin{pmatrix}\mathcal{R}^{-1} & -\mathcal{R}^{-1}\cdot\boldsymbol{t}\\ 0^{\mathsf{T}} & 1\end{pmatrix}. \tag{20}$$

Then the forward warping transformation in Eq. (13) can be expressed as

$$\mathcal{T}_2\mathcal{T}_1^{-1}=\begin{pmatrix}\mathcal{R}_2\mathcal{R}_1^{-1} & -\mathcal{R}_2\mathcal{R}_1^{-1}\cdot\boldsymbol{t}_1+\boldsymbol{t}_2\\ 0^{\mathsf{T}} & 1\end{pmatrix}, \tag{21}$$

where $\mathcal{T}_1^{-1}=\mathcal{T}^{-1}(t_k-t_{start};\boldsymbol{\theta})$ backward warps 3D points from t_k to t_{start}, $\mathcal{R}_1$ and $\boldsymbol{t}_1=\boldsymbol{v}\cdot(t_k-t_{start})$ are the rotation part and the translation part of $\mathcal{T}_1$, $\mathcal{T}_2=\mathcal{T}(t_{end}-t_{start};\boldsymbol{\theta})$ forward warps 3D points from t_{start} to t_{end}, $\mathcal{R}_2$ and $\boldsymbol{t}_2=\boldsymbol{v}\cdot(t_{end}-t_{start})$ are the rotation part and the translation part of $\mathcal{T}_2$.

3. Ablations on the Optimization Step and the Iteration Number

As described in Algorithm 1, our method consists of two parts: alternately updating the TS map with the latest estimated motion parameters; iteratively optimizing the motion parameters by minimizing the TS loss. In our experiments, we empirically set the iteration number $T=2$, *i.e.*, creating the TS map twice, and the optimization step $S=10$ in each iteration.

In Tab. 7, we present an ablation on the optimization step with the rotational model. For a fair comparison, we fix the total update number $T\cdot S=20$ and follow the same experimental settings with the dynamic batch size strategy, $1k$ event samples, and the bidirectional warping strategy. It shows that under the same update number, our method reports higher accuracy with the iterative alignment scheme, *e.g.*, $T=2$, than that without the iterative alignment scheme. However, with more iterations but fewer optimization steps, *e.g.*, $T=4$ and $S=5$, our method reports a performance degradation. Because in each iteration, a small optimization step number may be insufficient to make the parameters converge. Therefore, we set the optimization step $S=10$ in other experiments.

In Tab. 8, we report the estimation results with different iteration numbers T. We keep the same experimental settings as in Tab. 7, but fix the optimization step $S=10$. It shows that, in general, the accuracy increases with more iterations. With $T=5$ iterations, the motion parameters should reach convergence. Thus it may cause overfitting when we keep updating the TS map, *e.g.*, $T=10$. Considering the tradeoff between the processing time and the accuracy, $T=2$ and $S=10$ are the default values in our method.

T	S	*poster_rotation*			*shapes_rotation*		
		e_w	RMS_w	Time	e_w	RMS_w	Time
20	1	15.93	23.92	27.9	21.19	34.12	8.1
4	5	15.46	22.87	7.5	19.80	30.62	4.2
2	10	**6.87**	**10.11**	4.3	**7.17**	**10.85**	3.1
1	20	7.55	11.03	**3.9**	7.73	11.25	**2.8**

Table 7. Ablations on the optimization step with the rotational model. We fix the total update number $T\cdot S=20$. Note that $T=1$ means the iterative alignment scheme is not applied.

4. Visual Results

In Fig. 4, we present the qualitative results of the aligned event frames of different motion estimation methods in the Event-Camera dataset [6]. In the *poster_rotation* sequence, these methods show relatively close alignment results. In the *shapes_rotation* sequence, the fixed batch size of $30k$ events in CMax and ST-PPP can not adapt to the scene texture changes, leading to the misalignment of events.

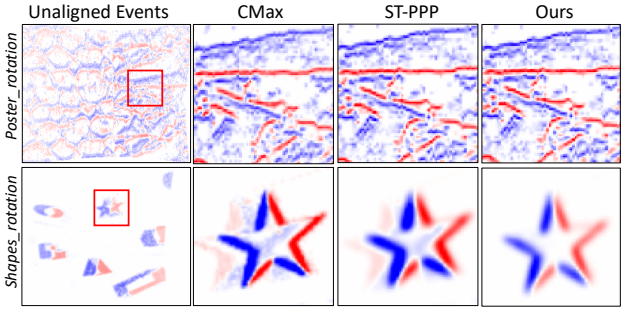

Figure 4. Visual results of the aligned event frames of different methods in the rotational motion estimation task.

5. Video Demonstration

We provide a visual demonstration of our framework with the 6-DOF model in the *indoor_flying* sequence [10], which is available at https://drive.google.com/file/d/1k96ixW2e9eN_TFSDhkHH-45_ltzsHITa/view?usp=share_link. Note that our results are acquired in the real-time mode while AEMin can not achieve the real-time implementation.

T	S	*poster_rotation*			*shapes_rotation*		
		e_w	RMS_w	Time	e_w	RMS_w	Time
1	10	7.92	11.13	**3.4**	8.20	11.92	**2.2**
2	10	6.87	10.11	4.3	7.17	10.85	3.1
5	10	**6.80**	**10.02**	13.2	**7.16**	**10.62**	7.7
10	10	6.81	10.10	25.5	7.18	10.63	14.5

Table 8. Ablations on the iteration number with the rotational model. The optimization step is fixed to 10. Note that $T = 1$ means the iterative alignment scheme is not applied.

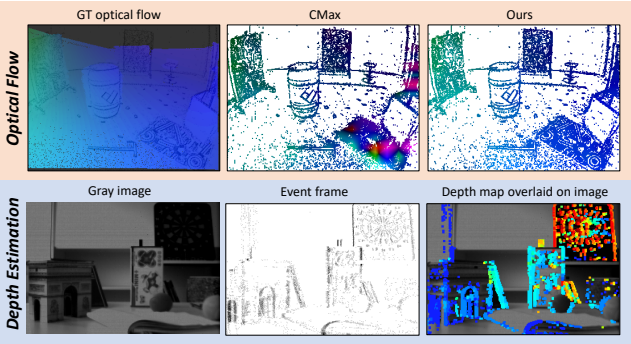

Figure 5. Visual results of our method in optical flow and depth estimation tasks.

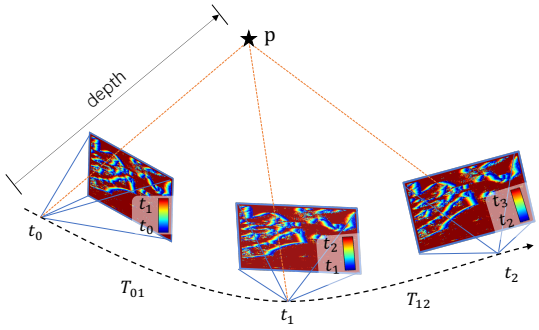

Figure 6. Schematic diagram of our depth estimation method.

6. More Applications

Except the motion estimation task, our method has the potential to solve other problems. We summarize core procedures to apply our method in the following. We also conduct experiments and show qualitative results for optical flow/depth estimation.

a) optical flow estimation. The optical flow motion satisfies the 2D motion model. We can use our method to derive dense optical flow estimation following routines in [9]. We provide the visual results of our method in the *infoor_flying1* sequence of the MVSEC dataset [10], shown in Fig. 5. Note that the estimated flow of CMax may collapse due to the aperture problem.

b) depth estimation. To estimate the depth of the scene, we construct several TS maps along the camera trajectory, as shown in Fig. 6. It is assumed that the camera poses are already known. We start by selecting an event and interpolating its pose according to its timestamp. We then iterate all possible depth values and warp the event to these TS maps using a 6-DOF motion model. We sum the warped pixel values of the event at each TS map, *i.e.,* the TS loss of this event. The best depth value should correspond to the minimum TS loss. Fig. 5 shows our depth estimation results of the *slider_depth* sequence in the Event-Camera dataset [6] Note that this process can be performed progressively, *i.e.,* the TS map can be updated based on the latest estimated depth value with our 6-DOF model. Additionally, when a good initial depth value is available, depth estimation can be performed using optimization methods instead of brute-force search.

References

[1] Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017. 1

[2] Guillermo Gallego, Mathias Gehrig, and Davide Scaramuzza. Focus is all you need: Loss functions for event-based vision. In *Computer Vision and Pattern Recognition*, 2019. 1

[3] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *Computer Vision and Pattern Recognition*, 2018. 1

[4] Cheng Gu, Erik Learned-Miller, Daniel Sheldon, Guillermo Gallego, and Pia Bideau. The spatio-temporal poisson point process: A simple model for the alignment of event camera data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13495–13504, 2021. 1

[5] Daqi Liu, Alvaro Parra, and Tat-Jun Chin. Spatiotemporal registration for event-based visual odometry. In *Computer Vision and Pattern Recognition*, 2021. 1

[6] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam:. *The International Journal of Robotics Research*, 36:142–149, 2017. 2, 3

[7] Jun Nagata, Yusuke Sekikawa, and Yoshimitsu Aoki. Optical flow estimation by matching time surface with event-based cameras. *Sensors*, 21:1150, 2021. 1

[8] Urbano Miguel Nunes and Yiannis Demiris. Entropy minimisation framework for event-based vision model estimation. In *European Conference on Computer Vision*, 2020. 1

[9] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Secrets of event-based optical flow. In *European Conference on Computer Vision*, pages 628–645. Springer, 2022. 3

[10] Alex Zihao Zhu, Dinesh Thakur, Tolga Ozaslan, Bernd G. Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multi

vehicle stereo event camera dataset: An event camera dataset for 3d perception. In *International Conference on Robotics and Automation*, 2018. 2, 3